Amazon Product Review Analysis Service

- *Datacenter Scale Computing - Fall 2024*

Group 14
Trapti Damodar Balgi
Nidhi Choudhary

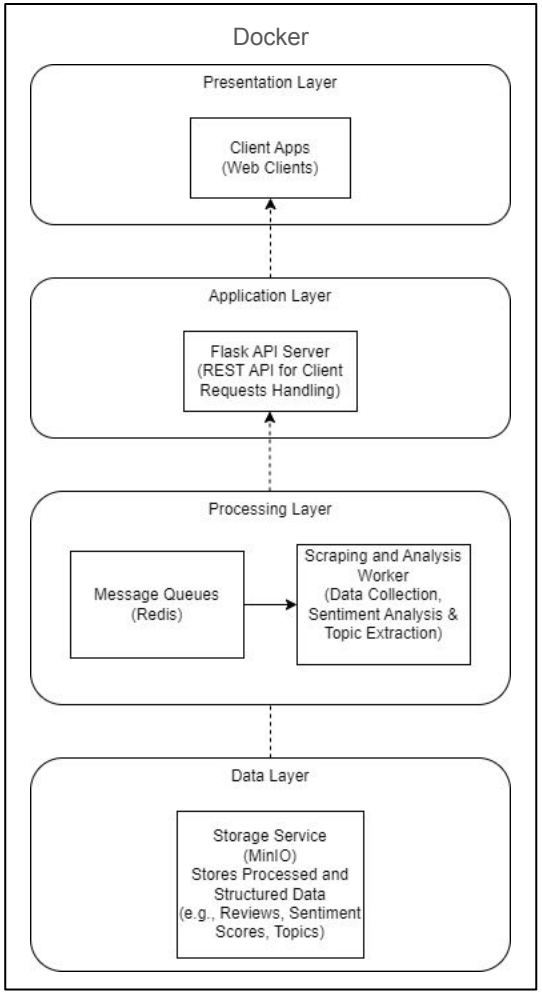
Project Goals

- Develop a service to collect and analyze Amazon product reviews.
- Extract product names and summarize customer feedback.
- Perform sentiment analysis to assess customer satisfaction (% positive, negative, neutral).
- Identify and store key topics from reviews in a data storage system to enhance decision-making for consumers and businesses.

Software Components

1. **REST API:** Flask-based API for user interactions (submit URLs, retrieve insights).
2. **Message Queues:** Redis to distribute tasks across worker nodes.
3. **Storage:** MinIO for storing processed review data.
4. **Text Analysis:** Pre-trained models for sentiment analysis, topic modeling, and summarization. *[Pre-trained models like "facebook/bart-large-cnn", "joeddav/distilbert-base-uncased-go-emotions-student", and Latent Dirichlet Allocation (LDA) models for summarization, sentiment analysis, and topic modeling, respectively.]*
5. **Docker:** Scalable, containerized application.

Architectural Diagram



Interaction between different Software Components

Presentation Layer:

- Users interact with the service through a web client interface, submitting Amazon product URLs and viewing analysis results, which are processed and delivered via the Flask REST API.

Application Layer:

- Flask API processes user requests and forwards tasks to Redis queues.

Processing Layer:

- Redis queues tasks for the worker.
- Workers handle scraping, sentiment analysis, topic extraction, and summarization using pre-trained models.

Data Layer:

- MinIO stores processed data (reviews, sentiment scores, topics).

Deployment:

- Docker containerizes the application, ensuring consistent environments and simplifying the deployment process.

