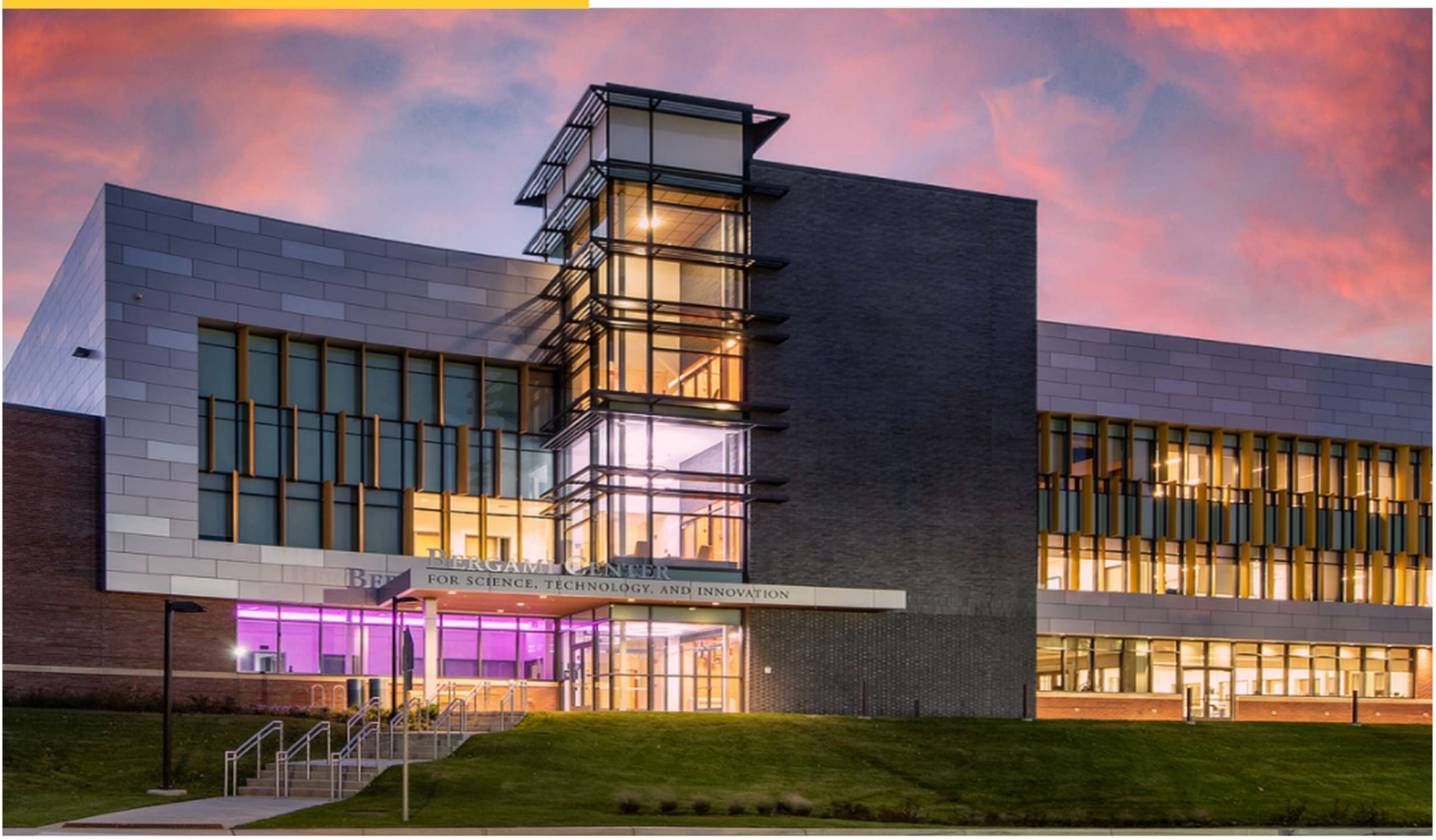
University of New Haven

REPORT

University of New Haven

TAGLIATELA COLLEGE OF ENGINEERING



DSCI 6007-03: TEAM 03
SEMESTER 2
Submitted on: 29th April 2025

CONTENTS

Project Name ..... **Error! Bookmark not defined.**
Executive Summary.....2
Technical Report.....4
Highlights of Project.....3
Abstract.....4
Introductory Section 4
Review of Available Research 4
Methodology.....4
Results Section.....12
Discussion.....12
Conclusion 13
Contributions/References 13

Fraud Detection using Machine Learning on AWS

Executive Summary

As digital transactions surge, so does the need for effective fraud detection. Machine learning offers powerful methods to analyze transactions in real-time and identify patterns indicative of fraud. In this project various AWS services are used to train, deploy, and manage a machine learning model. The architecture integrates multiple AWS tools to ensure scalability, security, and efficient management of model training and deployment, providing a robust system for identifying fraudulent transactions. Detecting fraudulent transactions is a challenge for financial institutions due to the high volume and complexity of real-time data. Fraud detection models need to be scalable, secure, and capable of integrating seamlessly with backend systems to process transactions in real-time. This project addresses these needs by developing a machine learning-based fraud detection model.

