**Serverless Real-time Analytics Dashboard with AWS Kinesis Analytics**

### A Course Project Report Submitted in partial fulfillment of the course requirements for the award of grades in the subject of

**CLOUD BASED AIML SPECIALITY (22SDCS07A)**

by

#### Anmol Agarwal 2210030362

### Under the esteemed guidance of

##### Ms. P. Sree Lakshmi

Assistant Professor,

Department of Computer Science and Engineering



##### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

##### K L Deemed to be UNIVERSITY

*Aziznagar, Moinabad, Hyderabad, Telangana, Pincode: 500075*

April 2025

**K L Deemed to be UNIVERSITY**



##### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### Certificate

This is Certified that the project entitled **“**Serverless Real-time Analytics Dashboard with AWS Kinesis Analytics” which is a experimental &/ theoretical &/ Simulation&/ hardware work carried out by Anmol Agarwal(2210030362), in partial fulfillment of the course requirements for the award of grades in the subject of **CLOUD BASED AIML SPECIALITY**, during the year **2024-2025**. The project has been approved as it satisfies the academic requirements.

##### Ms.P.Sree Lakshmi Dr. Arpita Gupta

##### Course Coordinator Head of the Department

##### Ms. P. Sree Lakshmi Course Instructor

**CONTENTS**

Page No.

1. [Introduction 3](#_TOC_250005)
2. [AWS Services Used as part of the project 5](#_TOC_250004)
3. [Steps involved in solving project problem statement 7](#_TOC_250003)
4. Stepwise Screenshots with brief description 8
5. [Learning Outcomes 17](#_TOC_250002)
6. [Conclusion 18](#_TOC_250001)
7. [References 19](#_TOC_250000)

# INTRODUCTION

## This project presents a Serverless Real-time Analytics Dashboard tailored for monitoring student placement activities using Amazon Web Services (AWS). The goal is to provide a live overview of placement events such as student applications, company visits, and selection results, enabling training and placement cells to make data-driven decisions instantly. The architecture is designed to be fully serverless, removing the need for manual infra- structure management while ensuring scalability and reliability.

## Amazon Kinesis Data Streams (KDS) is used to capture real-time placement data as it is generated. This streaming data is passed to Amazon Kinesis Data Firehose, which efficiently processes and delivers it to Amazon S3 for storage. This setup ensures that all placement-related events are logged and saved in near real-time, making the data available for immediate analysis.

## To visualize the stored data, Amazon QuickSight is integrated with the S3 buckets. QuickSight is used to create interactive dashboards that display key metrics such as the number of students placed, company-wise and branch- wise selections, number of offers, and live application statuses. These dash- boards are user-friendly and can be accessed by placement coordinators and faculty to track the ongoing status of placement drives.

## The serverless nature of the solution ensures reduced operational costs, au- tomatic scaling, and high availability. By leveraging AWS’s managed ser- vices, the system provides a modern, efficient, and insightful platform for colleges to monitor and analyze placement performance in real time. It en- ables better planning, transparency, and quicker decision-making during re- cruitment seasons.

# AWS SERVICES USED AS PART OF THE PROJECT

**Amazon Kinesis Data Streams:**

* Served as the high-throughput ingestion layer for real-time event data.
* Enabled seamless and reliable collection from multiple producers.
* Natively integrated with downstream analytics services.
* Provided consistent, uninterrupted data flow throughout the pipeline.
* Supported scalability and low-latency data capture.

**Apache Flink through Kinesis Data Analytics**

* Acted as the core real-time stream processing engine.
* Used a SQL and Java-based application for stream processing.
* Performed real-time computations like filtering, enrichment, and transformation.
* Supported advanced analytics such as windowed aggregations and pattern recognition.
* Ran in-memory for low-latency, high-speed processing.
* Offered managed deployment with built-in scalability and fault tolerance.

**Amazon S3 via Kinesis Data Firehose**

* Functioned as the final storage layer for enriched and processed data.
* Provided durable, secure, and scalable object storage.
* Enabled reliable data delivery from the processing engine with no custom infrastructure.
* Automatically handled data buffering, transformation, and compression.
* Facilitated historical archiving, batch analysis, and downstream integration.

**Amazon QuickSight**

* Served as the real-time, interactive data visualization layer.
* Offered auto-refreshing dashboards with drill-down capabilities.
* Displayed metrics such as event frequencies, anomalies, and categorical trends.
* Allowed easy querying of stored data from Amazon S3.
* Provided accessible, insight-rich dashboards to stakeholders for live monitoring.

# STEPS INVOLVED IN SOLVING PROJECT PROBLEM STATEMENT

##### Data Ingestion using KDS and FireHose

##### Focused on implementing a robust data ingestion mechanism.

##### Used Amazon Kinesi

##### s Data Streams for data ingestion.

##### Collected high-throughput data from various producers.

##### Buffered incoming streaming data efficiently.

##### Acted as the real-time entry point into the system.

##### Integrated Apache Flink through Kinesis Data Analytics after ingestion.

##### Performed in-stream transformations and computations.

##### Enabled advanced analytics such as aggregation.

##### Enabled filtering of streaming data.

##### Supported time-based windowing operations.

##### Required no manual intervention or server provisioning.

##### Persistent Storage using S3

##### Processed data needed to be securely stored for future analysis and reporting.

##### Used Amazon Kinesis Data Firehose to deliver transformed data.

##### Seamlessly transferred data into Amazon S3.

##### Ensured durability of stored data.

##### Provided scalability for growing data volumes.

##### Enabled easy access to structured datasets.

##### Organized S3 buckets for efficient querying.

##### Supported data lifecycle management.

##### Met both operational and archival storage needs.

##### Data Visualization using QuickSight

* Built a dynamic and interactive dashboard to visualize real-time insights. Used Amazon QuickSight for the visualization layer.
* Connected QuickSight to data stored in Amazon S3.
* Enabled users to view metrics and trends.
* Provided customizable charts and graphs for data exploration.
* Allowed stakeholders to interpret data instantly.
* Supported KPI tracking and real-time decision-making.
* Tested the entire pipeline for reliability.
* Evaluated the system for low latency and cost-effectiveness.
* Confirmed the ability to deliver real-time analytics in a completely serverless environments.

1. **STEPWISE SCREENSHOTS WITH BRIEF DESCRIPTION**

**Step -1: Create a new Kinesis Data Stream**

* Go to Kinesis console.
* Click Create data stream.
* Name it and set shards.