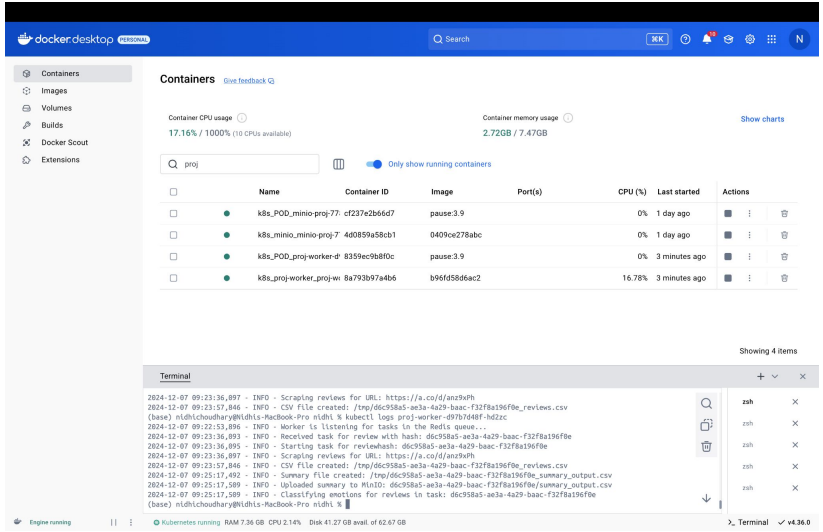
Debugging and Testing

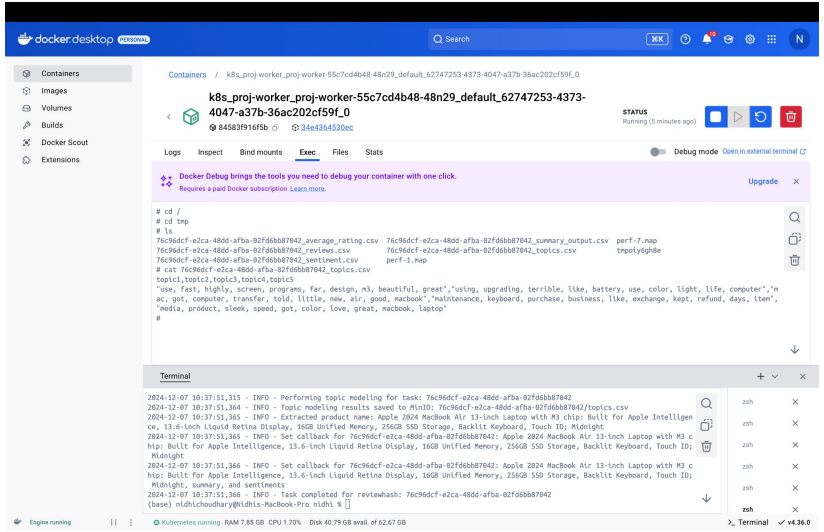
Developed and tested each component separately to ensure functionality:

- Scraping: Validated correct data extraction from Amazon product pages.
- UI and REST API: Tested API endpoints and UI for seamless interaction.
- Summarization: Verified proper summarization of reviews using the pre-trained model.
- Product Name Extraction: Checked accuracy in extracting product names.
- Topic Extraction and Sentiment Analysis: Tested accuracy of topic modeling and sentiment classification.
- Callback Feature: Ensured correct callback responses.

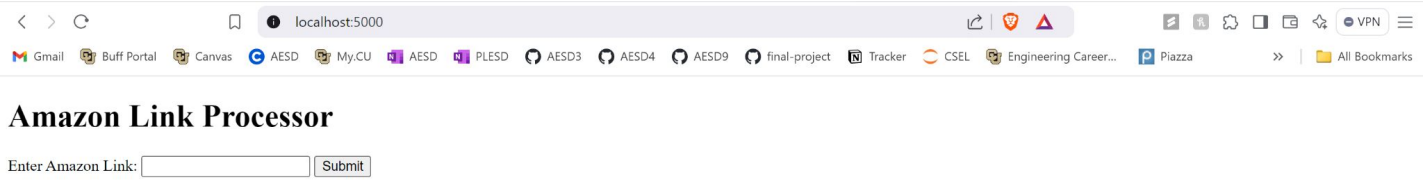
After individual tests, we integrated the components, checked the logs for errors, and verified the output through MinIO UI for proper data storage.

