CSCI 5411 – Advanced Cloud Architecting

Term Project
ClickClick

Aniket Mhatre

B00969798

an370985@dal.ca

Table of Contents

1. Introduction

ClickClick is an online photo sharing platform designed to streamline the process of uploading and sharing images. Its primary function is to provide users with a user-friendly interface where they can effortlessly upload their favorite photos and share them with their friends, family, or the broader online community.

The main features of ClickClick include:

1. **Seamless Login and Upload:** ClickClick offers a seamless login process, allowing users to quickly access their accounts and upload images with ease. This ensures users can share their creations without unnecessary hassle or delays.

2. **Tag Subscription:** ClickClick enables users to stay updated with the latest trends and discover new content effortlessly by subscribing to their favorite tags. This feature allows users to receive notifications whenever new posts are made with tags they've subscribed to, ensuring they never miss out on relevant content.

3. **Automatic Tag Generation:** ClickClick utilizes Amazon Rekognition, a powerful image recognition service, to automatically generate tags for uploaded images. This makes it easier for users to categorize and search for images based on their interests, enhancing the overall discoverability of their content.

4. Email Notifications: ClickClick employs Amazon Simple Notification Service (SNS) to send email notifications to users who have subscribed to specific tags. Whenever a new post is created with a tag they're subscribed to, users receive an email notification, keeping them informed about new content relevant to their interests.

Overall, ClickClick aims to provide a seamless and efficient platform for users to share their photos and discover new content based on their interests. Its target audience includes individuals who want to showcase their work and connect with like-minded users, as well as those who enjoy exploring and discovering new visual content online.

1.1 Hosted Application and Source Repository

Hosted URL - http://clickclick-alb-999926008.us-east-1.elb.amazonaws.com/

1.2 Project Choice

The ClickClick project was selected as it provided an excellent opportunity to leverage a broad range of AWS services across various categories, aligning with the requirements of the term assignment. This project allowed for hands-on experience with different AWS services, showcasing their unique capabilities and integration potential.

Diverse use of AWS Services:

1. **Compute:**
    a. **Amazon ECS with Fargate:** For frontend deployment, showcasing containerized application management without infrastructure overhead.
    b. **AWS Lambda:** For backend functions, demonstrating serverless architecture efficiency.
2. **Storage:**
    a. **Amazon S3:** For storing user-uploaded images, highlighting scalable and durable storage solutions with advanced features.
3. **Database:**
    a. **Amazon DynamoDB:** For managing semi-structured user and post data, emphasizing high availability and automatic scaling.
4. **Networking:**
    a. **AWS VPC:** For creating a secure, isolated network environment with public and private subnets.
5. **Load Balancing:**
    a. **Application Load Balancer (ALB):** To distribute traffic and maintain high availability through load balancing and health checks.
6. **Security and Identity:**
    a. **AWS Cognito:** For user authentication and authorization, integrating securely with API Gateway.
7. **API Management:**
    a. **API Gateway:** To manage REST API endpoints and enhance security with throttling and authorizers.
8. **Machine Learning:**
    a. **Amazon Rekognition:** For automatic image tagging, leveraging managed machine learning services.
9. **Messaging and Notification:**
    a. **Amazon SNS:** For sending real-time email notifications to users.
10. **Infrastructure as Code:**
    a. **AWS CloudFormation:** For automated infrastructure deployment and management.

AWS Well-Architected Framework Alignment:

1. **Operational Excellence:** Automated deployments and continuous monitoring.
2. **Security:** Robust authentication, encryption, and access control.
3. **Reliability:** High availability through ALB and DynamoDB.
4. **Performance Efficiency:** Serverless architecture and efficient resource management.
5. **Cost Optimization:** Pay-per-use services and auto-scaling.
6. **Sustainability:** AWS managed services to minimize resource consumption.

Overall, the ClickClick project provided a comprehensive platform to explore and utilize various AWS services, demonstrating their integration and effectiveness in building a scalable, secure, and efficient application. This experience is invaluable for understanding cloud architecture and best practices in real-world scenarios.

2. Architecture Diagram

This section explores the Architecture diagram of the ClickClick application which uses AWS services.

Note: You can checkout the image here as well ***LINK***

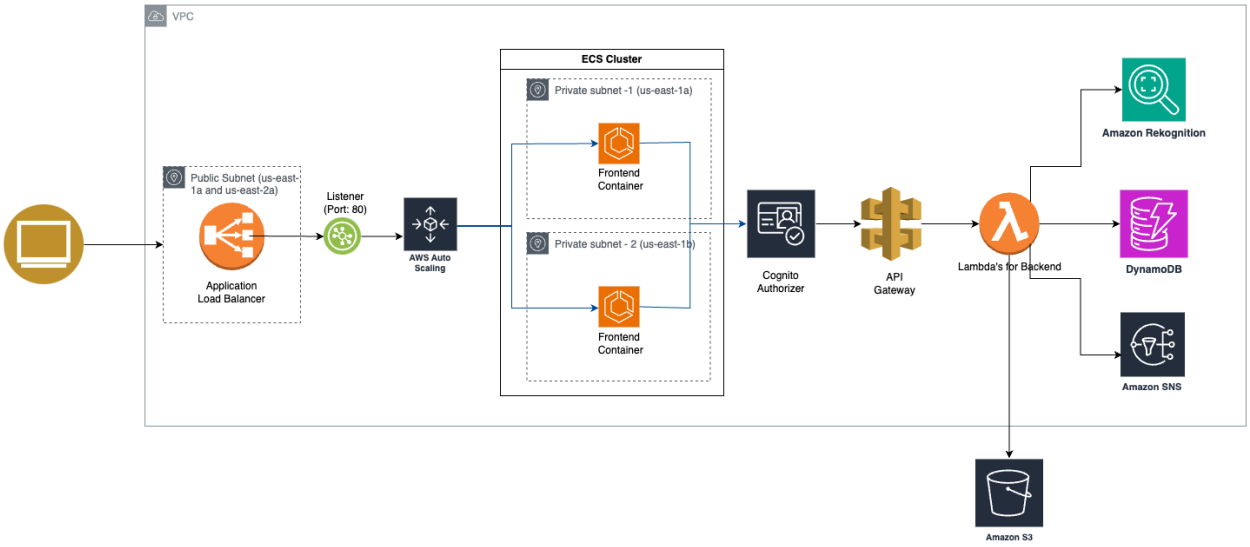

Figure 1. Architecture Diagram of ClickClick Application

3. AWS Services By Category

This section explores the AWS Services which makes up the ClickClick application with the rationale of choosing them.