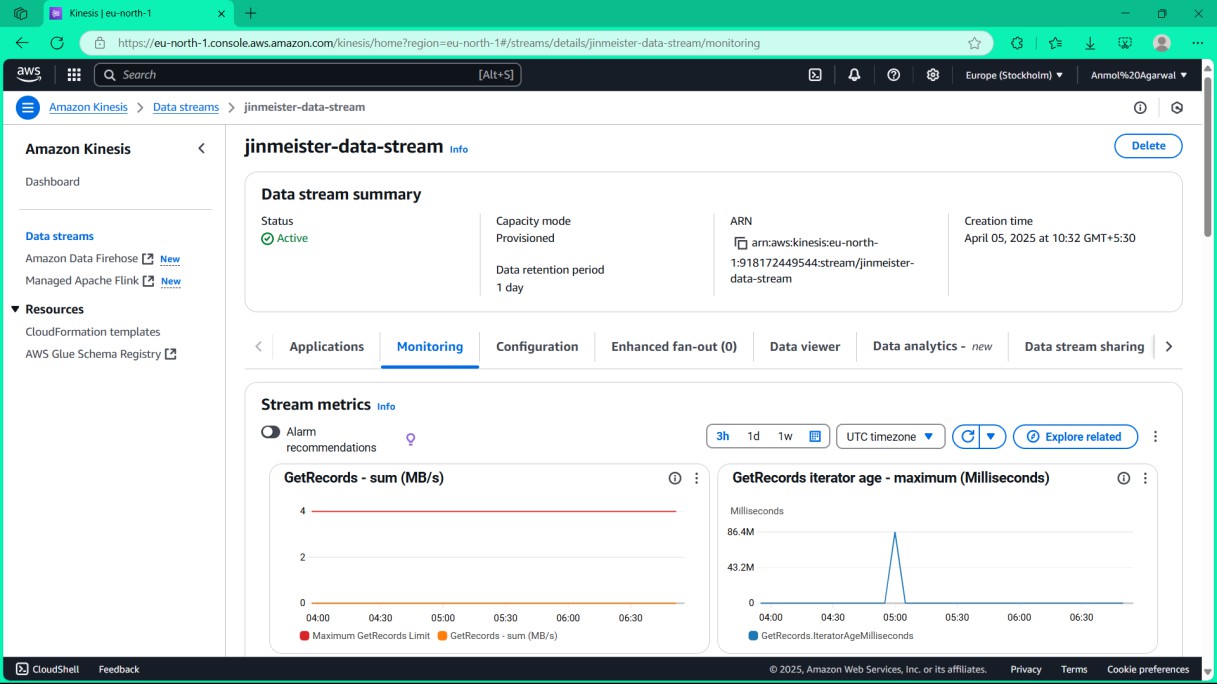
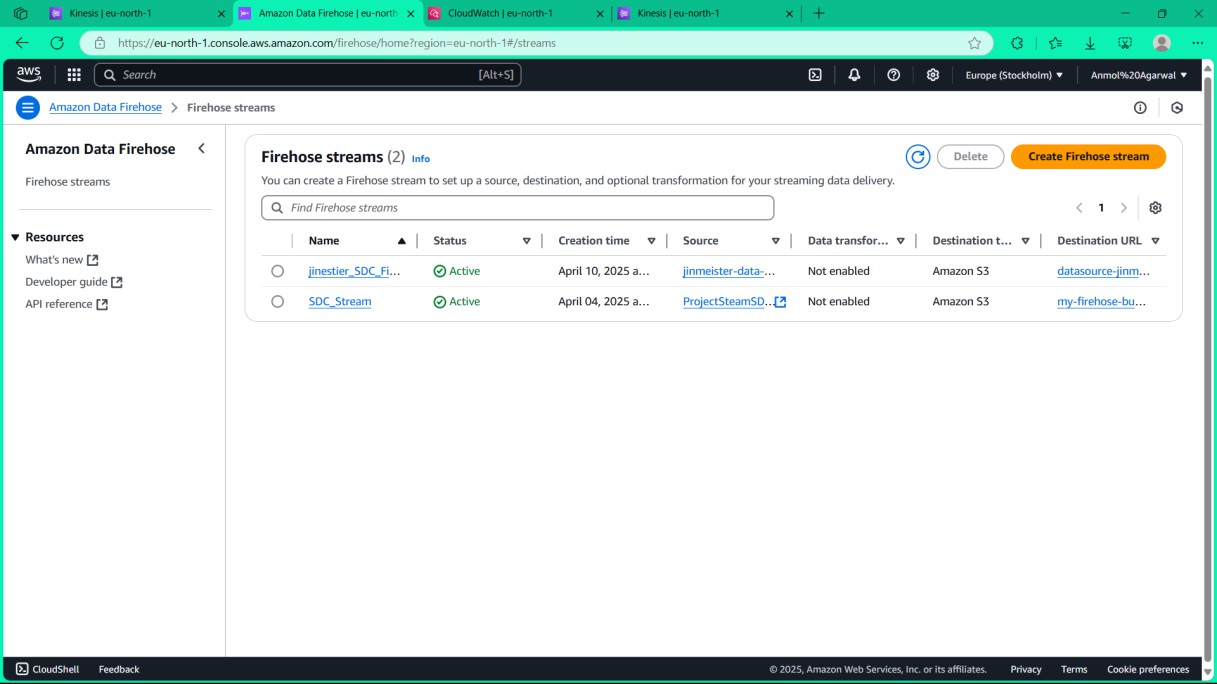
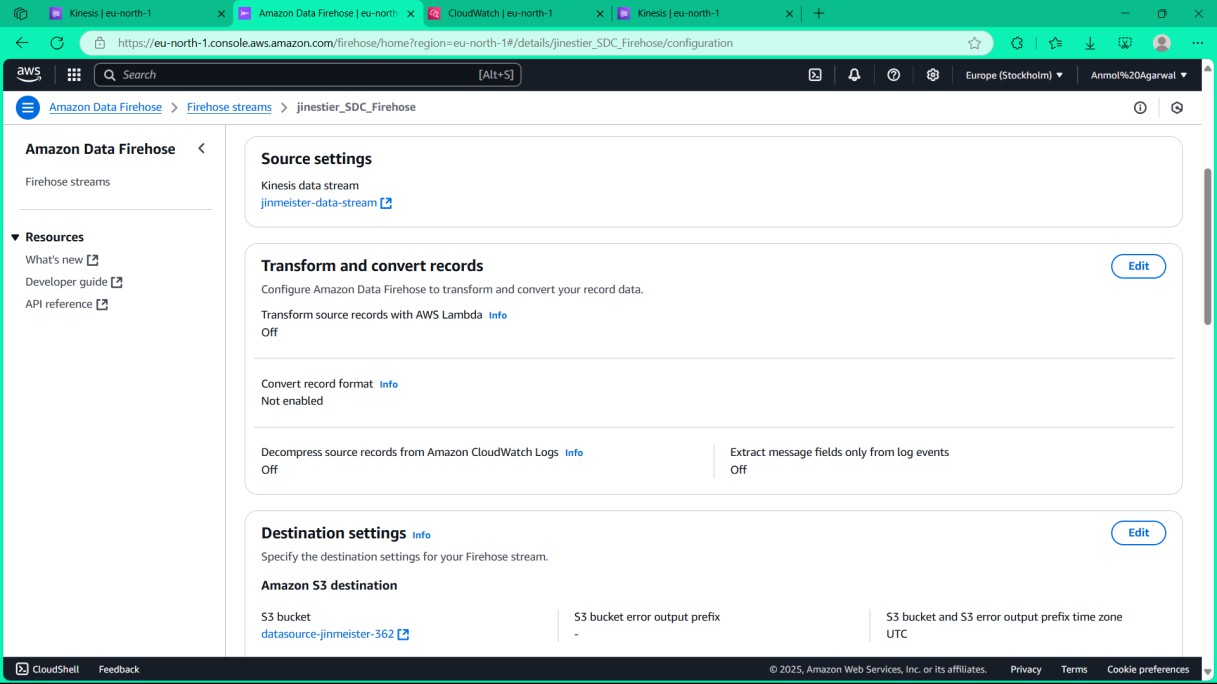
****

