(1)Region Selection: Focus on regions closest to Canada initially, with plans to expand to other regions as traffic increases.

(2)Bucket Size Estimation: Estimate the size of the S3 bucket based on the user list and potential growth.

(3)Latency Goals: Aim for inference latency between 5 to 10 seconds, balancing cost and performance.

(4)Log Management: Define the preferred log format (JSON) and ensure proper logging practices

(5)Cost Optimization: Implement cost-saving measures such as using async inference or cheaper OpenAI models.

(6A) Set Up SageMaker Studio & S3: Launch SageMaker Studio, create a role with S3 access, and upload blip2-endpoint.tar.gz to the S3 bucket.

(6B) Version Control and CI/CD: Set up GitHub and GitHub Actions for version control and CI/CD automation for both backend and SageMaker deployment.

(6C)Authentication Integration: Integrate Superbase authentication with AWS services, ensuring secure API calls.

(6D) Data Encryption: Ensure data encryption for API calls and model pulling to SageMaker

(6E) Webhook Integration: Use webhooks for asynchronous communication between Superbase and SageMaker.

(6F) Integration of Supabase Edge with AWS CloudWatch, with focus on asynchronous handling of logs and the impact on performance.

(7)API Gateway Integration: Create an HTTP API with ALB as integration, route POST requests to /analyze, and optionally connect Route 53 + ACM for HTTPS domain

(8)Deploy to ECS: Create an ECS cluster and Fargate service, attach an ALB, and open port 8080.

(9) Build FastAPI App: Develop the FastAPI app to handle file uploads, call SageMaker, format OpenAI prompts, and return structured JSON responses.

(10) Dockerize App: Create a Dockerfile and requirements.txt, build, tag, and push the Docker image to Amazon ECR.

(11) Implement Core Logic: Develop the core logic for BLIP2 execution and product report API in app.py.

(12)Deploy Model: Use sagemaker.pytorch.PyTorchModel to deploy the model to the endpoint blip2-endpoint

(13) Lifecycle Management: Implement a lifecycle policy to delete images from the S3 bucket after a certain period to save costs.

(14) Monitoring and Alerts: Set up monitoring and alerting for latency issues using PostHog