3.1 Compute

3.1.1 Amazon Elastic Container Service (ECS) with Fargate and Auto Scaling

ECS with Fargate was selected for its simplicity in deployment and scaling, eliminating the need to manage the underlying infrastructure. Tasks run on private subnets, ensuring secure access through the Application Load Balancer [1][4].

Cost: The Auto Scaling Group ensures cost optimization by maintaining a minimum of two containers and a maximum of four, dynamically adjusting based on demand [1].

Performance: Fargate ensures reliable performance by automatically managing container orchestration, with ECS tasks running securely and efficiently [1].

Security: Running tasks on private subnets and restricting access via the Application Load Balancer enhances security. As an AWS-managed service, Fargate provides robust security measures [1][4].

Scalability: Fargate's ability to scale out and scale in based on demand aligns with the AWS Well-Architected Framework's principles, ensuring seamless scalability [1][4].

3.1.2 Amazon Elastic Container Registry (ECR)

ECR was chosen to keep Docker images private and seamlessly integrate with ECS, facilitating the deployment process. ECR was selected over Docker Hub to maintain the privacy of Docker images [3].

Cost: ECR offers cost-effective storage and retrieval of container images, optimized for AWS environments [3].

Security: Security is enhanced through AWS credentials required for image pushes and server-side encryption [3].

3.1.3 AWS Lambda

AWS Lambda was chosen for its serverless architecture and seamless integration with other AWS services like API Gateway, SNS, DynamoDB, and Rekognition [2].

Cost: Lambda operates on a pay-per-invocation model, saving costs by only scaling the required functions based on demand. For example, if the **getAllPost** endpoint is more frequently accessed, only that Lambda function will scale [2].

Security: As an AWS-managed service, Lambda ensures robust security without the need to manage the underlying infrastructure [2].

Scalability: Lambda automatically scales based on the number of invocations, optimizing performance and cost efficiency [2].

This approach satisfies the Cost Optimization and Sustainability pillars of the AWS Well-Architected Framework. By leveraging serverless computing, there is minimal power consumption compared to continuously running EC2 instances, contributing to sustainability efforts [2].

3.2 Storage

3.2.1 Amazon Simple Storage Service (S3)

Amazon S3 is utilized for storing the post images uploaded by users in the ClickClick application due to its superior data durability and availability features. S3 ensures high data durability by storing multiple copies of each object across multiple devices and facilities, significantly reducing the risk of data loss [6][16].

Cost: S3's lifecycle policies help optimize storage costs by automatically moving older, infrequently accessed images to more cost-effective storage options. For instance, images are moved to a cost-effective storage solution after 60 days of upload [6][16].

Performance: S3 can store and retrieve any amount of data, allowing the application to scale usage up or down based on demand without any disruption in service. This flexibility ensures that ClickClick can handle varying volumes of image uploads and access requests efficiently [6] [16].

Security: S3 enhances data management and security through features like object versioning and server-side encryption. Versioning helps protect against accidental deletions and enables easy recovery of previous versions of objects, maintaining the integrity of user-uploaded images over time [6][16].

Scalability: S3 scales seamlessly with the application's needs, providing reliable and scalable storage. Its ability to handle any amount of data ensures that ClickClick can efficiently manage varying volumes of image uploads and access requests [6][16].

Advanced Features:

- **Versioning:** S3 allows for object versioning, which helps protect against accidental deletions and enables easy recovery of previous versions of objects. This feature is particularly useful for maintaining the integrity of user-uploaded images over time [16].

- **Lifecycle Policies:** With S3's lifecycle policies, it is possible to automate the transition of objects between different storage classes based on predefined rules. This helps optimize storage costs by automatically moving old images, which are accessed infrequently, to more cost-effective storage options [16].

This implementation aligns with the AWS Well-Architected Framework's principles of Reliability and Cost Optimization. S3 provides high durability, and images are moved to a cost-effective storage solution after 60 days of upload through lifecycle policies.

3.3 Database

3.3.1 Amazon DynamoDB

Amazon DynamoDB is utilized for storing post data and user data in a NoSQL format. Given the semi-structured nature of the data, particularly with tags being stored as an array, DynamoDB was selected over solutions like RDS, which are optimized for SQL data [5].

Cost: DynamoDB's managed service model offers cost-efficient scaling. It automatically scales throughput capacity to meet application demands, ensuring cost efficiency by allocating capacity units dynamically based on the number of requests [5].

Performance: DynamoDB provides consistent performance as data size grows. It automatically scales throughput capacity to handle increasing requests, maintaining low-latency read and write operations. This ensures that the performance remains stable regardless of data volume [5].

Security: By default, DynamoDB encrypts data in transit using encryption keys managed by AWS Key Management Service (KMS). The service also replicates data across multiple Availability Zones within AWS Regions, providing built-in redundancy and enhancing data durability and availability [5].

Scalability: DynamoDB is designed to scale both horizontally and vertically. It can add additional servers to increase capacity and throughput (horizontal scaling) and enhance the capability of individual servers (vertical scaling). This dual approach ensures that the database can handle varying loads efficiently [5].

Durability and Availability: DynamoDB ensures high durability and availability by automatically replicating data across multiple Availability Zones. This replication provides redundancy, allowing the service to maintain high availability even in the event of an infrastructure outage or failure. The synchronous replication across all Availability Zones ensures a low-latency read and write experience [5].

3.4 Networking and Content Delivery

3.4.1 Amazon VPC

AWS Virtual Private Cloud (VPC) is chosen to create a logically isolated network in the cloud, providing complete control over the resources inside it. VPC allows restricting access to resources from outside and permitting only trusted entities to interact with the VPC resources [10][11].

Cost: Amazon VPC doesn't incur any additional cost. The resources deployed in the VPC are billed [10][11].

Security: The VPC setup includes 2 public subnets and 2 private subnets, with one each in Availability Zones us-east-1a and us-east-2a. The Application Load Balancer is placed in the public subnets and is accessible from the internet, while the frontend applications deployed over ECS tasks are in private subnets, restricting public access. This setup ensures strong security by preventing direct traffic to the ECS tasks. Strong security policies are applied to the subnets to ensure that only specified security groups can invoke the resources within those subnets [10][11][12][13].

3.4.2 Application Load Balancer

The ClickClick application uses an Application Load Balancer (ALB) deployed in the public subnet to distribute traffic between different frontend containers in the private subnet. ALB serves as the front-facing component of the application, facilitating access to the frontend and other services [15].

Cost: ALB operates on a pay-per-use model, where charges are based on the amount of data processed and the number of Load Balancer Capacity Units (LCUs) used. This ensures cost efficiency by aligning costs with actual usage [15].

