FlashCommerce: Real-time E-commerce Analytics on AWS

1. Introduction

In today's fast-changing online selling world, information gives sellers their most significant edge. Still, many aren't using it well - there's an apparent shortfall. FlashCommerce fixes that problem by offering full-on live analytics right when you need them. Running fully on AWS, this cloud-based tool helps store owners see what shoppers do, track sales shifts, and check stock levels - all instantly.

2. Problem Statement

Sellers using regular online stores usually struggle to get their hands on data easily - yet it's crucial for decisions. While some find workarounds, others just hit dead ends despite trying hard.

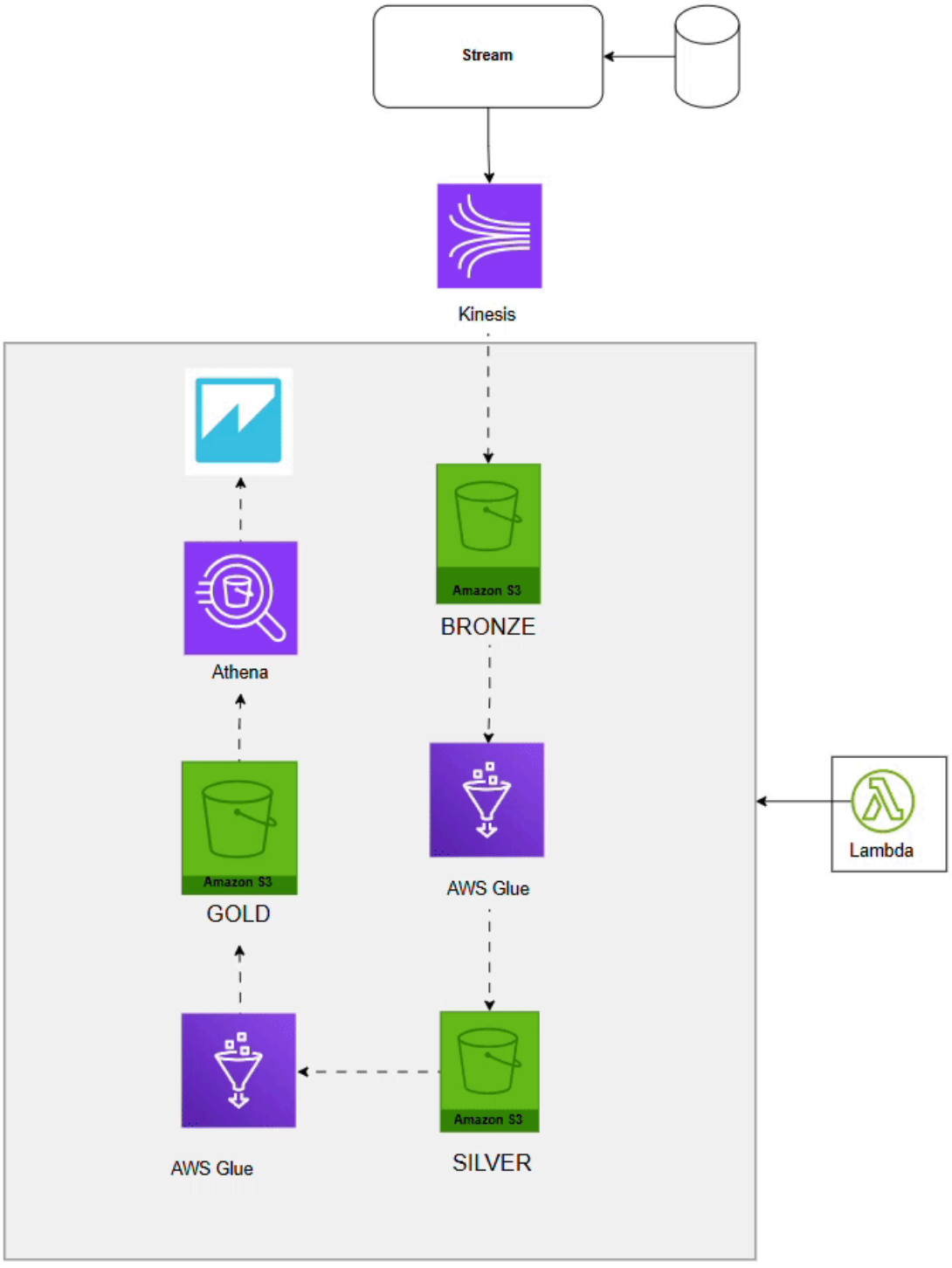
- **Delayed Info:** Old-style sales summaries usually come out slowly, so reps miss real-time updates.
- **Lacking Live Updates:** This makes it hard to spot fast-moving items right away - so prices can't adjust quickly, while slow sellers pile up without warning.
- **Limited Choices:** This happens when sellers miss real-time info - say, abandoned carts or how often views turn into sales. They end up responding instead of staying ahead.