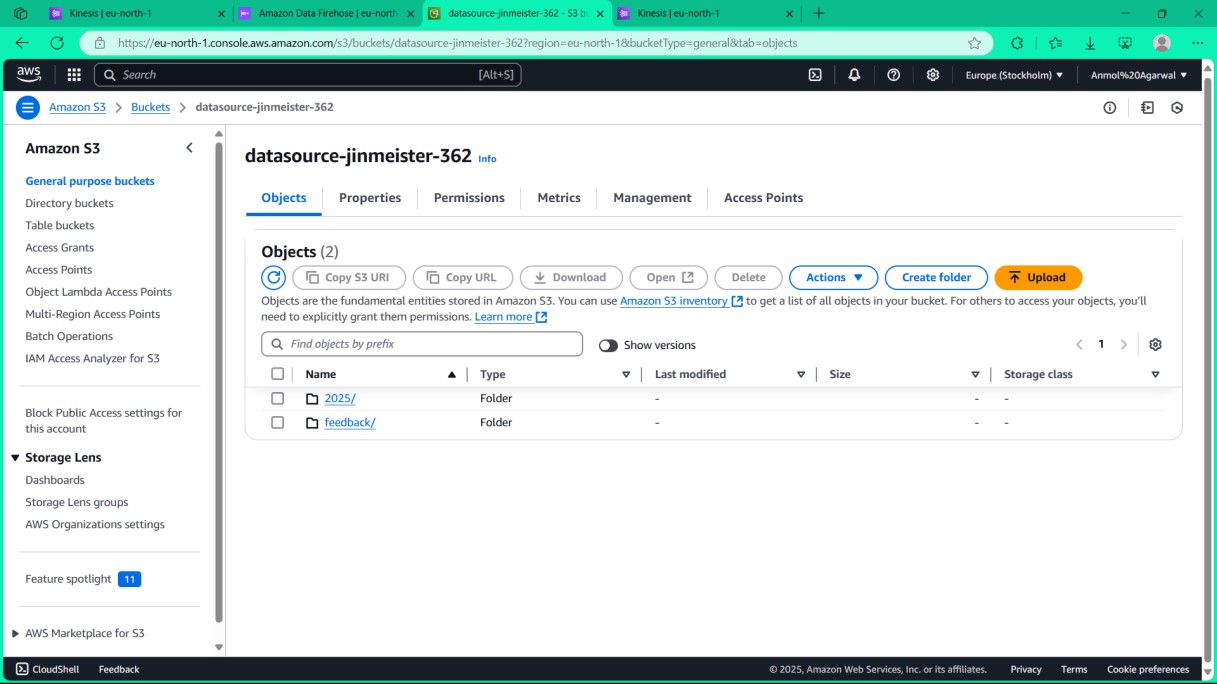
Fig. 4.1: Create a new Kinesis Data Stream



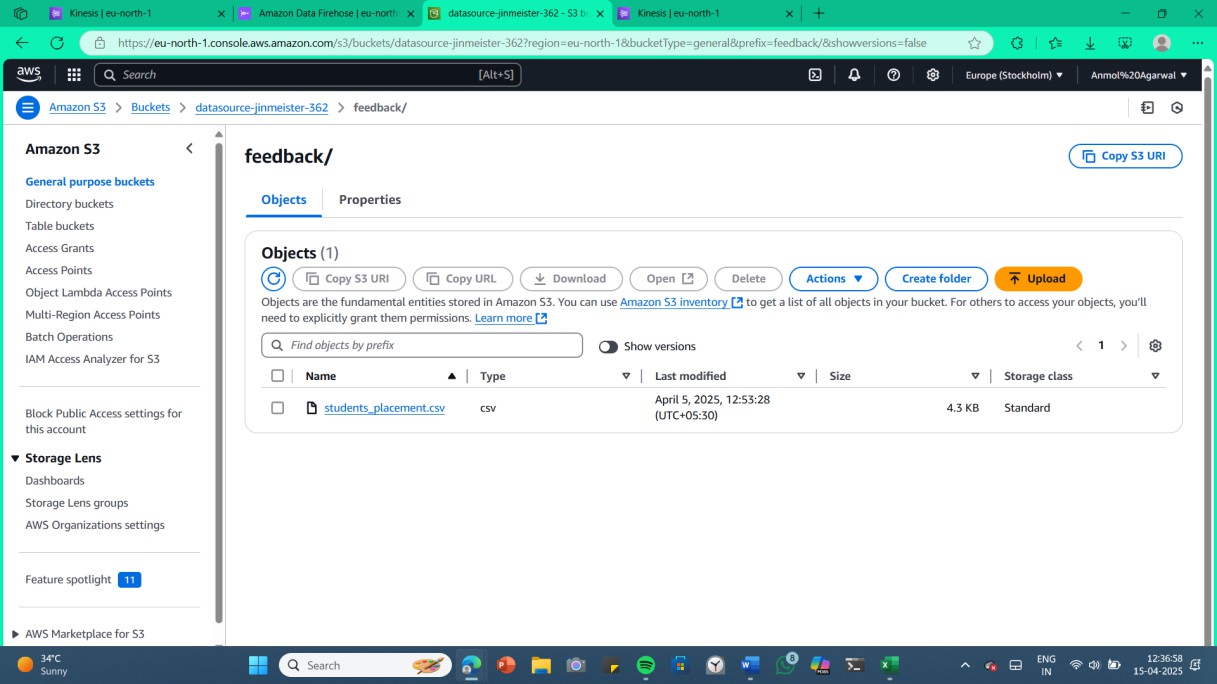
Step-2: Create a new FireHose for data transformation



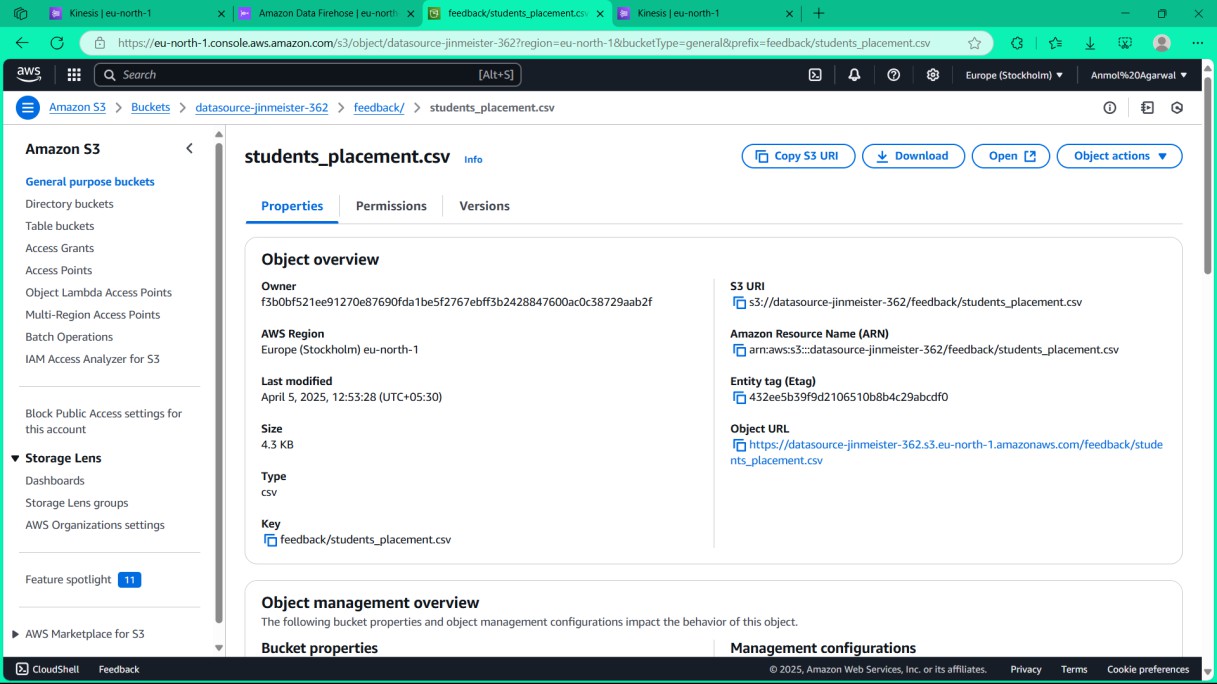
Step-3: Set the source as the KDS and the destination as the S3 bucket