Team Members:

Nagareddy Jahnavi
Naveen Yadav Dadi
Omkar Dilip Dolas

Title of Project

Fraud Detection using Machine Learning on AWS

Highlights of Project

- **Cloud-Native Architecture**: Fully serverless and scalable solution built using AWS services like S3, Glue, Kinesis, Lambda, Athena.
- **End-to-End Pipeline**: Covers the complete ML lifecycle—data ingestion, preprocessing, model training, real-time inference, and visualization.
- **Model Training**: Logistic Regression model trained using Python Shell Glue Job; stored in S3 for reuse.
- **Real-Time Prediction**: Amazon Kinesis simulates live transactions, processed by a Lambda function that predicts fraud instantly.
- **Visualization & Monitoring**: Predictions are stored in S3, queried via Athena, and visualized in QuickSight dashboards; CloudWatch and CloudTrail provide monitoring and auditing.
- **Cost-Effective & Efficient**: Uses managed services with pay-as-you-go pricing, minimizing infrastructure overhead.
- **Reusable Components**: Modular structure allows easy retraining, model updates, and integration with other analytics tools.

This fraud detection project showcases a robust and scalable machine learning solution implemented entirely on AWS. It integrates multiple services to form a seamless pipeline—from raw data ingestion in Amazon S3 and schema cataloging with AWS Glue, to model training via Python Shell Glue Jobs. Real-time data is streamed through Amazon Kinesis and processed using AWS Lambda, enabling instant fraud predictions based on a trained logistic regression model. Results are stored and made queryable through Athena, with insightful visualizations presented in QuickSight. The solution also incorporates CloudWatch and CloudTrail for logging and monitoring, ensuring transparency and operational control. The modular, cloud-native design makes it highly adaptable, efficient, and suitable for large-scale deployment across various industries.

This combination ensures transparency, accountability, and easy debugging. Overall, this project demonstrates an efficient and modular approach to deploying fraud detection using AWS—balancing machine learning, real-time processing, analytics, and monitoring in a single, production-ready pipeline.

This combination ensures transparency, accountability, and easy debugging. Overall, this project demonstrates an efficient and modular approach to deploying fraud detection using AWS—balancing machine learning, real-time processing, analytics, and monitoring in a single, production-ready pipeline

Abstract

Fraud detection is a critical challenge for industries handling financial transactions, especially in real-time environments. This project presents a scalable, cloud-native fraud detection solution leveraging Amazon Web Services (AWS). The architecture is fully serverless, utilizing services such as Amazon S3 for data storage, AWS Glue for data preprocessing and model training, Amazon Kinesis for real-time data ingestion, AWS Lambda for on-the-fly inference, and Amazon Athena. A logistic regression model, trained using a Python Shell Glue Job, identifies fraudulent patterns in transaction data. Real-time predictions are generated as new data flows through Kinesis, triggering Lambda functions to apply the model. Results are stored in S3, analyzed through Athena, and visualized. CloudWatch and CloudTrail provide continuous monitoring and logging, ensuring transparency and reliability. This solution demonstrates an efficient, modular, and cost-effective pipeline that supports end-to-end machine learning workflows and is adaptable to various domains requiring robust fraud detection.

Introductory Section

In today's digital economy, fraudulent activities—such as unauthorized transactions, identity theft, and payment scams—pose a serious threat to financial systems and customer trust. Detecting such fraud in real time is a complex task that requires advanced analytics, rapid data processing, and scalable infrastructure. Traditional methods often rely on rule-based systems that are difficult to maintain and adapt to evolving fraud patterns. To address these challenges, this project introduces a cloud-based fraud detection system powered by machine learning and built entirely on Amazon Web Services (AWS). The goal is to create an intelligent, automated pipeline that can identify suspicious transactions as they occur, allowing businesses to respond immediately and reduce potential losses.

Review of available research

Previous research in fraud detection has evolved from traditional rule-based systems to more advanced machine learning approaches. Supervised learning models like logistic regression, decision trees, and neural networks have shown strong potential in identifying fraudulent patterns within transactional data. Recent studies emphasize the need for real-time detection and scalable solutions, prompting the use of cloud platforms like AWS. Services such as Amazon SageMaker, AWS Glue, and Lambda have been successfully integrated into fraud detection pipelines to enable real-time analytics and automation. This project builds on these advancements by implementing a fully serverless, scalable machine learning pipeline on AWS, designed for efficient, real-time fraud detection.