Debugging and Testing Snapshots





Working System - UI



Working System - Results

Amazon Link Processor

Enter Amazon Link: [ ] Submit
Status: completed

Callback Details

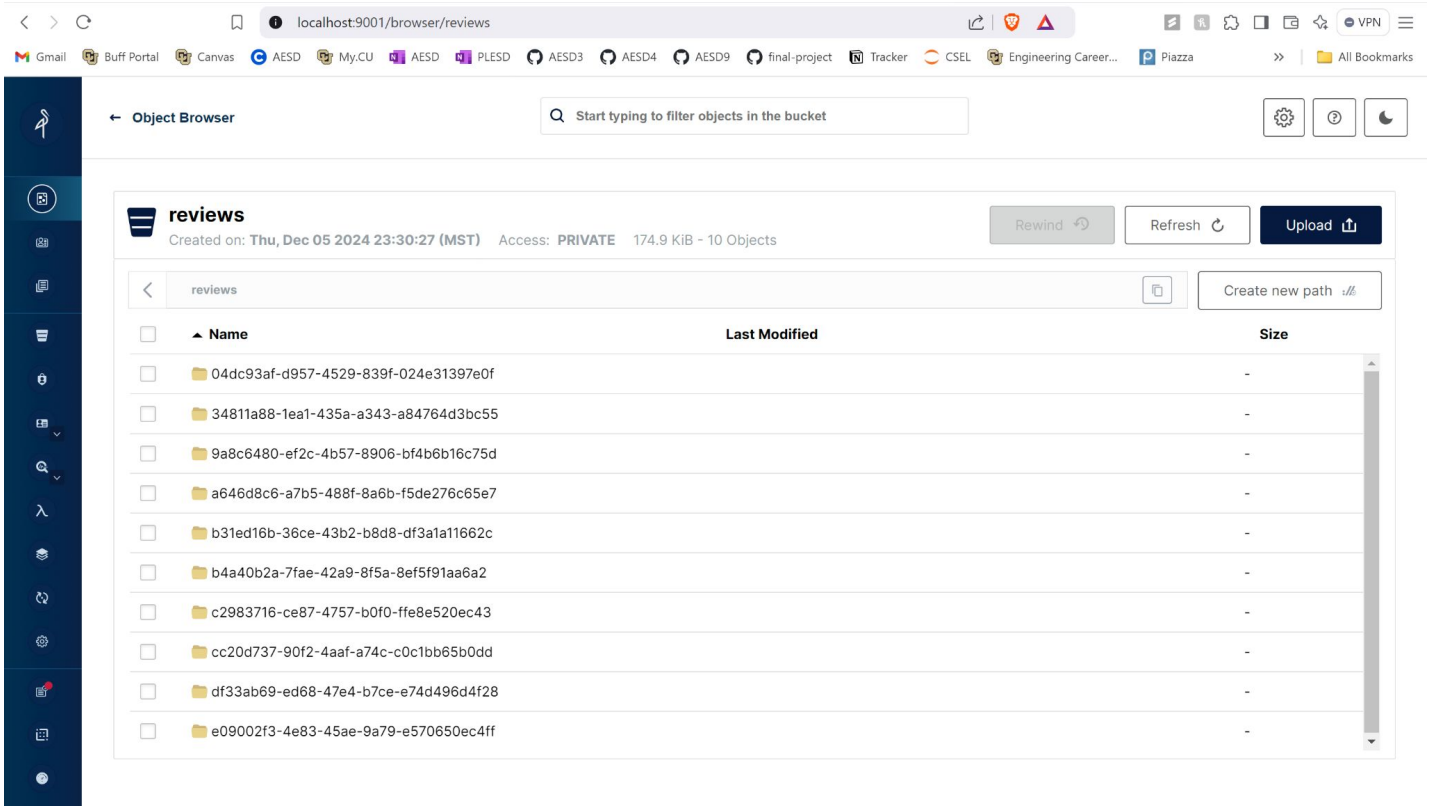
Product Name: Caseative for iPhone 14 Pro Case, Solid Color Curly Wave Frame Soft Compatible with iPhone Case (Green Blue,iPhone 14 Pro)

Summary: Love it but with this particular color it gets dirty quick! Would definitely buy again just in another color. Got so many compliments so that was a plus! Love how it protected my phone, wasn't bulky, & was ascetically pleasing! Also, you can't beat the price either!