Project Goal & Objectives

The main aim of FlashCommerce? To give sellers easier access to data through a real-time, clickable dashboard. Its focused goals cover:

- **Real-Time Processing:** Capturing customer events (product views, add-to-cart, purchases) the moment they occur.
- **Actionable Intelligence:** Visualising key metrics such as best-selling products, geographical revenue distribution, and conversion funnels.
- **Scalability:** It runs on AWS without servers, which means less upkeep, solid performance, and even under heavy load. Built to stay quick and steady while cutting hands-on fixes.

3. Architecture Diagram



Description of Architecture Flow

The FlashCommerce setup uses a fresh lakehouse model, starting with rough data in the Bronze zone - then moving it through cleanup steps via the Silver stage before reaching polished results at Gold level.

Data Ingestion (The Source)

The process starts by creating online shopping activity - things like checking out an item or buying it. Right away, that info flows into Amazon Kinesis, handling loads of data quickly as it comes in.

Bronze Layer (Raw Data Storage)

A single AWS Lambda runs when the Kinesis stream activates. Once triggered, it unpacks the incoming info instead of leaving it encoded. Then, without delays, it drops everything straight into Amazon S3's Bronze Layer. Right now, the content stays untouched - just plain JSON, nothing altered. This setup keeps records honest, plus allows playback whenever needed.

Silver Layer (Transformation & Cleaning)

AWS Glue grabs raw info from the Bronze zone. Then it cleans things up - tossing out bad entries, changing formats like turning text into time stamps, while adding useful bits such as margin totals. The cleaned set lands in Amazon S3’s Silver area, split by seller ID so it's quicker to reach.

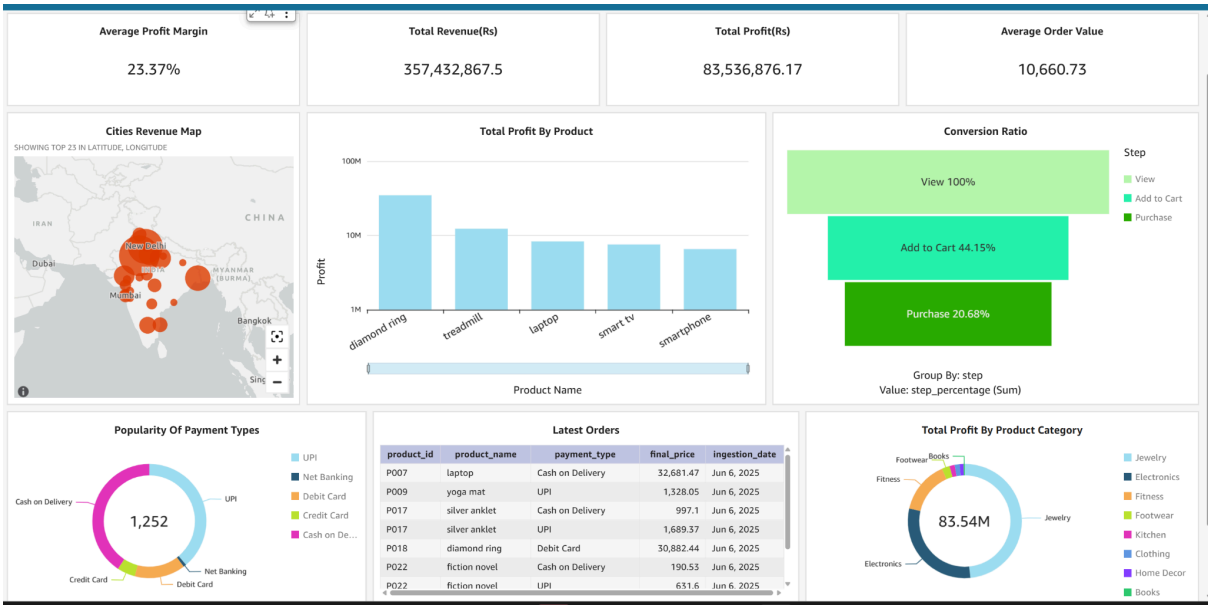
Gold Layer (Aggregation & Business Logic)

A different group of AWS Glue tasks pulls together info into company-ready stats - like summing up earnings for each city, spotting the five most profitable items, or checking how many visits turn into sales. After processing, these outcomes get saved into Amazon S3's "Gold Layer," but in streamlined Parquet format instead.