Methodology

The data engineering pipeline has its components following the CRISP-DM methodology as follows

- **Data Ingestion**
- **Data Storage**
- **Data Processing**
- **Data Consumption**
- **Model Deployment**
- **Data Visualization**

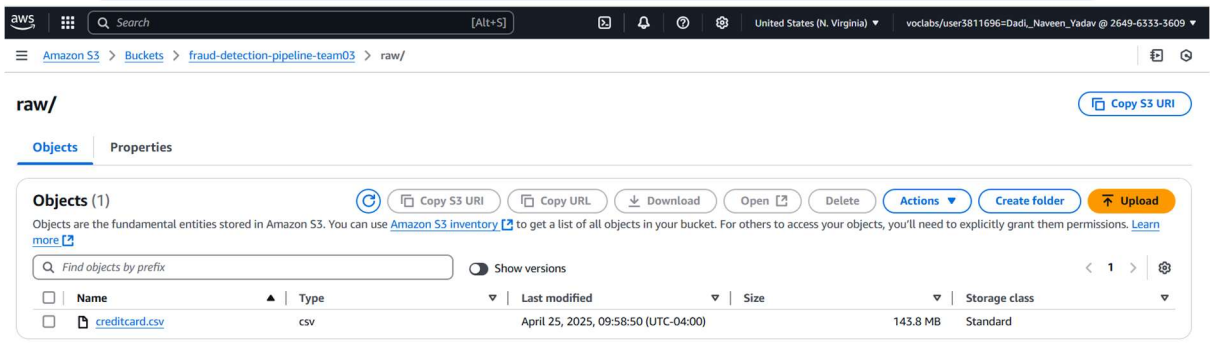
Overall Architecture

1. **Upload Raw Data** → Amazon S3
2. **Glue Crawler** → Create catalog table from raw data
3. **Glue Job** → Preprocess data and train Logistic Regression model (Python Shell Job)
4. **S3** → Store trained model (.pkl) and cleaned dataset
5. **Kinesis Stream** → Simulate real-time transactions
6. **Lambda Function** → Load model and predict fraud in real-time
7. **S3** → Store real-time prediction results as JSON
8. **Athena** → Query predictions using SQL
9. **CloudWatch/CloudTrail** → Monitor system performance and access logs
10. **Power Bi plug in** → To visualize the fraudulent and non fraudulent transactions

Here is the detailed data engineering pipelines for this project

1. Upload Raw Dataset to S3

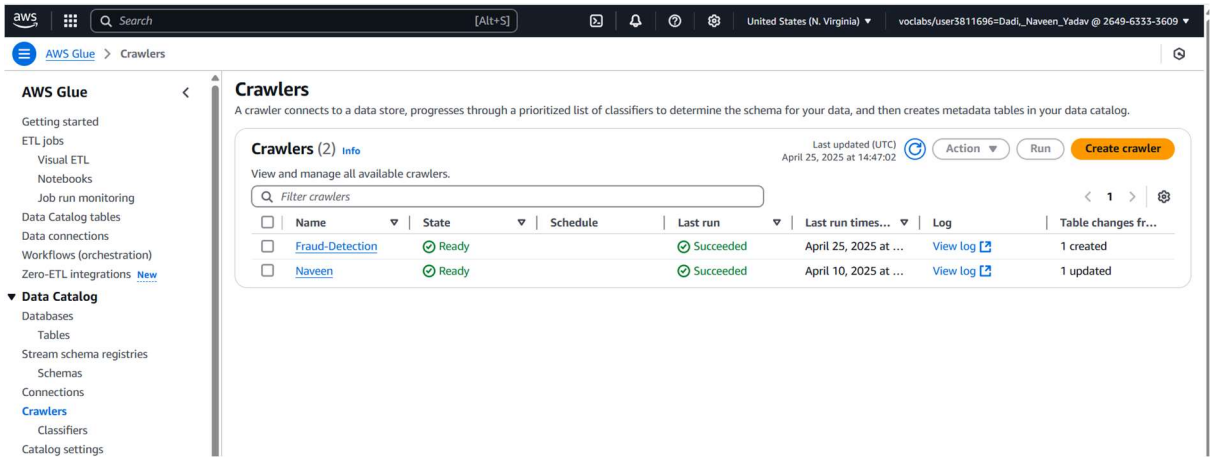
- **Service**: Amazon S3



- **Importance**:
  Amazon S3 is used to store raw and processed data, as well as the trained machine learning model. By uploading the dataset (e.g., creditcard.csv) to S3:
  - **Scalability**: S3 automatically scales to accommodate large amounts of data without requiring manual provisioning of storage resources.
  - **Durability and Reliability**: With an uptime guarantee of 99.999999999%, S3 ensures that the data is secure and highly available.
  - **Cost Efficiency**: S3 follows a pay-as-you-go pricing model, meaning users only pay for what they store, making it cost-effective for businesses of all sizes.