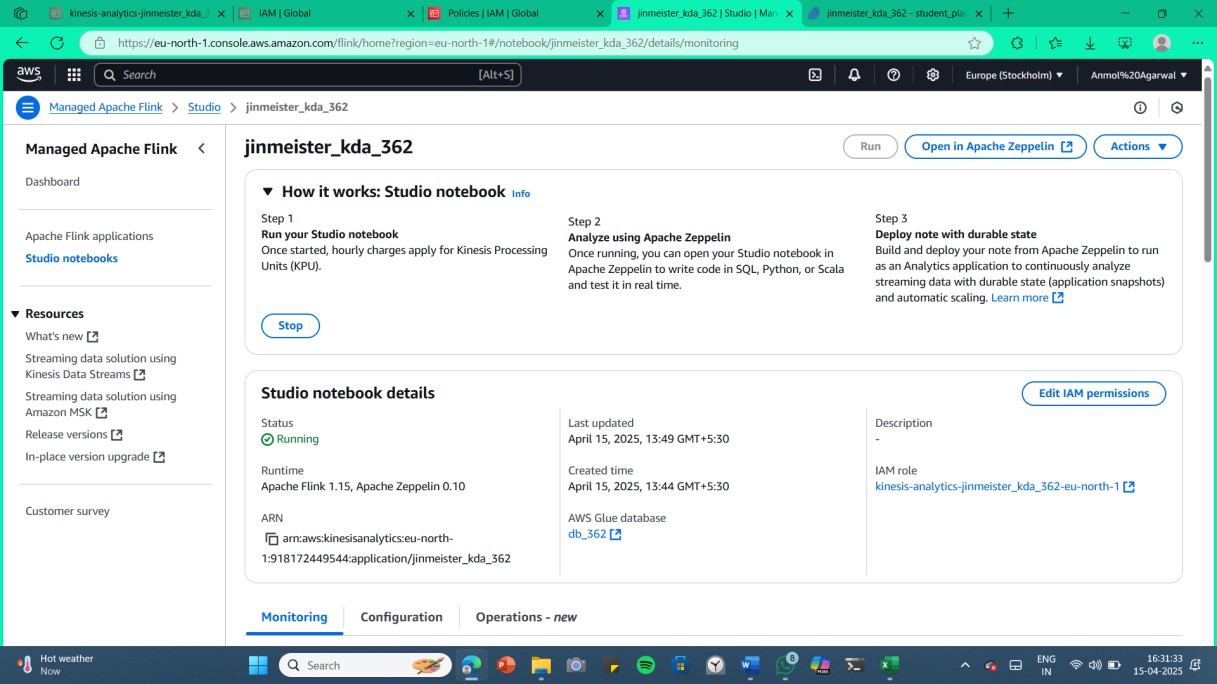
Step-4: Create a folder in S3 bucket named feedback



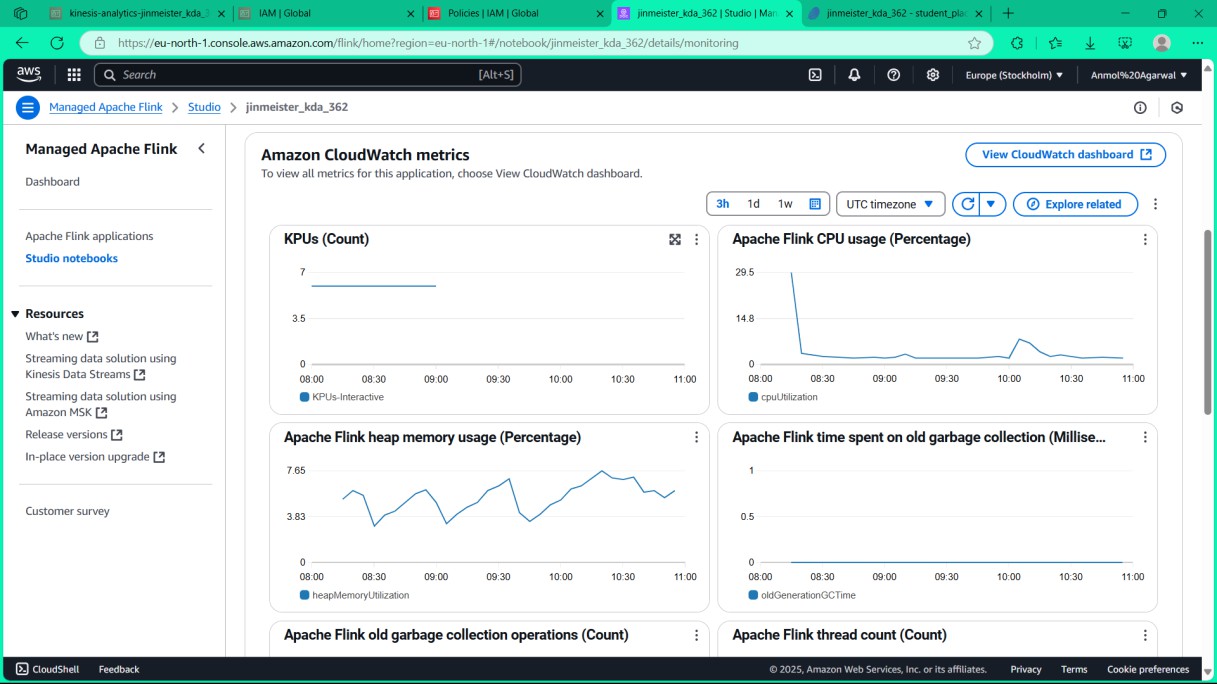
Step-5: Upload the data file students\_placements.csv