Cataloging & Visualization

AWS Glue Crawlers scan the data, so it can be searched using Amazon Athena. Then, Amazon QuickSight pulls from Athena to show real-time visuals to users.

4. Dashboard Screenshots



Dashboard Analysis

The FlashCommerce dashboard gives a full snapshot of your business right away. Its main features show up clearly: one part tracks sales fast, while another highlights customer activity instantly; each section updates live so you always see what's changing, yet stays simple to understand - no clutter, just clear info that matters.

- **KPI Headers:** Show key numbers right away - like total sales, overall profit, typical profit margin (say, 23.37%), or average order price. That way, sellers can quickly check how well things are going financially.
- **Funnel Graph:** Shows how well a sales process works - starting with views at 100%, then moving to add-to-cart, say around 44.15%. From there, it follows through to actual buys, like 20.68%. This layout reveals exactly where people stop engaging. Sellers use this to spot weak points quickly. Each stage flows into the next, making leaks obvious.
- **Geographic View:** Showing income across cities - like Mumbai or New Delhi - using bubble size to represent earnings. Bigger circles mean more money was made there, so teams can focus ads where they'll work best.
- **Product Performance:** A bar chart ranking the "Total Profit by Product," highlighting top performers like "diamond ring" or "treadmill," enabling inventory optimization.
- **Category Distribution:** A donut chart showing "Total Profit by Product Category," breaking down which segments (e.g., Electronics, Fitness, Jewelry) are driving the most value.

5. AWS Services, Platforms, and Technologies Used

This setup uses serverless plus managed tools to keep maintenance light while scaling easily. Here's a clear look at what tech we picked - along with why each choice makes sense.

A. Processing along with task management

Service	Role	Motivation
AWS Lambda	When fresh info hits Kinesis, it kicks into action right away. Works like a link from incoming flow straight to where things get saved.	Because it's serverless, we went with Lambda. Since it runs code when triggered, costs are based just on actual processing time for each record. That means no upfront setup or handling of servers for data intake. As a result, the system adjusts smoothly when traffic jumps up or down.
AWS Glue (ETL & Crawlers)	Manages tough tasks like shifting data around - also takes care of labeling and organizing info behind the scenes.	Glue gives you an easier way to use Apache Spark - no need to set up clusters yourself. It handles setup behind the scenes, saving time while keeping things running smoothly. We had big data sets to handle – so we cleaned them up for Silver layer while combining info into Gold. That's where Glue Jobs came in handy; they ran tough cleanup tasks through PySpark. Crawlers: These tools scan your files automatically, updating the catalog whenever fresh data lands in S3 - so it's ready to search right away.

B. Keeping data safe + how it’s collected

Service	Role	Motivation
Amazon Kinesis Data Streams	The entry point for real-time data.	Kinesis handles fast-moving data well. So it keeps things running smoothly when info flows quickly from the shop to analysis tools. Even if the analytics part lags behind, incoming records won't get lost. This setup separates creators of data from those using it, making timing less critical.
Amazon S3 (Simple Storage Service)	Data lake works as main storage for bronze, silver, also gold levels.	Because S3 gives top-tier reliability - think 11 nines - and super affordable space, separating storage from processing makes sense. Instead of mixing them, we stash old raw data without spending much (that’s the Bronze level), yet still keep cleaned-up results (the Gold tier) on hand for fast queries.

C. Stats plus charts

Service	Role	Motivation
Amazon Athena	Acts like a live question-answer tool.	Athena gives us a way to use basic SQL commands on data sitting in S3 - no extra steps needed because it pulls straight from storage using familiar syntax that’s easy to work with day after day. Motivation: No more setting up databases or warehouses just to run queries - skip the ETL hassle. That cuts down how long it takes to get answers, while also saving money since you’re charged only for what your query actually scans.

Service	Role	Motivation
Amazon QuickSight	The business intelligence (BI) layer.	QuickSight links up smooth with Athena - no hassle there. This tool scales on demand without needing servers, so it's ready when you are. The SPICE engine? That's what makes reports snap to life fast, no matter how big the data gets.

