LSDE Coursework Part 1 – AWS Architecture for ArtAI

Rohan Anthony (2704500)

Overview

The task is to design an AWS architecture for "ArtAI", a public web service that allows users to upload images and receive AI-generated variations. The architecture must deliver low-latency global access, secure and private model access, automatic backup and versioning of images, and an automated alert if daily requests exceed 10,000. We focus on achieving high availability, reliability, and scalability by integrating AWS services.

Architecture and AWS Services

This architecture necessitates integrating several AWS services that collectively deliver performance, security, scalability, and cost efficiency. The key services and their roles are listed below, followed by an explanation of how they interact within the architecture.

1. Amazon Route 53 – Latency-based DNS routing for worldwide access.
2. Amazon CloudFront – Global CDN for caching and HTTPS delivery. Delivers high performance through edge location caching.
3. Amazon S3 – Hosts static frontend website and stores all uploaded and generated images with versioning, replication and lifecycle management.
4. AWS Lambda – Serverless compute for backend processing.
5. Amazon WAF – Web application firewall in front of CloudFront.
6. Amazon Certificate Manager – Managed TLS certificates.
7. Amazon API Gateway – Secure API entry point for backend logic.
8. Amazon SageMaker Endpoint – Hosts the pre-trained model inside a private VPC. SageMaker is chosen over EC2 or Lambda because it provides a fully managed auto-scaling solution for real-time inference, large models, automatic deployment and health checks while remaining private and highly performant.
9. Amazon SQS – Queue to decouple uploads from inference jobs.
10. Amazon DynamoDB – Metadata store with Point in Time Recovery.
11. Amazon CloudWatch – Metrics and alarms.
12. Amazon SNS – Sends email when alarm is triggered.
13. AWS IAM – Access control.
14. AWS KMS – Encryption of data at rest and in transit.

Cost-efficiency is achieved through Lambda's pay-per-execution model, which eliminates idle server costs. S3 lifecycle rules move older data to cheaper storage. SageMaker auto-scaling reduces costs during low-traffic periods.

The figure on page 2 illustrates the full AWS architecture.

Service Interaction and Workflow

When a user accesses our public URL (art.ai for example), **Route 53** uses latency-based routing to direct them to the nearest CloudFront edge location. **CloudFront** serves the cached static frontend from **S3** and forwards dynamic API requests to the backend while **WAF** filters malicious traffic. AWS **Certificate Manager** provides TLS certificates, so traffic can be securely delivered over HTTPS, supporting low-latency global access. Backend operations are handled by the **API Gateway**, which invokes our **Lambda** functions. We have a Lambda function that generates a pre-signed S3 URL so the

user can upload their image directly to S3. Each new S3 object pushes an event onto an **SQS** queue, decoupling uploads from processing. A worker Lambda function consumes messages from SQS and calls a **SageMaker** endpoint where our model is hosted inside a private VPC. The processed output image is saved in another S3 bucket, and metadata is written to DynamoDB for point-in-time recovery. IAM enforces least-privilege access and KMS encrypts data at rest and in transit.

Data Backup and Recovery

When an image is uploaded to S3, versioning automatically creates a new version of the object, ensuring the previous versions can always be restored. Cross-Region Replication provides an additional layer of resilience by replicating objects to a secondary AWS region. Lifecycle rules move older versions to lower cost storage to balance cost and recoverability. Metadata stored in DynamoDB is protected through Point-in-Time recovery. These combined mechanisms ensure all image data is automatically backed up.

Monitoring and Alerts

Amazon **CloudWatch** collects metrics from API Gateway and Lambda, including the total number of API requests received. A CloudWatch alarm is configured to monitor the number of requests over any 24-hour period, and if the total exceeds 10,000 requests, the alarm is triggered, and a notification is published to an Amazon **SNS** topic, which sends an automated email alert.

Conclusion

This architecture combines global content delivery, secure private model hosting, automated backups, and robust monitoring to meet the requirements of the ArtAI service. By leveraging these AWS services, the design delivers high availability, security, and cost-efficient scalability.

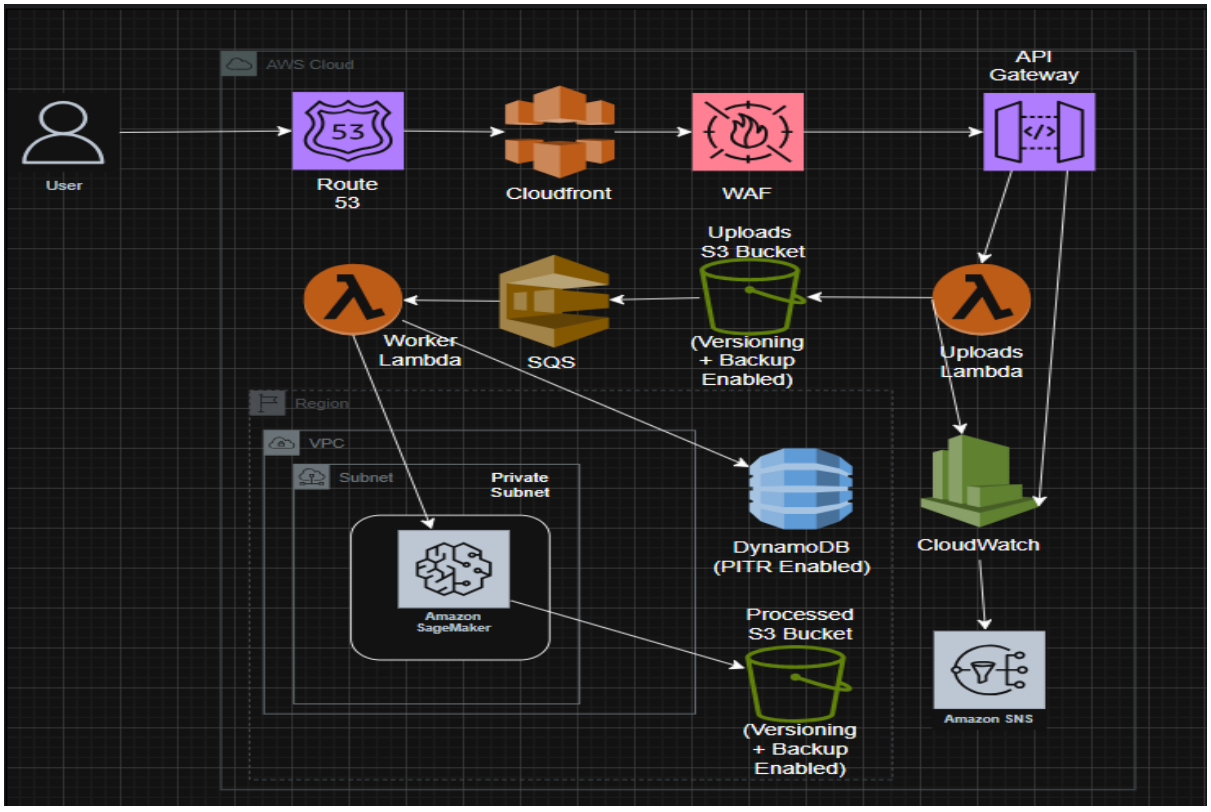

Figure 1: ArtAI AWS Architecture