Performance: ALB improves efficiency by dynamically balancing traffic across containers, which helps avoid overloading any single instance. It also supports automatic scaling, which allows it to effectively manage varying traffic levels and sustain peak performance [15].

Scalability: ALB supports auto-scaling, which allows it to efficiently manage varying loads by scaling resources up or down as needed. This ensures high availability and reliability, with the ability to handle traffic spikes and varying user demands seamlessly [15].

Health Monitoring: ALB manages health checks for each target group independently, providing improved resiliency by ensuring that traffic is only routed to healthy instances. Detailed logs

provided by ALB offer better insights into traffic patterns and system performance compared to the Classic ELB [15].

3.4.3 Amazon API Gateway

API Gateway is utilized to map the REST API endpoints to the correct AWS Lambda functions. It offers seamless integration with Lambda, simplifying the process of connecting HTTP requests to backend services.

Cost: API Gateway operates on a pay-per-use model, charging based on the number of API calls made and the amount of data transferred. This ensures cost efficiency by aligning expenses with actual usage [14].

Performance: API Gateway helps manage traffic to backend systems by allowing the setup of throttling rules based on the number of requests per second for each HTTP method in the APIs, preventing overload and ensuring consistent performance [14].

Security: API Gateway provides built-in authorizers, such as AWS Cognito, to protect endpoints from unauthorized access. This integration ensures that only authenticated users can access specific API endpoints, enhancing the security of the application [14].

Scalability: API Gateway supports automatic scaling to handle varying levels of API traffic. It can scale up to accommodate high traffic loads and scale down when demand decreases, ensuring consistent performance and cost efficiency [14].

3.5 Security

3.5.1 AWS Cognito

AWS Cognito is used to store users in UserPools and provide an access control mechanism using JWT tokens. When a user registers on the portal, they are added to the user pool, and upon login, their email and password are validated in the user pool, returning a JWT token [7].

Cost: Cognito offers cost efficiency by automatically scaling to handle millions of users, ensuring that resources are only used as needed without manual intervention [7].

Security: Cognito enhances security by offering built-in features like multi-factor authentication, encryption at rest, and encryption in transit. It also integrates with API Gateway, adding a Cognito authorizer to endpoints such as **getAllPosts** and **subscribeToTags**, ensuring these endpoints are protected and only accessible to authenticated users [7].

Scalability: Cognito scales automatically to handle a large number of users, ensuring that the application can support a growing user base without performance degradation [7].

3.6 Machine Learning

3.6.1 Amazon Rekognition

Amazon Rekognition is used to automatically generate tags for images uploaded to the system. This service is preferred over training an in-house model due to its managed nature and ease of integration with AWS Lambda [8].

Cost: Rekognition operates on a pay-as-you-use model, eliminating the need for significant upfront investment in computing power and large datasets required for training custom models [8].

Performance: Rekognition continuously updates its dataset, providing better and more accurate results compared to what could be achieved with a custom model. This enhances the overall performance of the tagging feature [8].

Security: As a managed service, Rekognition benefits from AWS's robust security infrastructure, ensuring data protection without additional management overhead [8].

Scalability: As a fully managed service, Rekognition scales automatically based on usage, ensuring that it can handle varying volumes of image processing requests efficiently [8].

3.7 Management and Governance

3.7.1 CloudFormation

AWS CloudFormation is used to automate the creation and deployment of the entire infrastructure using YAML scripts. This automation reduces manual errors and improves resource utilization [9].

Cost: CloudFormation helps keep costs low by automating the provisioning and management of resources, reducing the need for manual intervention and associated costs [9].

Performance: Automated deployment and monitoring of resources enhance system reliability and ensure consistent performance across environments [9].

Scalability: CloudFormation enables the deployment of infrastructure stacks across multiple global regions in an automated way, supporting scalable and consistent infrastructure management [9].

3.8 Other Services

3.8.1 Amazon SNS

Amazon SNS is used to send email notifications to users when new images are uploaded with tags they have subscribed to. It integrates seamlessly with AWS Lambda [17].

Cost: SNS offers a cost-efficient messaging service by ensuring messages are only sent when necessary and retrying delivery if endpoints are unavailable, minimizing waste [17].

Performance: SNS provides message durability by storing published messages across multiple, geographically-separated servers and data centers, ensuring reliable message delivery [17].

Scalability: SNS scales automatically to handle varying volumes of messages, ensuring efficient and timely delivery of notifications to users [17].

4. AWS Well Architected Framework

The AWS Well-Architected Framework consists of six pillars: Operational Excellence, Security, Reliability, Performance Efficiency, Cost Optimization, and Sustainability. The ClickClick application adheres to these principles through various implementations and services. Below is an explanation of how each pillar is satisfied in ClickClick:

4.1 Operational Excellence

Automation: The infrastructure is deployed using AWS CloudFormation scripts, which helps reduce manual errors, improve resource utilization, and increase system reliability. This automated approach supports the principle of performing operations as code [18][19].

Continuous Improvement: The use of monitoring and logging tools, such as detailed logs from the Application Load Balancer (ALB), provides insights into system performance and health, enabling continuous improvement [18][19].

4.2 Security

Access Control: AWS Cognito is used to manage user authentication and provide secure access to API endpoints via JWT tokens. This aligns with the principles of implementing a strong identity foundation and applying security at all layers [18][19].

Data Protection: Data in transit is encrypted using AWS KMS, and data at rest is secured through AWS services like S3 and ECR, which use server-side encryption. This ensures that data is protected in transit and at rest [18][19].

Endpoint Protection: API Gateway integrates with Cognito to protect API endpoints, ensuring that only authorized users can access specific services [18][19].

4.3 Reliability

Auto Scaling: Amazon ECS with Fargate and Auto Scaling Group ensures that the number of container instances adjusts automatically based on demand, maintaining application reliability during traffic spikes [18][19].

Data Replication: Amazon DynamoDB automatically replicates data across multiple Availability Zones, providing high availability and durability [18][19].

Health Checks: ALB performs health checks on target groups, ensuring traffic is only routed to healthy instances, which supports automatic recovery from failure [18][19].

4.4 Performance Efficiency

Serverless Architecture: AWS Lambda is used for backend functions, which scales automatically based on request volume, ensuring efficient use of computing resources [18][19].

Load Balancing: ALB distributes incoming traffic across multiple containers, preventing any single instance from becoming a bottleneck and maintaining optimal performance [18][19].

Image Processing: Amazon Rekognition is used for image tagging, leveraging AWS's advanced machine learning capabilities to provide efficient and scalable image processing [18][19].

4.5 Cost Optimization

Pay-as-You-Go: Services like Lambda, S3, and API Gateway operate on a pay-per-use model, ensuring that costs are directly tied to usage, preventing over-provisioning [18][19].