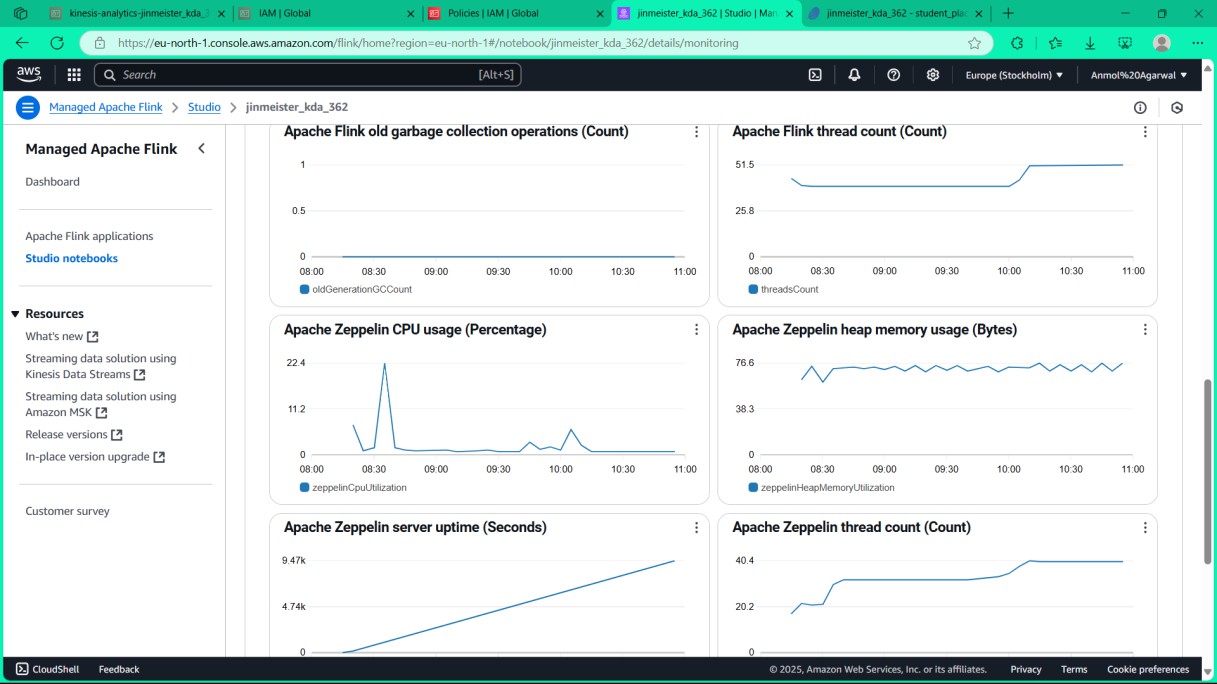
Step-6:Uploaded the file in the bucket



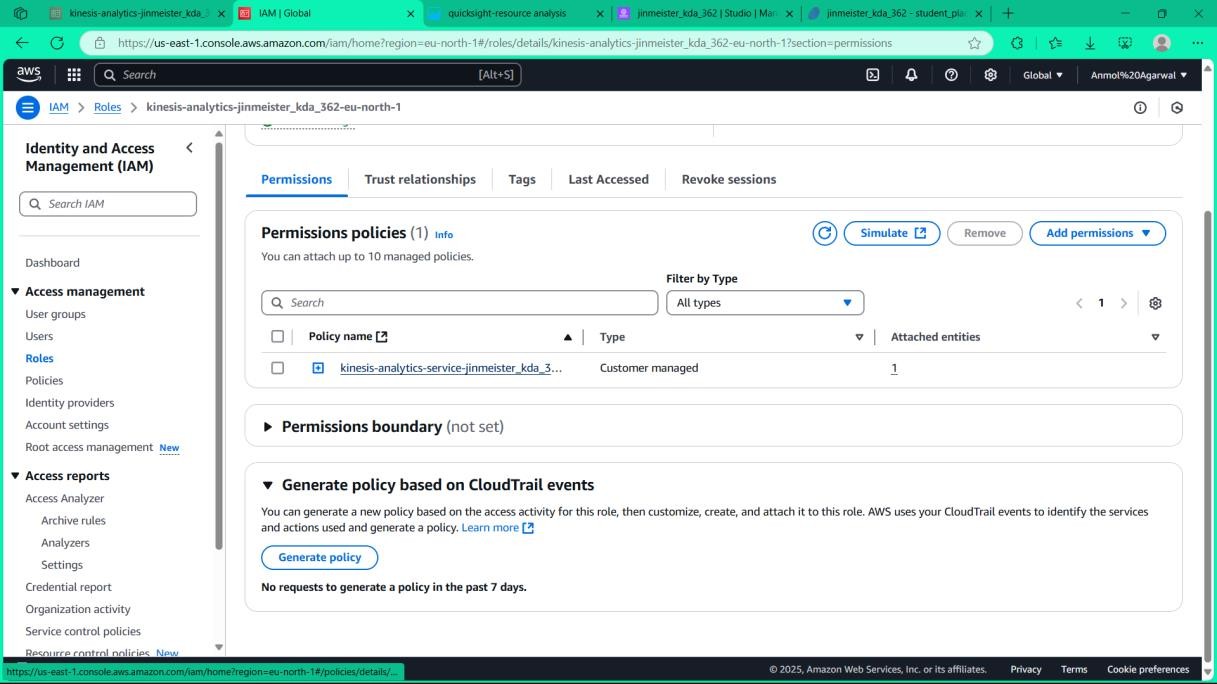
Step-7: Create a KDA in the Apache Flink for the Data Analysis



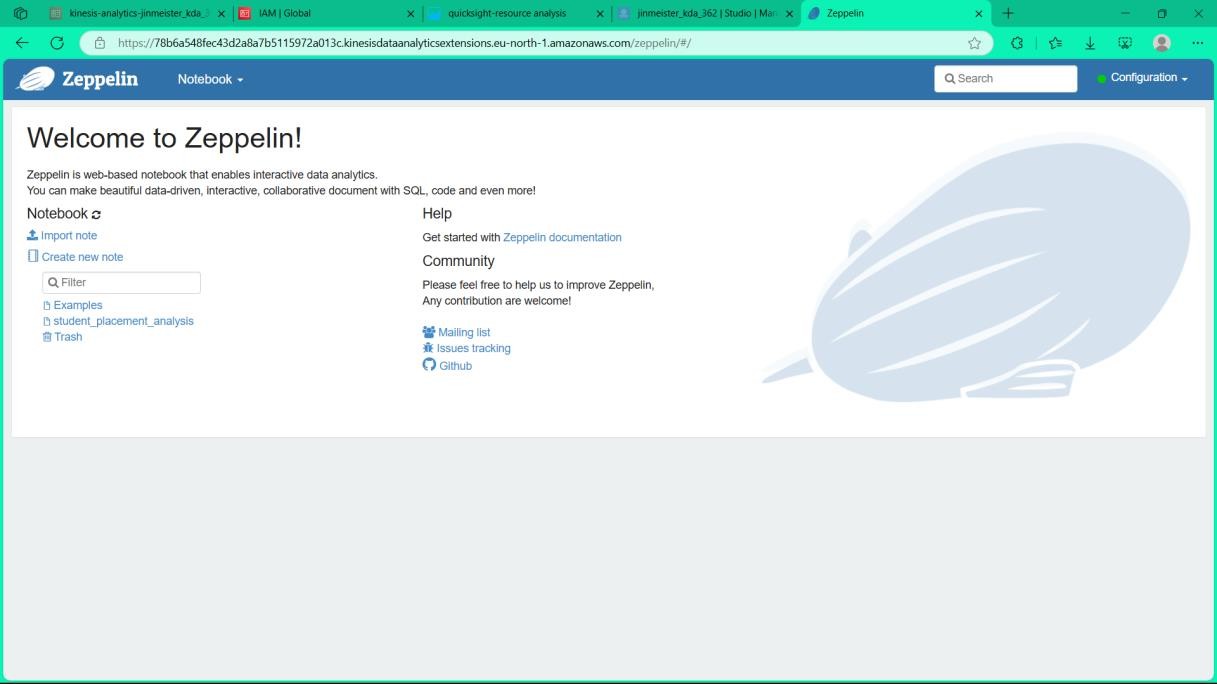
Step-8:Cloud Metrics of Apache Flink representing the memory usage ,CPU usage.