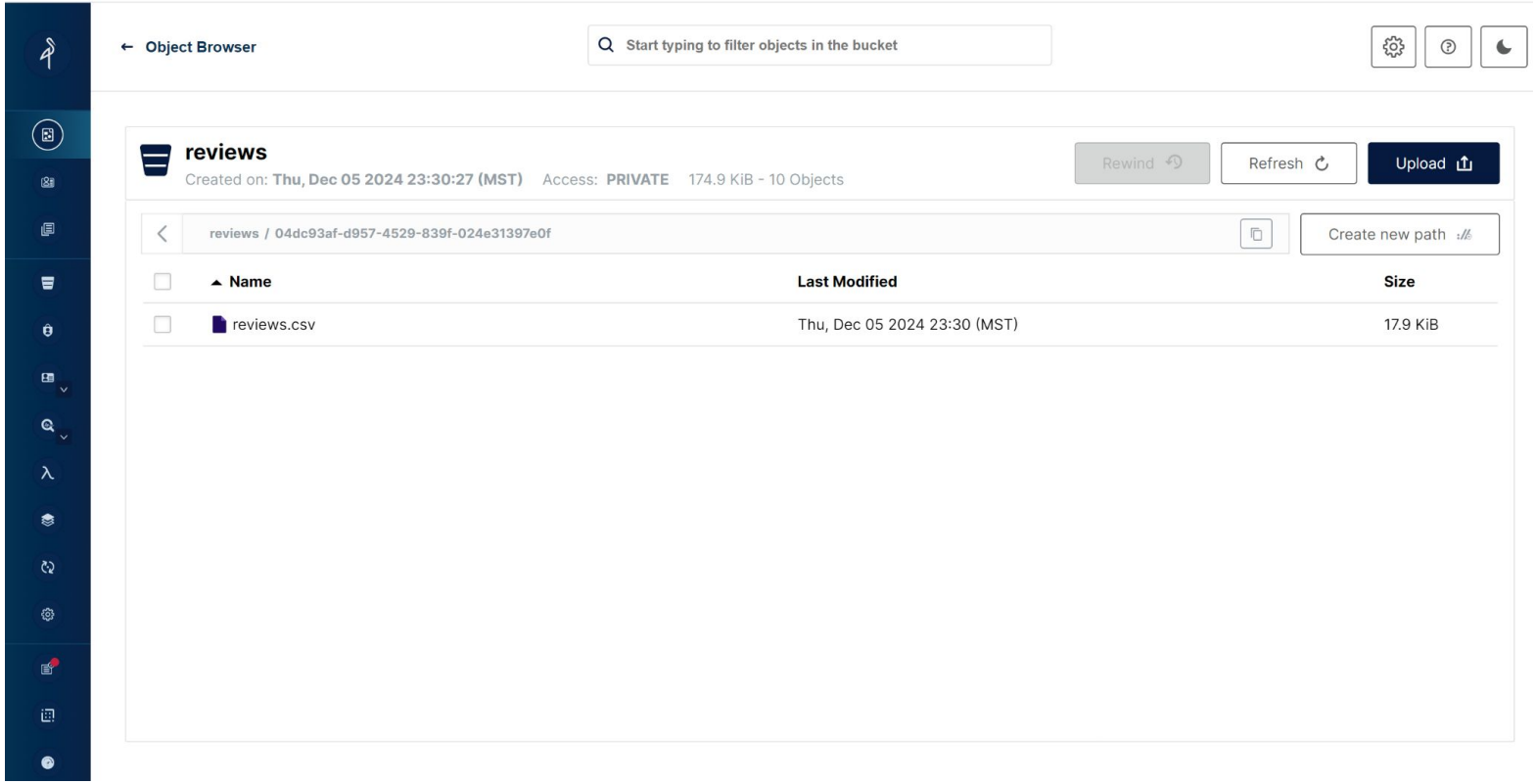
Sentiments:

- Positive: 43.67010703897662%
- Neutral: 16.63010647495578%
- Negative: 39.6997864860676%

Working System - Storage (Processed Data)



Working System - Storage (Processed Data)



Workload Handling and Bottlenecks

Workload Handling:

The system can currently handle light workloads, processing a few hundred simple operations per second. This can be scaled as necessary.

Bottlenecks:

1. **Rate Limiting from Amazon:** Restrictions on the frequency and volume of requests to prevent overloading their servers. Currently, we extract reviews only from the first 5 pages to stay within these limits
   Future Work: Implementing request throttling and rotating proxy servers to bypass rate limits.
2. **Fake Reviews:** Finding and removing unreliable or fake reviews.
   Future Work: Using advanced machine learning models to improve the detection and removal of fake reviews.
3. **Models for Analysis:** Ensuring accurate and efficient text processing.
   OpenAI API: Limited by usage caps, requiring a paid version for continued analysis.
   Future Work: Improve performance with fine-tuning and faster models for real-time processing.

DEMO

References

- https://www.geeksforgeeks.org/web-scraping-amazon-customer-reviews/
- https://arxiv.org/abs/1810.04805
- https://arxiv.org/abs/1910.13461
- https://huggingface.co/docs/transformers/index
- https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0
- https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2
- https://realpython.com/python-redis/
- https://min.io/docs/kes/
- Class notes, slides, assignments