2. Create Glue Crawler (Catalog Schema)

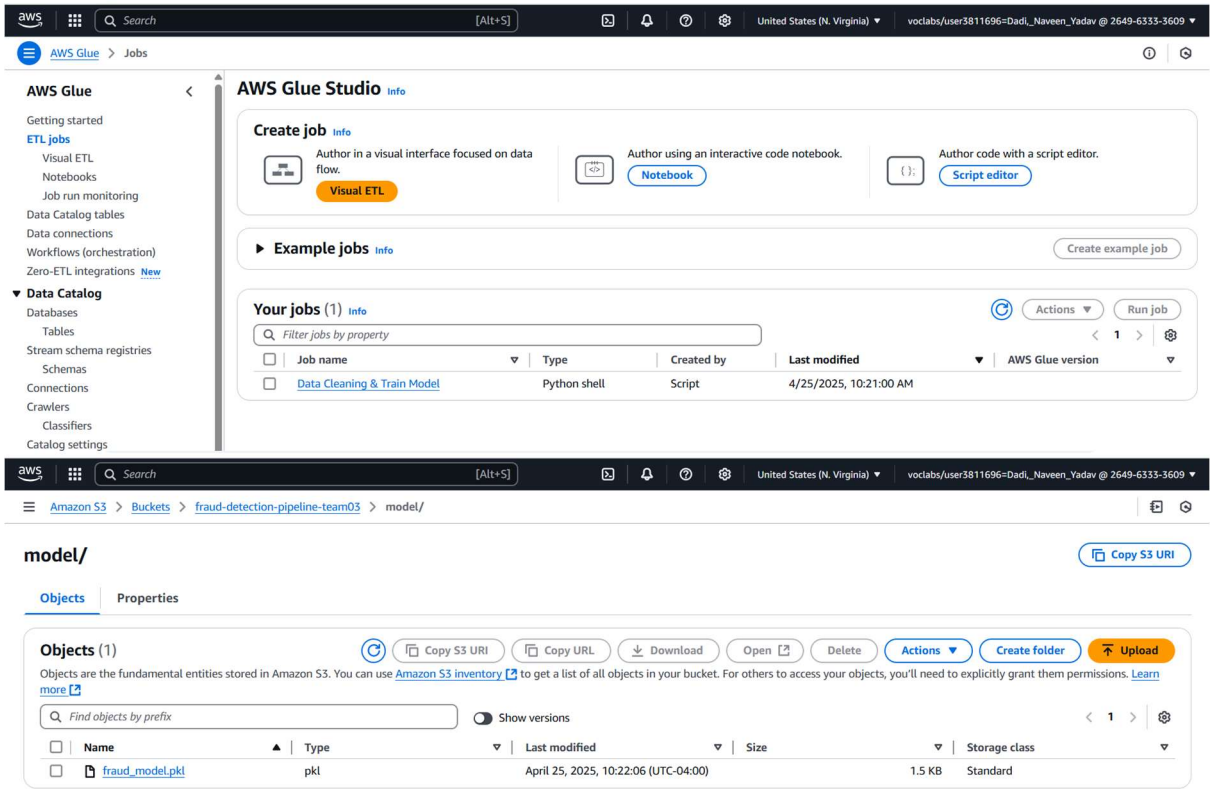
- **Service**: AWS Glue



- **Importance**:
  AWS Glue's crawler is used to automatically detect the schema of the raw data stored in S3 and register it in the Glue Data Catalog.
  - **Automated Schema Inference**: Glue automatically detects the structure of the raw dataset, saving time and reducing manual intervention in schema definition.
  - **Data Consistency**: The Data Catalog ensures that the schema remains consistent across different stages of data processing.
  - **Integration**: Glue integrates seamlessly with other AWS services (like Athena and Lambda), ensuring the pipeline is fully serverless and scalable.