Auto Scaling: Both ECS with Fargate and DynamoDB auto-scale based on demand, which helps in cost savings by scaling resources only when necessary [18][19].

Lifecycle Policies: S3 lifecycle policies move infrequently accessed data to more cost-effective storage classes, optimizing storage costs [18][19].

4.6 Sustainability

Efficient Resource Usage: The use of serverless architectures, such as AWS Lambda, minimizes idle resource consumption and energy use, contributing to sustainability [18][19].

Managed Services: Utilizing AWS managed services like DynamoDB, ECS, and Rekognition allows ClickClick to benefit from AWS's optimized resource management and sustainability practices. These services are managed in a serverless and cost-optimized manner, reducing the environmental footprint by efficiently using resources [18][19].

Scalable Infrastructure: Auto-scaling capabilities across various services ensure that only the necessary resources are used, reducing waste and enhancing overall sustainability [18][19].

By aligning with the AWS Well-Architected Framework, ClickClick ensures a robust, secure, and cost-effective architecture that can scale and adapt to changing requirements, while maintaining high performance and operational excellence.

5. Screenshots

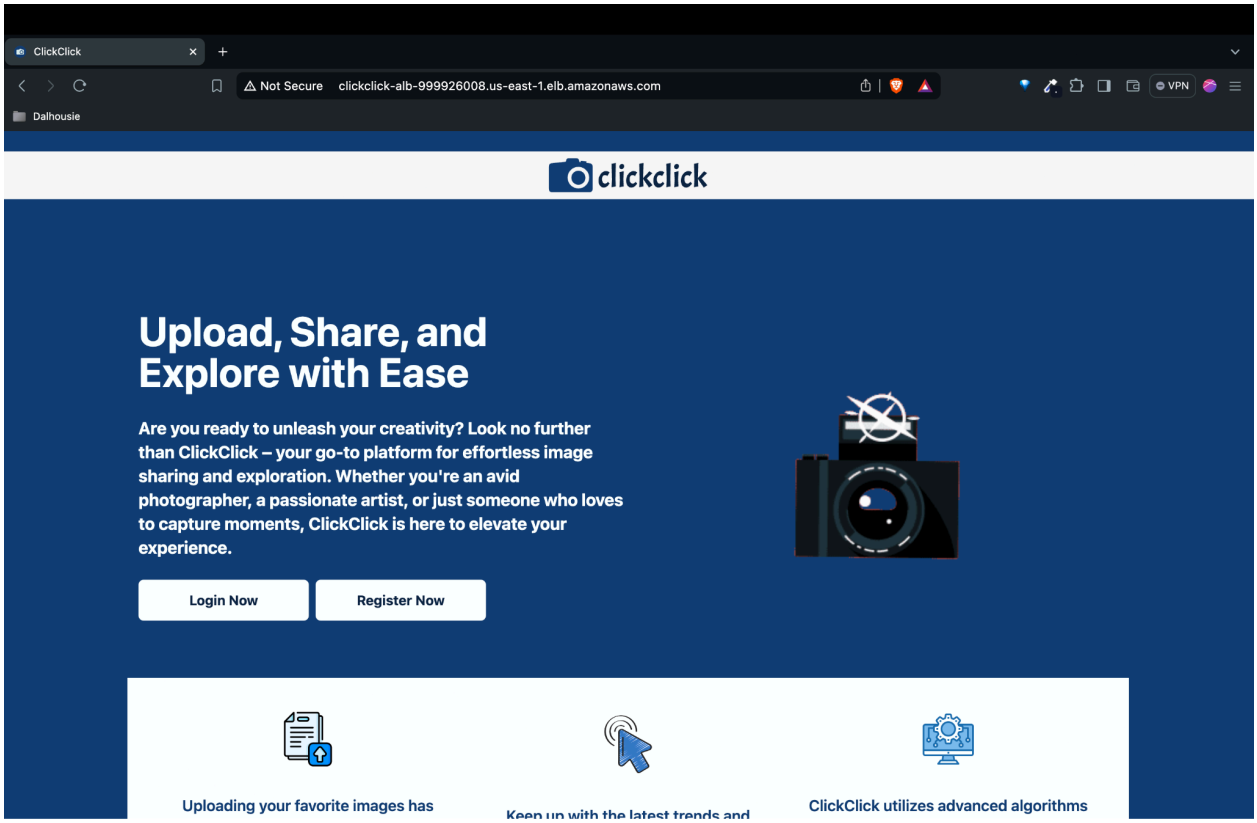

Figure 2. Landing page of the ClickClick Application

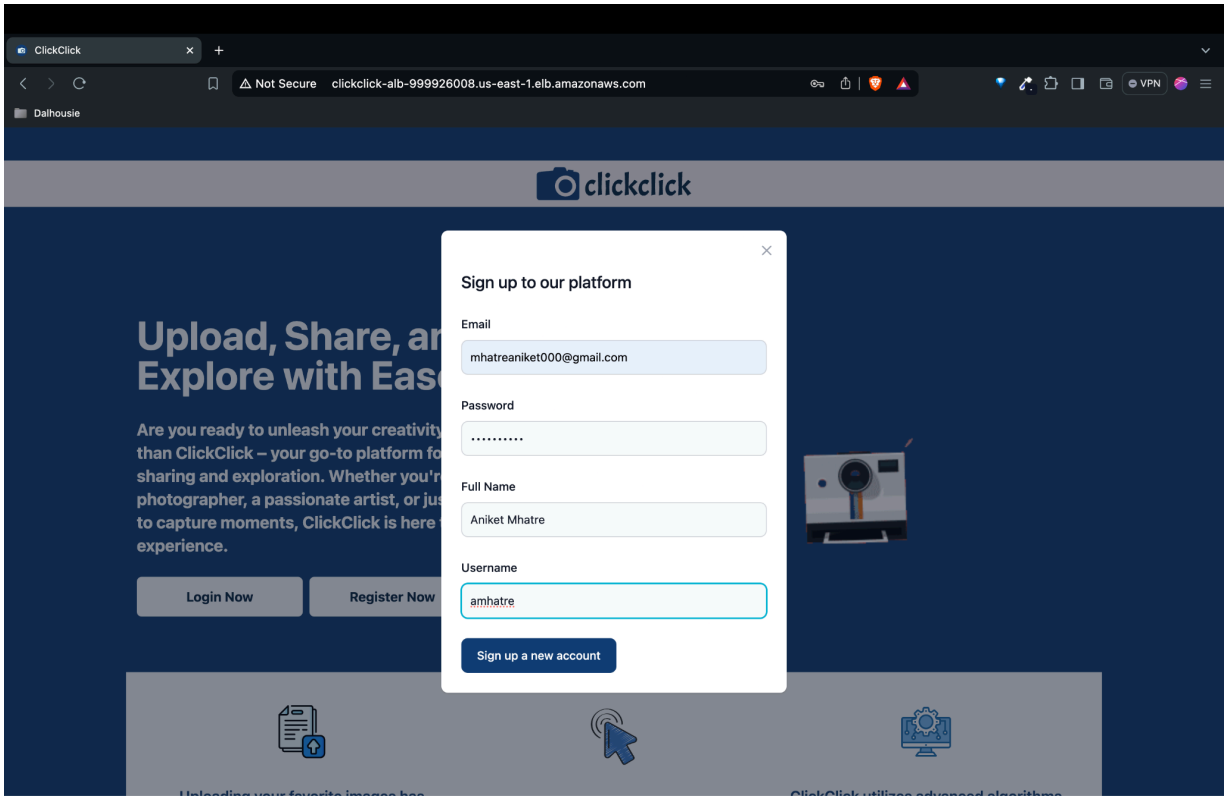

Figure 3. Registration modal of the ClickClick application.

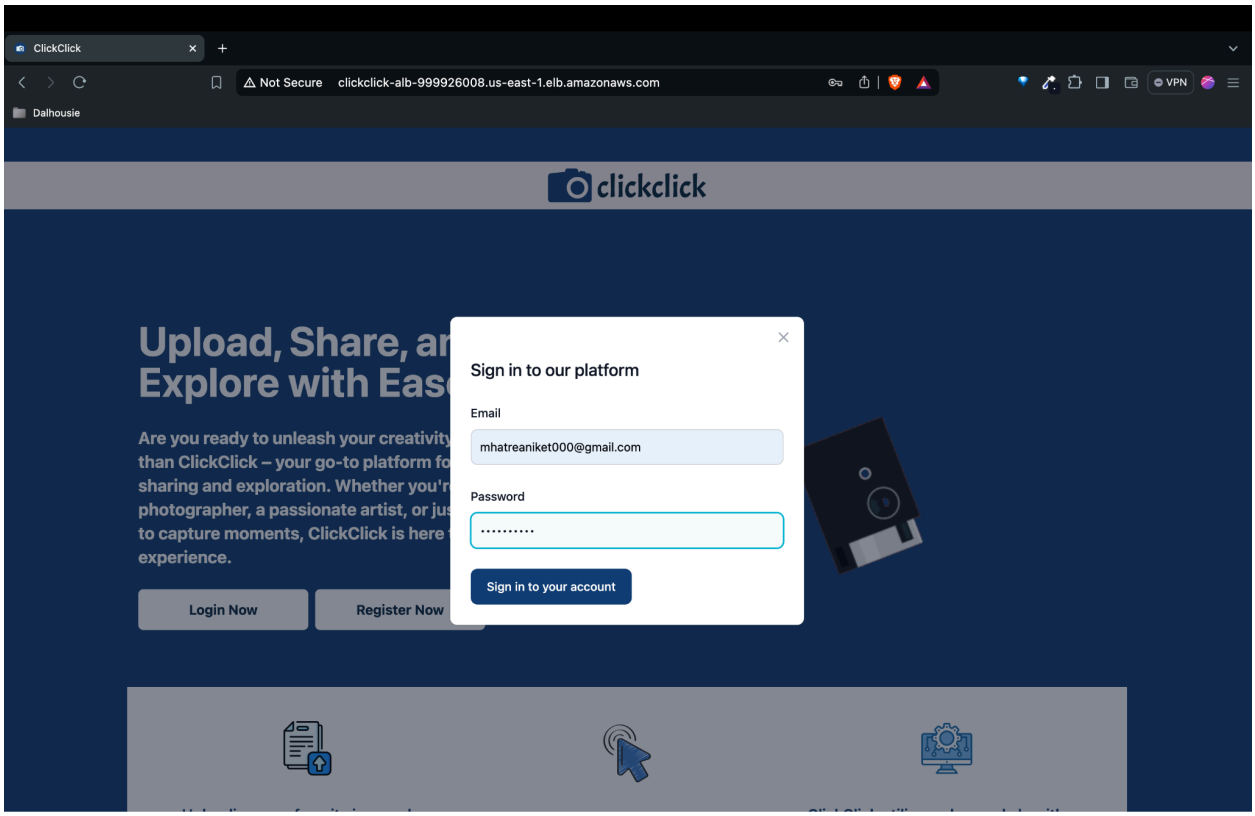

Figure 4. Login modal of the ClickClick application.