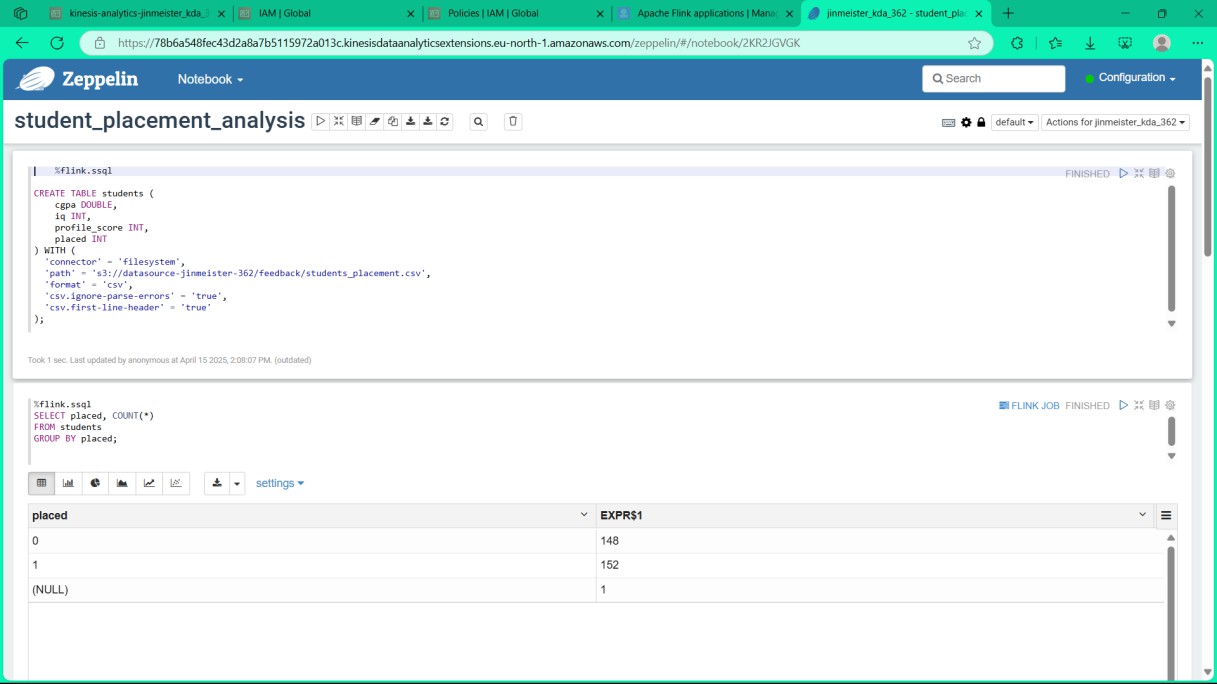
Step-9: Apache Flink Metrics representing the heap memory, Thread count and the server uptime.



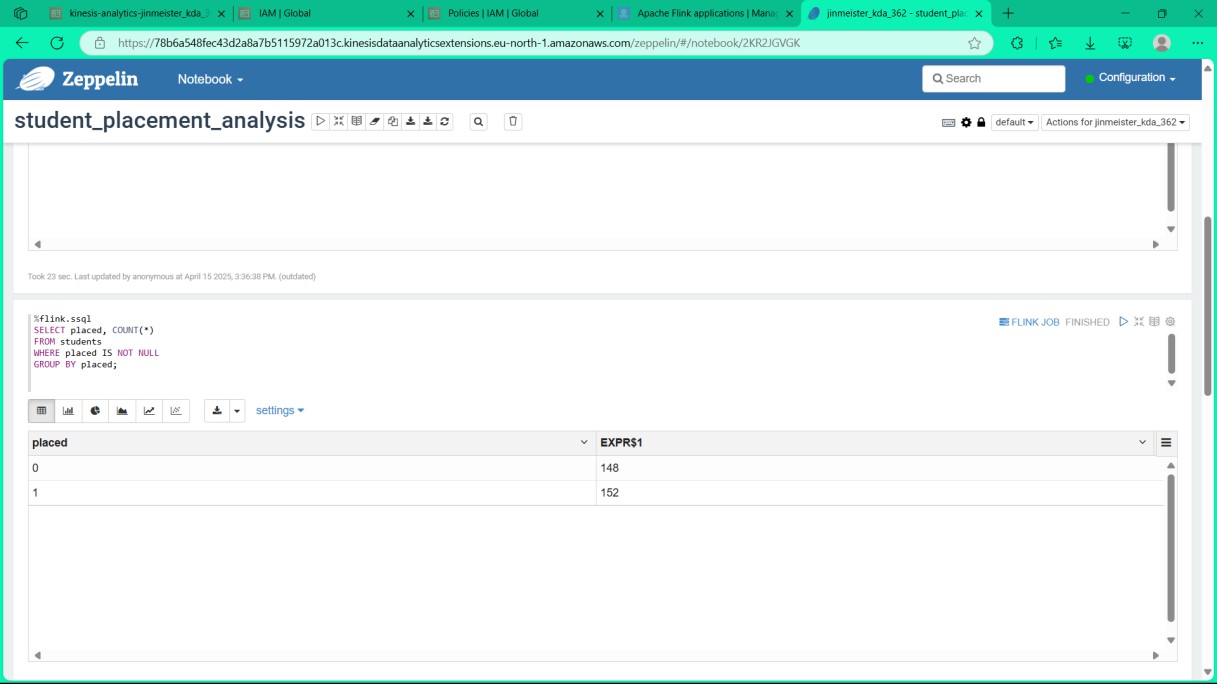
Step-10: Attaching the policies to access and edit the S3 bucket files data.



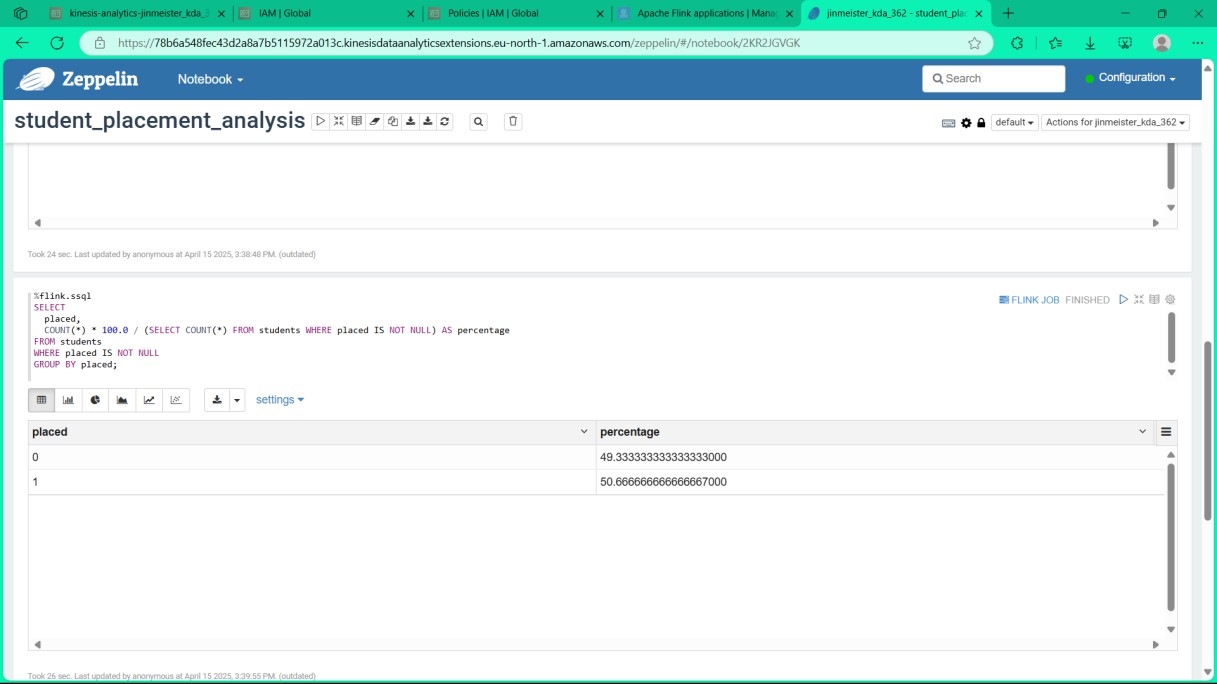
Step-11: Create a Zepplin notebook for data analysis.