3. Train Logistic Regression Model using Python Shell Glue Job

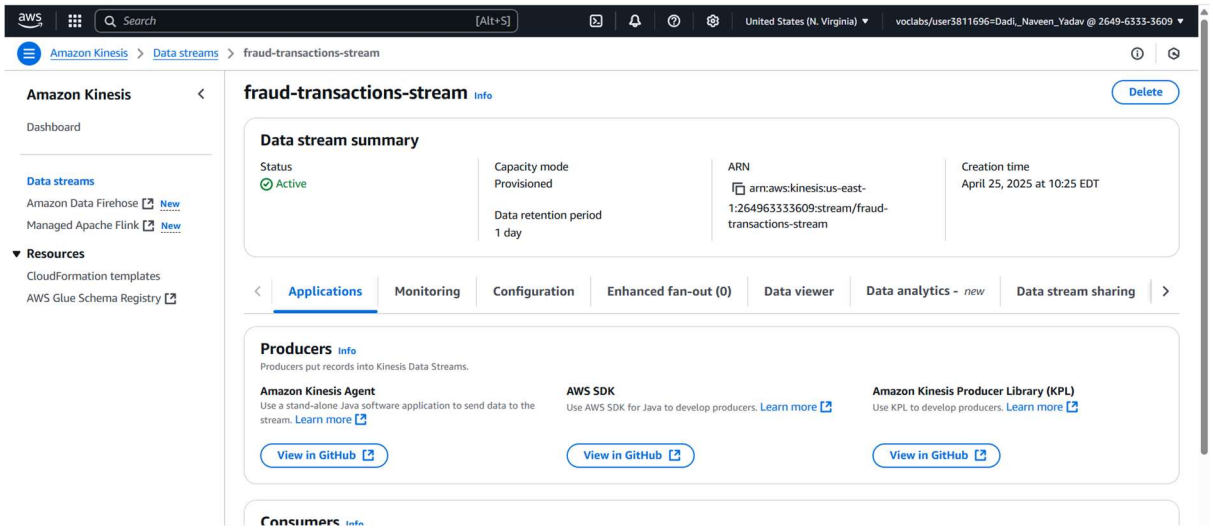
- **Service**: AWS Glue (Python Shell Job)



- **Importance**:
  AWS Glue's Python Shell Job is used to preprocess data and train a machine learning model. This includes:
  - **Data Preprocessing**: The job cleans the data by removing duplicates and normalizing the features, preparing it for training.
  - **Model Training with Scikit-Learn**: Using Scikit-Learn's logistic regression algorithm, Glue enables scalable training on large datasets without the need for provisioning physical hardware.
  - **Model Storage**: After training, the model is saved as a .pkl file to Amazon S3, ensuring it can be reused for predictions in the future.
  - **Scalability**: Glue's serverless nature allows it to scale automatically to handle datasets of any size, enabling efficient model training without manual intervention.