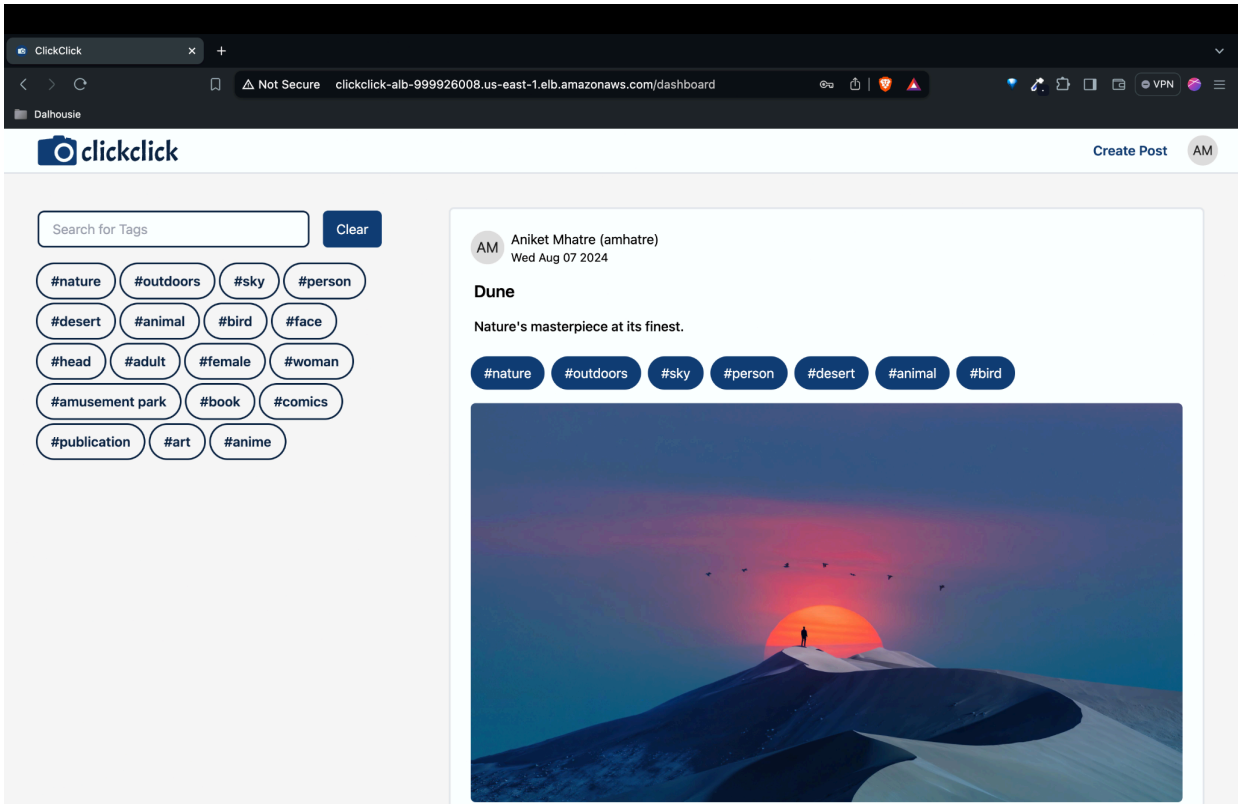

Figure 5. Dashboard of the ClickClick application.

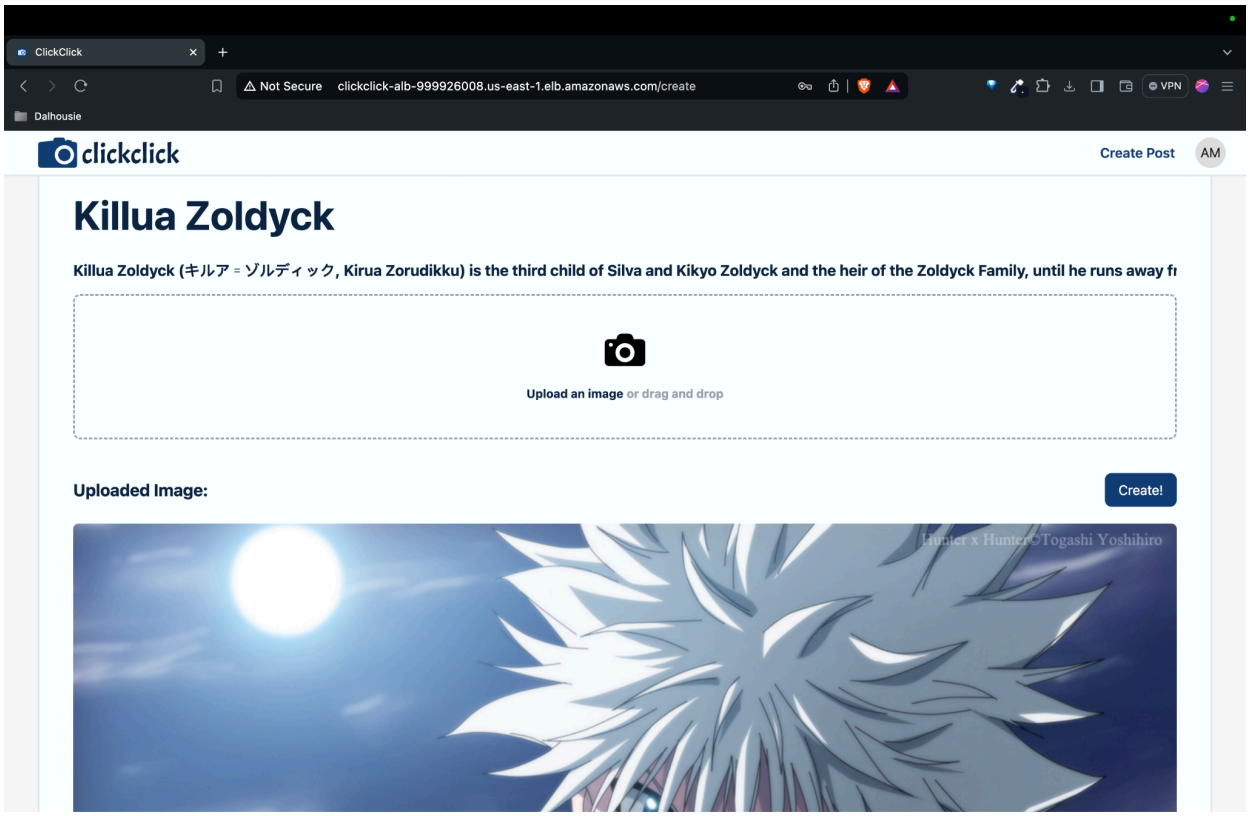

Figure 6. Create a new post in the application.