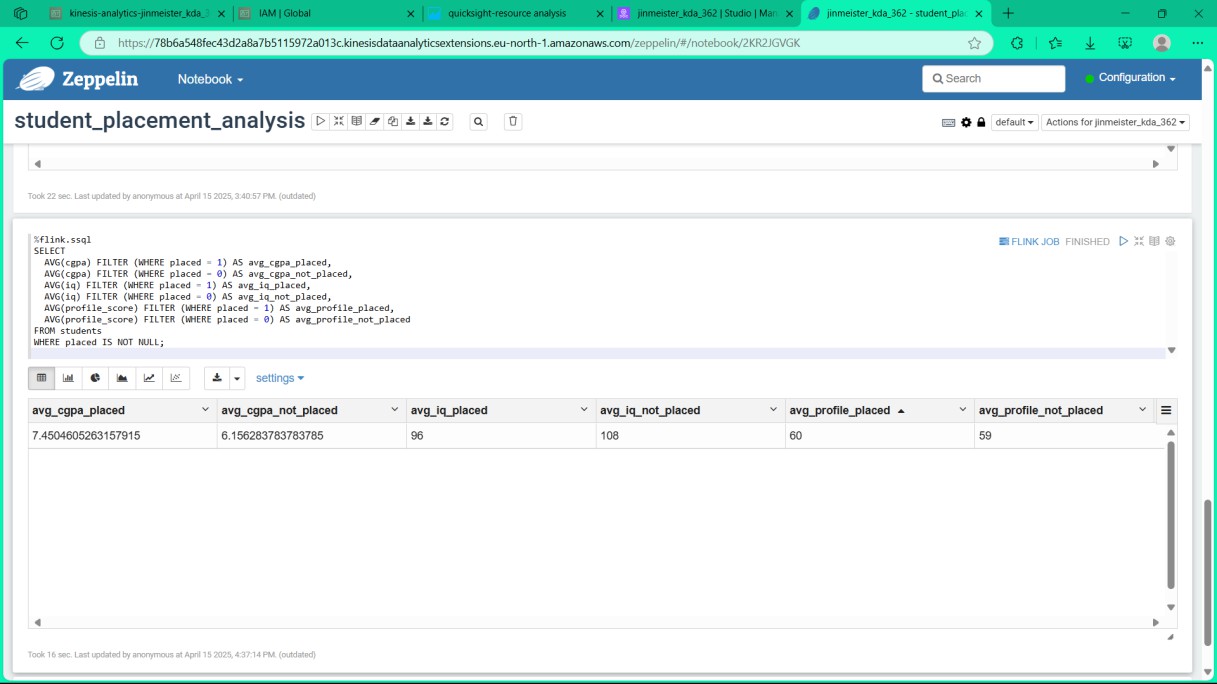
Step-12: Created a student\_placement\_analysis notebook for all the data analysis



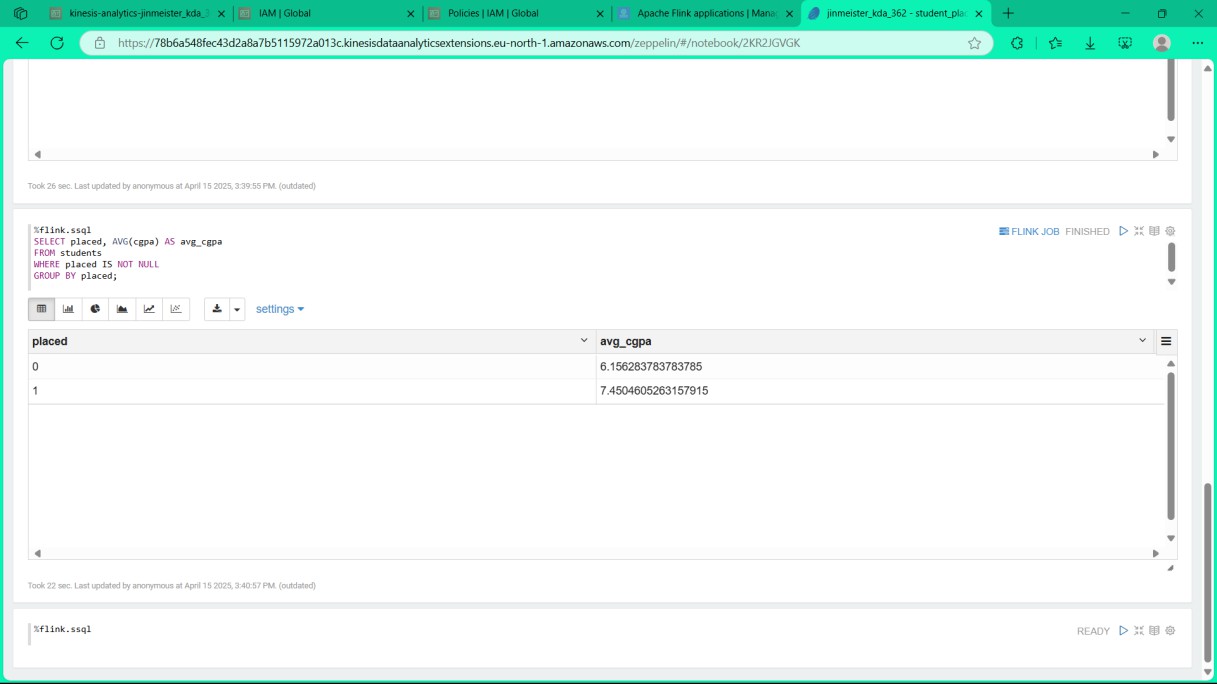
Step-13: Quering the data using SQL queries



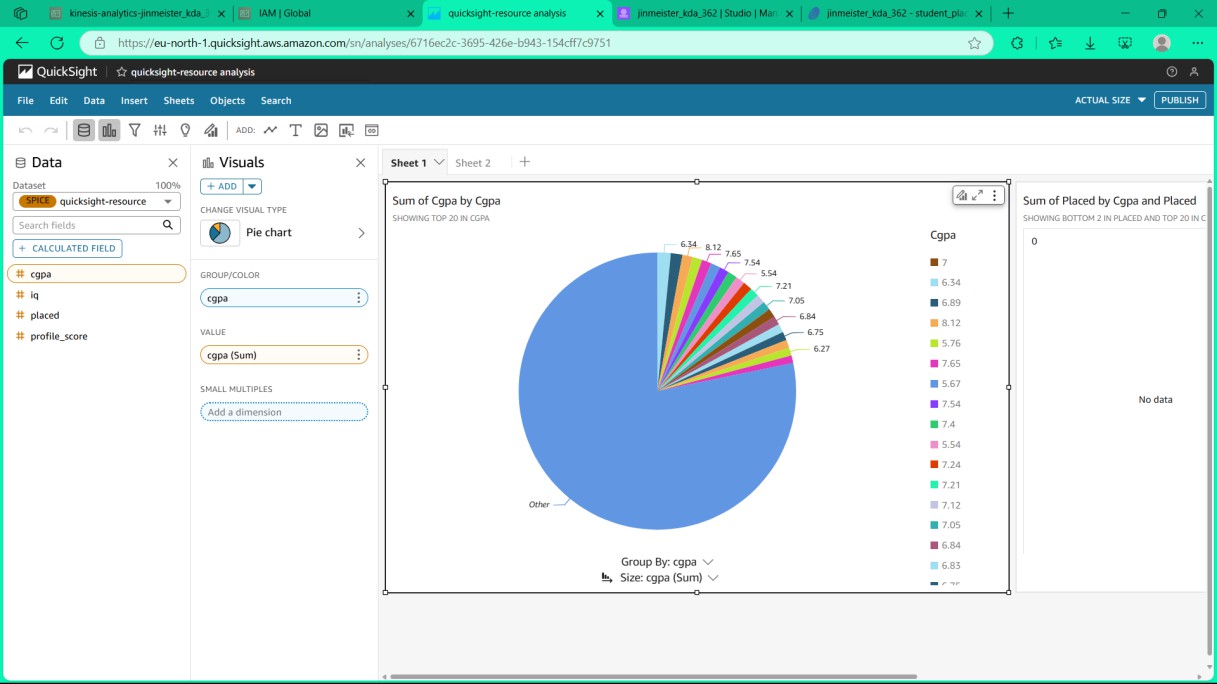
Step-14: Quering the data using SQL queries and finding the percentages