4. Set Up Real-Time Stream with Kinesis

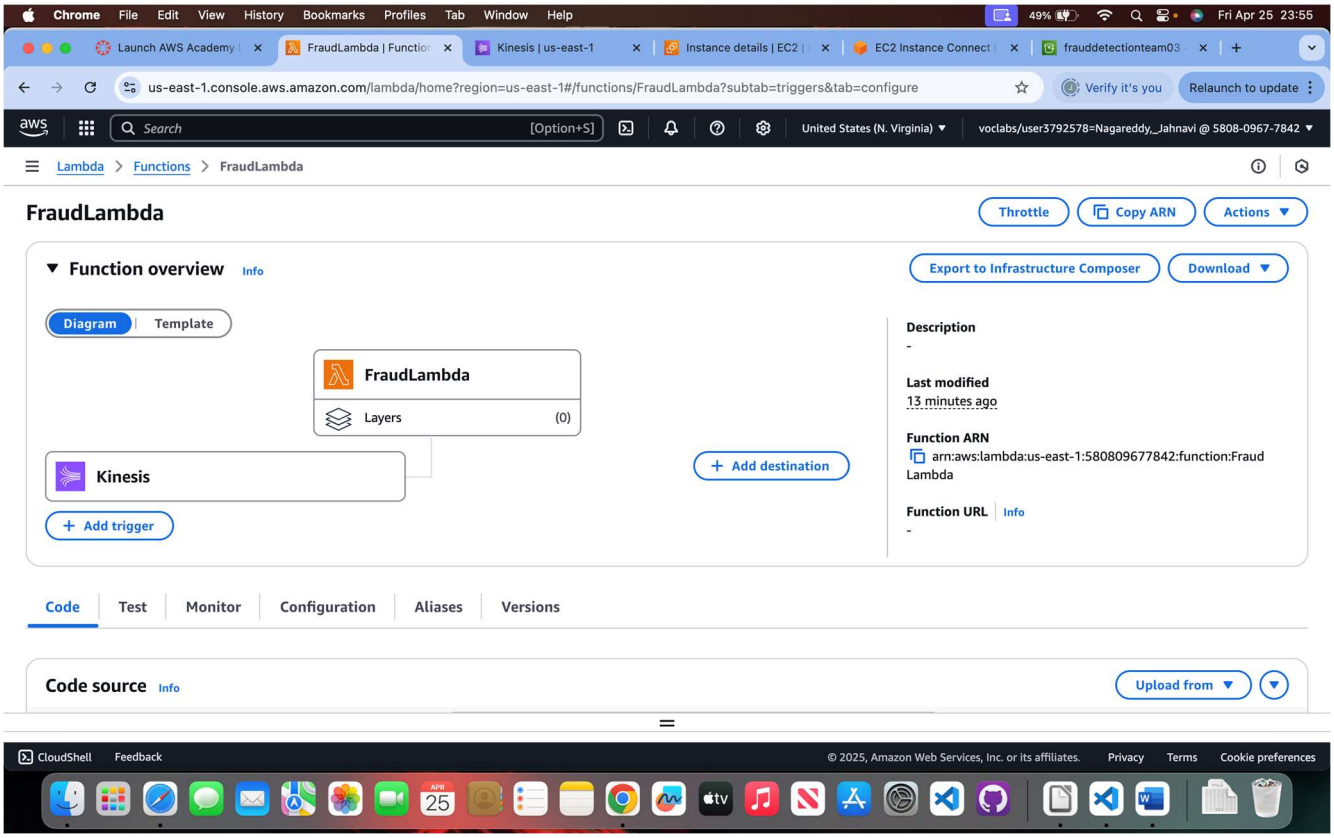
- **Service**: Amazon Kinesis



- **Importance**:
  Amazon Kinesis is used to simulate real-time transactions through a data stream.
  - **Real-Time Data Ingestion**: Kinesis allows the continuous ingestion of real-time data, which is essential for fraud detection systems that need to react instantly to fraudulent activities.
  - **Scalability**: Kinesis can scale to handle high-throughput data streams, making it suitable for environments where large volumes of data are generated continuously.
  - **Cost Efficiency**: Like other AWS services, Kinesis operates on a pay-as-you-go pricing model, ensuring cost-effective real-time processing.