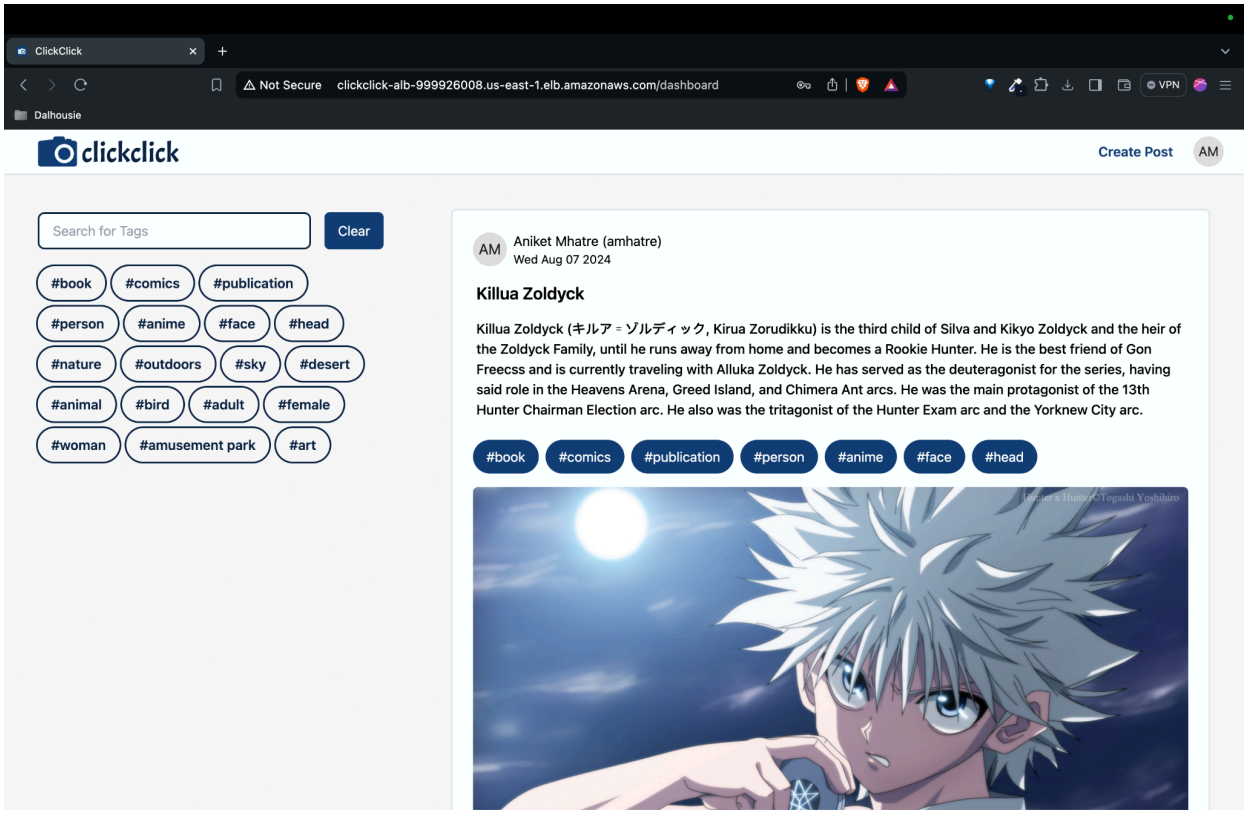

Figure 7. New post created with tags generated.

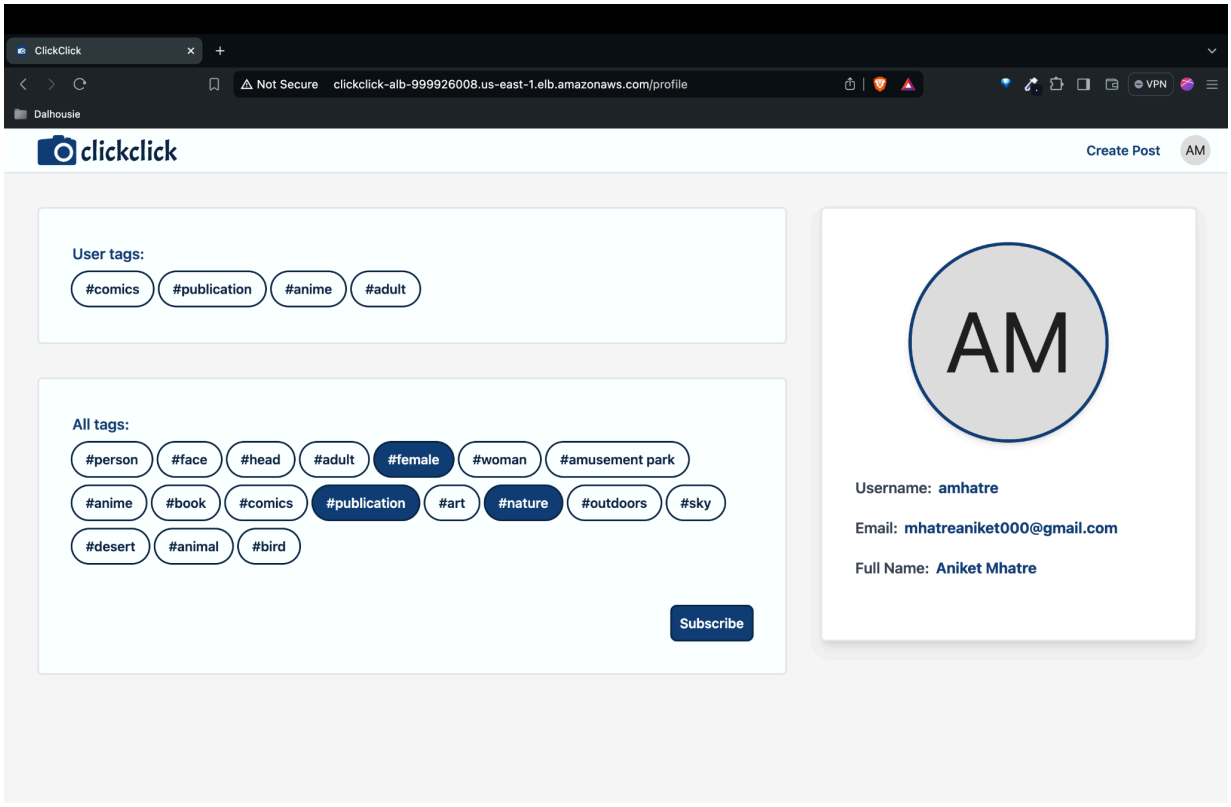

Figure 8. User profile showing User subscribed tags and selected tags for subscription.