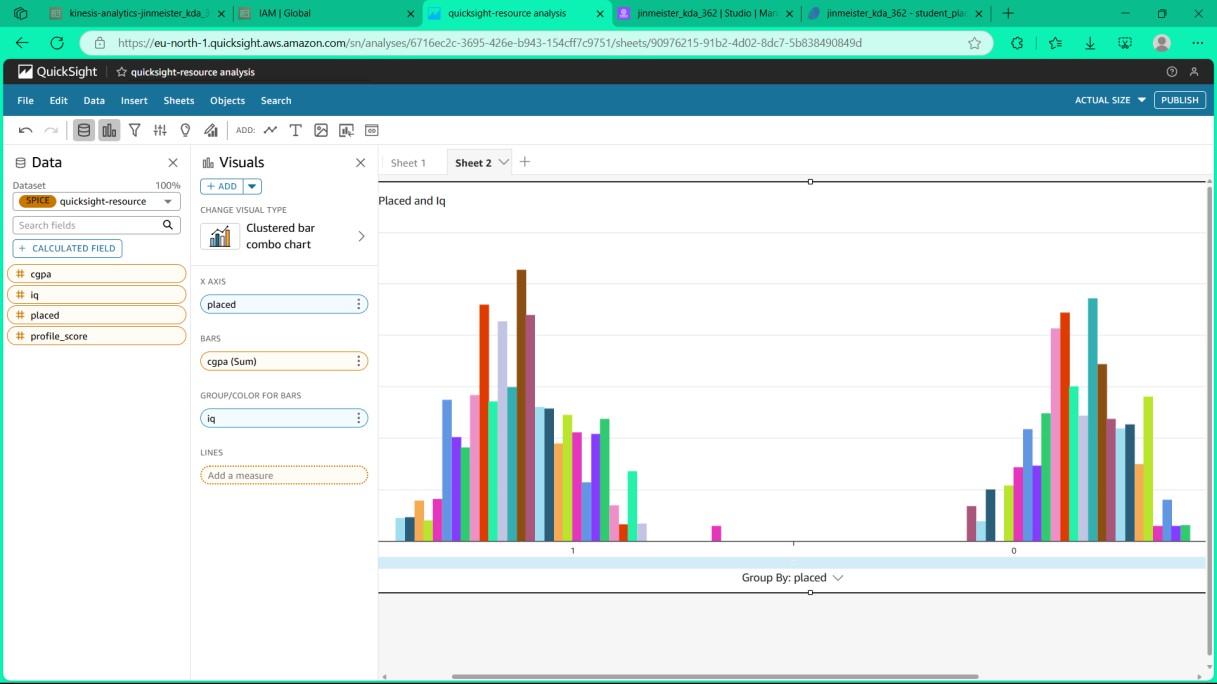
Step-15:Finding the average cgpa of placed people



Step-16: Quering using SQL queries



Step-17: Creating a DashBoard using QuickSight for the analysis



Step-18: Creating a BarGraph using QuickSight for the analysis

# LEARNING OUTCOMES

This project provided valuable hands-on experience with designing and implementing a fully serverless data streaming and processing architecture using AWS Kinesis Data Analytics (KDA). By utilizing services like Amazon S3 for storage, IAM for access control, and KDA with Apache Flink for stream processing, we were able to analyse real-time or batch data without managing any servers. We learned to define table schemas, write and execute Flink SQL queries in Zeppelin notebooks, and effectively connect cloud resources to build a reliable analytics pipeline. The serverless nature of this project reduced complexity and allowed us to focus entirely on data logic and insights.

Through this work, we gained practical skills in analysing structured data and extracting meaningful insights from it. Using the dataset containing student performance metrics (such as CGPA, IQ, and profile score), we wrote SQL queries to evaluate placement trends and statistics. We grouped and aggregated data by placement outcomes, allowing us to discover how various factors influenced success rates. This helped us understand the real-world application of data transformation, grouping, and aggregation techniques to derive actionable conclusions from raw datasets, which is a critical skill for any data-driven decision-making environment.

Lastly, learned how to securely configure and automate cloud-based services to produce a live, interactive dashboard. We implemented IAM role policies to securely grant access to services, simulated permissions to debug issues, and created visual outputs from our SQL queries using Zeppelin. This gave us a strong understanding of AWS security practices, service integrations, and visualization techniques—all within a modern, scalable, and cost- efficient cloud ecosystem. The ability to produce a serverless dashboard that displays live data insights reinforced our learning of end-to-end cloud-based analytics workflows.

# CONCLUSION

The successful implementation of a serverless real-time analytics dashboard using AWS Kinesis Data Analytics highlights the immense potential of cloud-native solutions for data processing and visualization. Through this project, we explored how various AWS services such as Amazon S3 for data storage, IAM for secure access control, Kinesis Data Streams and Firehose for real-time ingestion, and Apache Flink via Kinesis Data Analytics for stream processing can work together in a seamless, cost-effective, and scalable architecture. This approach eliminated the need for managing traditional server infrastructure, significantly reducing operational overhead while allowing for easy deployment and auto-scaling.

The project also offered a hands-on opportunity to work with real-world datasets—in this case, student placement data—allowing us to extract meaningful insights such as placement trends and candidate success rates in real time. We learned how to clean, transform, and query data using Flink SQL and how to monitor the pipeline using built-in AWS tools and dashboards. These insights could be further extended to support decision-making processes for academic institutions, career guidance, and performance tracking.

In conclusion, this project not only enhanced our technical proficiency in data engineering and cloud computing but also fostered a deeper understanding of how modern organizations can leverage serverless platforms to drive agility and innovation. The skills and experiences gained through this initiative are directly applicable to industry use cases, positioning us to tackle more complex real-time analytics challenges in future endeavours.

# REFERENCES

1. Amazon Web Services (AWS). (n.d.). *Kinesis Data Analytics*. Retrieved from:

## <https://aws.amazon.com/kinesis/data-analytics/>

1. Apache Flink. (n.d.). *Apache Flink - Stateful Computations over Data Streams*. Retrieved from: <https://flink.apache.org/>
2. Amazon Web Services (AWS). (n.d.). *What is Serverless?*. Retrieved from:

## <https://aws.amazon.com/serverless/>

1. AWS Documentation. (n.d.). *Getting Started with Kinesis Data Analytics for Apache Flink*. Retrieved from: [https://docs.aws.amazon.com/kinesisanalytics/latest/java/getting-](https://docs.aws.amazon.com/kinesisanalytics/latest/java/getting-started.html) [started.html](https://docs.aws.amazon.com/kinesisanalytics/latest/java/getting-started.html)
2. Zeppelin Documentation. (n.d.). *Apache Zeppelin + Flink Integration*. Retrieved from:

## <https://zeppelin.apache.org/docs/latest/interpreter/flink.html>

1. Noghabi, S. A., et al. (2017). *Samza: stateful scalable stream processing at LinkedIn*. Proceedings of the VLDB Endowment, 10(12), 1634-1645. <https://www.vldb.org/pvldb/vol10/p1634-noghabi.pdf>
2. Kleppmann, M. (2017). *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O’Reilly Media.
3. AWS Architecture Blog. (n.d.). *Real-Time Analytics with Kinesis and Apache Flink*. Retrieved from: [https://aws.amazon.com/blogs/big-data/real-time-analytics-with-](https://aws.amazon.com/blogs/big-data/real-time-analytics-with-kinesis-and-apache-flink/) [kinesis-and-apache-flink/](https://aws.amazon.com/blogs/big-data/real-time-analytics-with-kinesis-and-apache-flink/)