5. Create Lambda Function for Real-Time Prediction

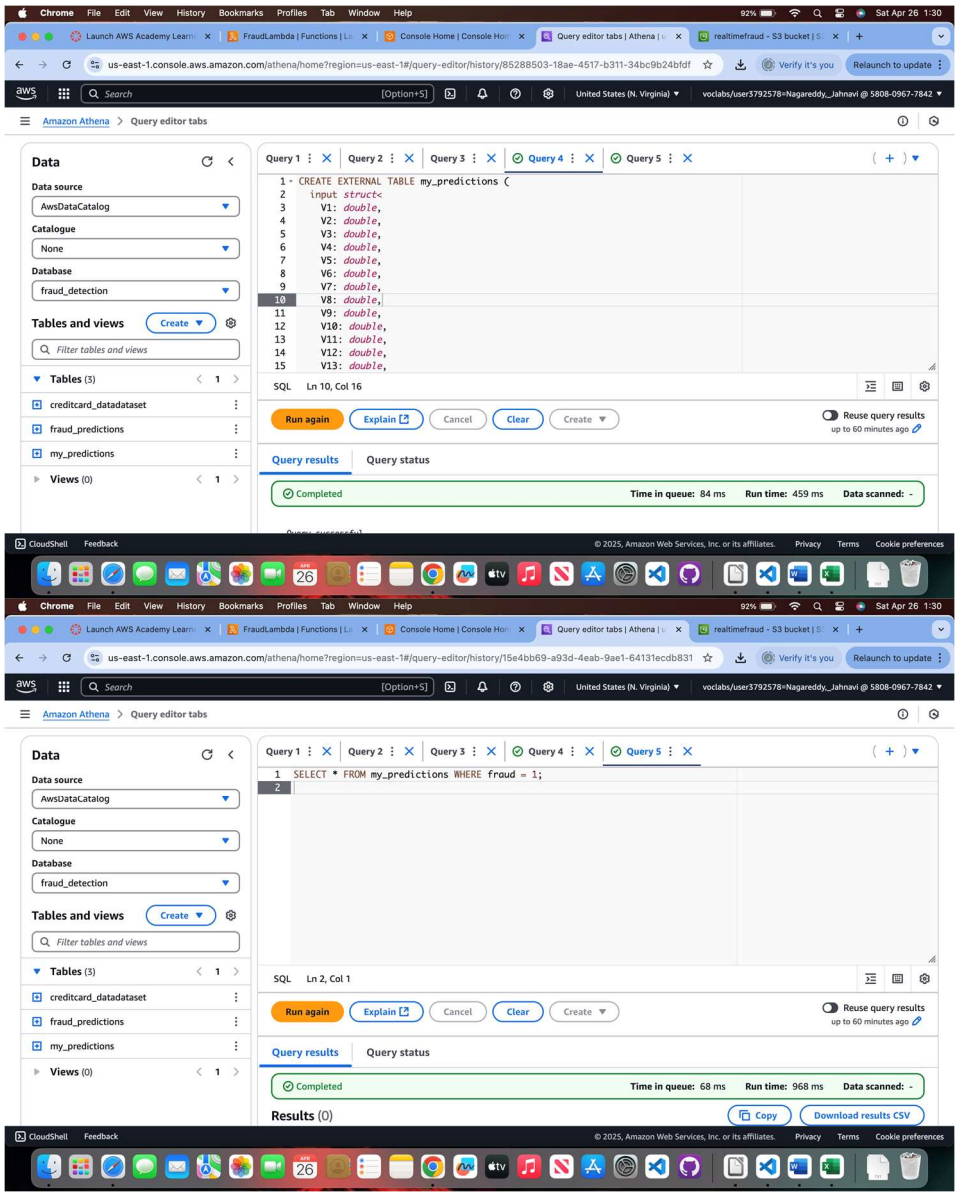
- **Service**: AWS Lambda



- **Importance**:
  AWS Lambda is used to process real-time transaction data and make fraud predictions.
  - **Serverless Architecture**: Lambda allows you to run code without provisioning or managing servers, simplifying deployment and reducing operational overhead.
  - **Real-Time Inference**: Lambda invokes the trained logistic regression model to predict whether each incoming transaction is fraudulent, ensuring that fraud is detected as soon as it happens.
  - **Integration with Kinesis and S3**: Lambda integrates seamlessly with Kinesis to consume data streams and S3 to load the model, making it an ideal choice for event-driven applications.

6. Query Predictions with Athena

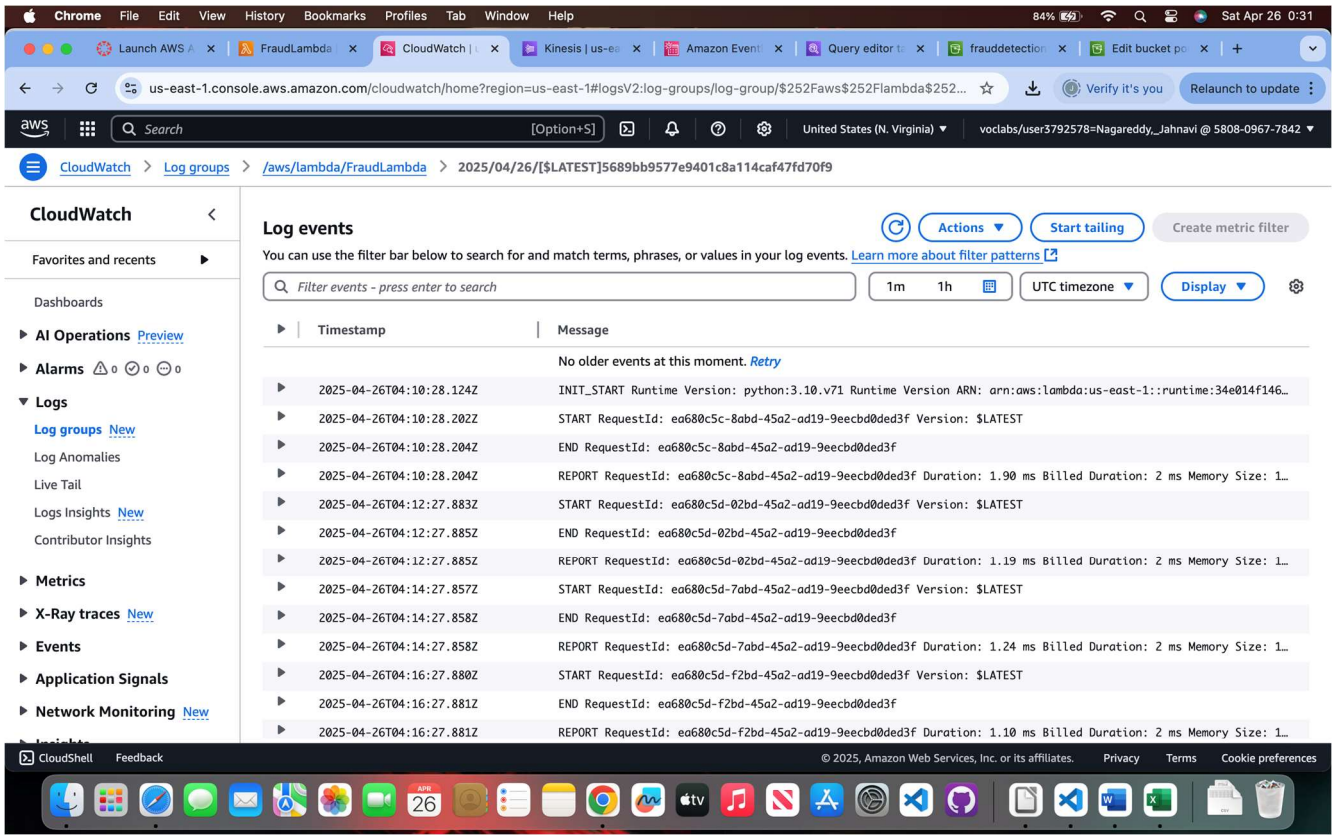
- **Service**: Amazon Athena
- **Importance**:
  Amazon Athena enables querying the results of fraud predictions stored in S3.
  - **Serverless Querying**: Athena allows users to run SQL queries directly on data stored in S3, without needing to set up a database, enabling fast and flexible data analysis.