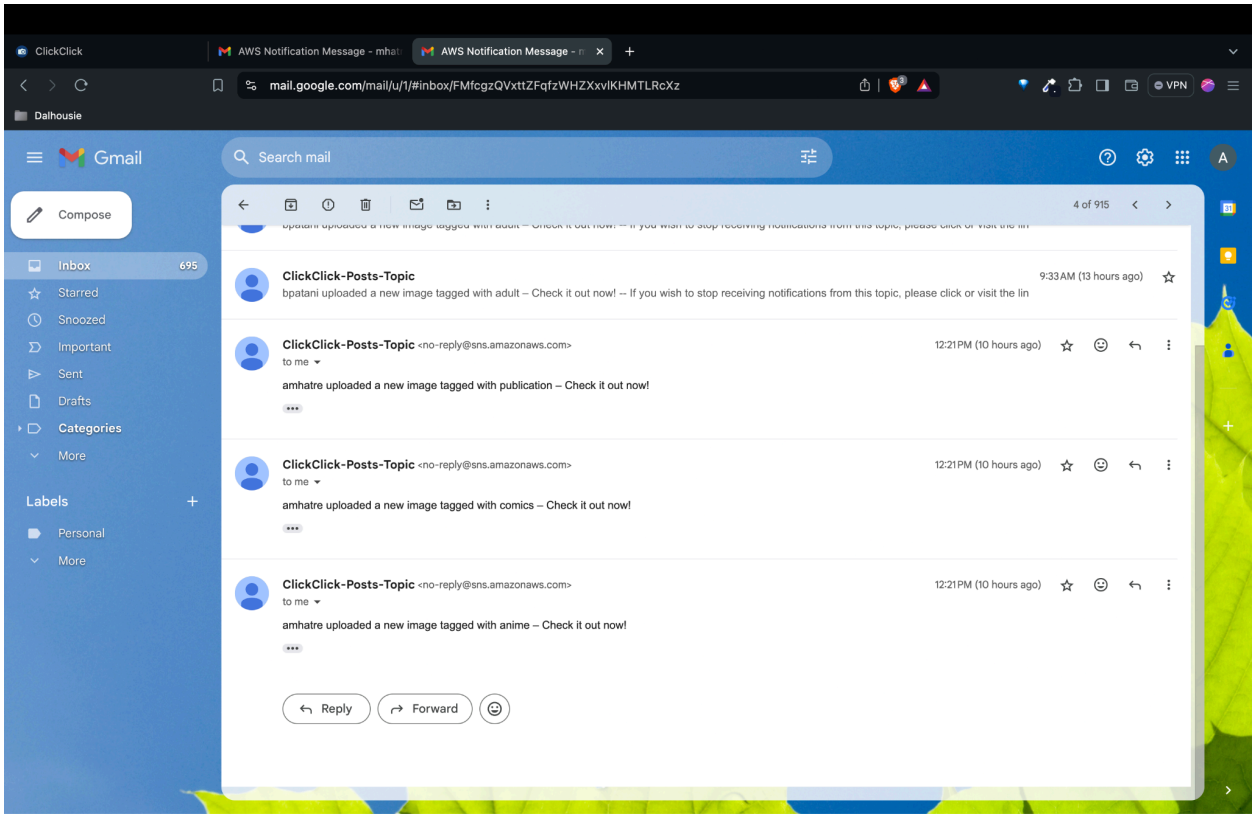

Figure 9. A user receives emails if any image is uploaded with tags they have subscribed to.

References

[1] Pető, G., “AWS Container Services Unveiled: EC2 on ECS vs. Fargate Explained,” *Medium*, Feb. 10, 2023. [Online]. Available: https://medium.com/@grpeto/aws-container-services-unveiled-ec2-on-ecs-vs-fargate-explained-dee1cd5c1563. [Accessed: Jun. 16, 2024].

[2] Guruge, G., “The Good, Bad & the Ugly of AWS Lambda,” *Medium*, Mar. 15, 2023. [Online]. Available: https://medium.com/@geethika.guruge/the-good-bad-the-ugly-of-aws-lambda-e06df5930a9. [Accessed: Jun. 16, 2024].

[3] “Amazon ECR vs. Docker Hub vs. GitHub Container Registry,” *cloudonaut*, Jun. 10, 2023. [Online]. Available: https://cloudonaut.io/amazon-ecr-vs-docker-hub-vs-github-container-registry/. [Accessed: Jun. 16, 2024].

[4] “Comparing Amazon ECS Launch Types: EC2 vs. Fargate,” *Lumigo Blog*, Apr. 18, 2023. [Online]. Available: https://lumigo.io/blog/comparing-amazon-ecs-launch-types-ec2-vs-fargate/. [Accessed: Jun. 16, 2024].

[5] “Amazon RDS vs. DynamoDB: 12 Differences,” *QA Blog*, May 25, 2023. [Online]. Available: https://www.qa.com/resources/blog/amazon-rds-vs-dynamodb-12-differences/. [Accessed: Jun. 20, 2024].

[6] “Setup AWS S3 Bucket for Storing Files/Images,” *Medium*, Jul. 15, 2023. [Online]. Available: https://medium.com/@thilinawaks/setup-aws-s3-bucket-for-storing-files-images-90cbcc7558d7#:~:text=S3%20offers%20several%20features%2C%20including,help%20you%20secure%20your%20data. [Accessed: Jun. 30, 2024].

[7] “Pros and Cons of Using Amazon Cognito for User Authentication,” *Gavant Blog*, Mar. 22, 2023. [Online]. Available: https://www.gavant.com/library/pros-and-cons-of-using-amazon-cognito-for-user-authentication. [Accessed: Jul. 05, 2024].

[8] “Amazon Rekognition Custom Labels Features,” *AWS*, [Online]. Available: https://aws.amazon.com/rekognition/custom-labels-features/?nc=sn&loc=3&dn=4. [Accessed: Jul. 15, 2024].

[9] "AWS CloudFormation Overview," *Forbytes Blog*, Apr. 14, 2023. [Online]. Available: https://forbytes.com/blog/aws-cloudformation-overview/. [Accessed: Aug. 01, 2024].

[10] "AWS VPC Security Best Practices," *Hyperglance Blog*, May 10, 2023. [Online]. Available: https://www.hyperglance.com/blog/aws-vpc-security-best-practices/. [Accessed: Jun. 2, 2024].

[11] Amazon Web Services, Inc., "VPC Security Best Practices," *AWS Documentation*, May 18, 2023. [Online]. Available: https://docs.aws.amazon.com/vpc/latest/userguide/vpc-security-best-practices.html. [Accessed: Jun. 12, 2024].

[12] "How to Secure Your AWS Infrastructure," *Medium*, Jun. 12, 2023. [Online]. Available: https://medium.com/codex/how-to-secure-your-aws-infrastructure-e11275c2936e. [Accessed: Jun. 12, 2024].

[13] "Infrastructure Protection on AWS for Beginners," *AWS Community*, Jan. 15, 2024. [Online]. Available: https://community.aws/content/2egoT8yUjjzxYQrkeUDzZev4GWG/infrastructure-protection-on-aws-for-beginners?lang=en. [Accessed: Jun. 12, 2024].

[14] Amazon Web Services, Inc., "Amazon API Gateway Features," *AWS*, [Online]. Available: https://aws.amazon.com/api-gateway/features/. [Accessed: Jul. 14, 2024].

[15] Amazon Web Services, Inc., "Application Load Balancer," *AWS*, [Online]. Available: https://aws.amazon.com/elasticloadbalancing/application-load-balancer/?nc=sn&loc=2&dn=2. [Accessed: Jul. 26, 2024].

[16] Amazon Web Services, Inc., "Amazon S3 Features," *AWS*, [Online]. Available: https://aws.amazon.com/s3/features/. [Accessed: Jun. 30, 2024].

[17] Amazon Web Services, Inc., "Amazon SNS Features," *AWS*, [Online]. Available: https://aws.amazon.com/sns/features/. [Accessed: Jul 03, 2024].

[18] Piyush Jain, "AWS Well-Architected Framework 101," *Medium*, Feb. 5, 2023. [Online]. Available: https://piyushj02.medium.com/aws-well-architected-framework-1o1-1c9521403046. [Accessed: Jun. 05, 2024].

[19] Amazon Web Services, Inc., "The 6 Pillars of the AWS Well-Architected Framework," *AWS Partner Network (APN) Blog*, May 10, 2023. [Online]. Available: https://aws.amazon.com/blogs/apn/the-6-pillars-of-the-aws-well-architected-framework/. [Accessed: Jun. 05, 2024].