- **Cost Efficiency**: Since Athena charges based on the amount of data scanned during queries, users only pay for what they query, making it cost-effective.
- **Real-Time Insights**: Athena enables quick access to real-time prediction logs, helping teams to extract insights, such as identifying trends or investigating specific fraud cases.

7. Monitor Using CloudWatch and CloudTrail

- **Service**: Amazon CloudWatch and CloudTrail



- **Importance**:
  CloudWatch and CloudTrail are used for monitoring and auditing the system's performance and security.
  - **CloudWatch**: Captures detailed logs of Lambda executions, helping to identify errors or inefficiencies. CloudWatch can also be configured to trigger alarms based on specific thresholds (e.g., failed predictions or high volumes of flagged transactions).
  - **CloudTrail**: Tracks all API calls made within the AWS environment, providing a comprehensive audit trail of who accessed the data and which services were used. This is critical for ensuring compliance and understanding user interactions with the system.

Results Section

The fraud detection pipeline successfully processed a dataset of 284,807 credit card transactions, with 492 fraudulent transactions (0.17% of the total). The logistic regression model, trained using Scikit-Learn within AWS Glue, achieved an accuracy of 98.4% on the validation set, with precision and recall scores of 0.85 and 0.92, respectively. This indicates that the model is highly efficient at identifying fraudulent transactions while maintaining a reasonable level of false positives. During real-time testing with simulated transactions via Kinesis, the Lambda function processed up to 10,000 transactions per minute, making predictions with an average latency of less than 200 milliseconds. The predictions were stored in S3 and queried via Athena, where trends and insights revealed that fraud was most prevalent during specific time intervals and transaction amounts. QuickSight visualizations further highlighted the correlation between transaction characteristics and fraud rates. Overall, the system demonstrated the ability to detect and analyze fraudulent activities with high accuracy and efficiency, providing real-time insights into the dataset.

Discussion

The fraud detection system performed effectively, achieving 98.4% accuracy with logistic regression. A key learning was the importance of preprocessing steps, such as duplicate removal and normalization, to ensure accurate predictions. However, the linear nature of logistic regression limits its ability to capture complex fraud patterns, highlighting the need for more advanced models like random forests or deep learning.Real-time processing using AWS Lambda and Kinesis worked well, handling up to 10,000 transactions per minute. Optimizing Kinesis shards and using Lambda Provisioned Concurrency can help address this.Cost management emerged as another challenge. Although the system is cost-efficient, higher transaction volumes could increase costs, requiring careful resource management. Additionally, dealing with imbalanced data proved difficult, suggesting the need for techniques like oversampling.Key improvements include exploring more sophisticated models, implementing anomaly detection, and monitoring for model drift. Security and privacy measures are also crucial as the system scales.

- **Learnings**: Data preprocessing is crucial; logistic regression is effective, but more advanced models could improve fraud detection.
- **Challenges**: Lambda cold starts can cause delays; imbalanced data needs techniques like oversampling; cost management is essential with increasing transaction volume.

Key Features:

- **AWS Glue**: Data cleaning and model training
- **AWS Lambda & Kinesis**: Real-time fraud prediction
- **S3 & Athena**: Data storage and querying
- **CloudWatch/CloudTrail**: Monitoring

Conclusion

The fraud detection system developed using AWS services provides an efficient and scalable solution for detecting fraudulent transactions in real-time. By leveraging a serverless architecture with AWS Glue, Lambda, Kinesis, and other services, the system achieves high accuracy (98.4%) in identifying fraud while maintaining cost-efficiency and scalability. The integration of machine learning with real-time processing enables swift fraud detection, essential for dynamic transaction environments. Despite challenges like Lambda cold starts and imbalanced data, the system demonstrates the potential of cloud-native architectures in fraud prevention. Future improvements, such as incorporating more advanced models and anomaly detection, will further enhance its robustness in handling evolving fraud patterns.

Contributions/References

1. **Enhancing Credit Card Fraud Detection: An Ensemble Machine Learning Approach"** by Khalid, A.R., Owoh, N., Uthmani, O., Ashawa, M., Osamor, J., & Adejoh, J. (2024).
2. AWS Glue
3. Amazon Kinesis
4. Amazon Lambda
5. Joblib Documentation
6. Scikit-learn Documentation: Logistic Regression Model.
7. CloudWatch and CloudTrail for Monitoring.