D. Coding languages along with tools

Python runs the Lambda stuff - uses Boto3 to talk to AWS bits. For Glue tasks, it leans on PySpark to handle big data across clusters. The pick came down to solid tools for number crunching and smooth hookups with Amazon's cloud setup.

6. Detailed Platforms and Technologies Justification

This part explains the exact AWS tools used in FlashCommerce - shows why they were picked by looking at tech requirements, perks, and what drove the project.

A. Data intake plus keeping

Amazon Kinesis Data Streams

Motivation & Need: In fast-moving online stores, data like clicks or buys flows nonstop - grabbing it on the spot stops losses. Old-school bulk updates just can't keep up when you need live insights.

Justification: Kinesis works as a serverless option built just for live data streams. It handles everything without needing setup or management. Besides separating data creators from users, it helps the system manage sudden surges - like those at discount events - without breaking connected tools.

Amazon S3 (Simple Storage Service)

Motivation & Need: We needed a storage setup that could grow easily without costing too much, so we went with the Lakehouse model - split into Bronze, Silver, and Gold levels. S3 holds everything together in the data lake, using object storage that works smoothly with AWS Glue - while also linking well with Athena.

Benefit	Description
Scaling	Happens on its own as we collect more data.
Grouping	Set up clear zones for unprocessed info (Bronze), tidy datasets (Silver), also summary stats (Gold) through folder labels.

B. Processing plus task coordination

AWS Lambda

Motivation & Need: A light, event-based setup had to handle Kinesis entries right when they show up, kicking off later tasks without delay - using triggers that respond fast.

Justification: Lambda runs code without needing servers. Since we needed live data, it pulled info straight from Kinesis. After that, it kicked off Glue Crawlers using automated triggers.

Benefit	Description
Cheap to use	You pay just for the time your code runs - down to milliseconds.
Automation	It runs everything on its own - starts with grabbing data, then moves to scanning it, after that kicks off processing tasks.

AWS Glue (ETL & Crawlers)

Motivation & Need: Raw JSON data usually comes cluttered, disorganized. So we wanted a solid way to tidy it up - turning chaos into something usable for business.

Justification: Because glue's a tool without servers, it simplifies finding, cleaning, or merging info. Since it runs on Spark, handling data gets smoother through automatic setup.

Benefit	Description
Managed Spark	Allows writing complex PySpark logic for data transformation (e.g., silver_layer_job.py) without managing clusters.
Crawlers	Figure out table structures on their own, then refresh the Data Catalog so you can run queries right away.

C. Checking data + showing it clearly

Amazon Athena

Motivation & Need: To check if data's correct - also to quickly explore files in S3 without installing a database.

Justification: Athena allows SQL querying directly on S3 files using the Glue Data Catalog metadata. Besides cutting down on maintenance work, serverless setups charge only when you run queries - making them a solid fit for occasional reports.

Amazon QuickSight

Motivation & Need: Business stakeholders need visual dashboards, not raw SQL query results.

Justification: QuickSight works fast since it's built right into AWS and connects straight to Athena - so no extra setup needed. Besides showing live visuals - like area maps or flow diagrams - it adjusts capacity when more people join in at once.

7. Team Member Contributions

The project got done because everyone worked together - each person took on jobs that fit their skills best.

Team Member	Role	Key Contributions
Manish	Team Lead & Cloud Architect	Set up the full "Lakehouse" system layout - chose right AWS tools. Set up user access rules. Pipeline Orchestration: Built `automate_lambda.py` and `kinesis_to_s3_lambda.py` scripts. Handled the last merge of every part.
Adarsh	Data Engineering (Ingestion & Bronze Layer)	Data Simulation: Built a script called `data.py` that creates fake but lifelike online shopping info based on India. Set up the streaming system using Amazon Kinesis Data Stream (E_com). Set up the Bronze layer in S3.

Team Member	Role	Key Contributions
Dev	Data Engineering (Processing & Visualization)	Created ETL processes using PySpark for both Silver plus Gold stages. Silver Job: Did data cleanup along with changing types plus built new features like `event_time_ist` or `margin`. Gold Job: Built detailed summaries to help spot trends. Set up Amazon Athena tables and linked them to QuickSight for live views and custom charts.

8. Results, Discussion, and Conclusion

8.1 Results

A rollout of FlashCommerce's live analytics setup led to a working auto-updated Lakehouse model. This framework handled fast-moving online sales info across three stages - Bronze, then Silver, finally Gold - sharing instant updates using Amazon QuickSight.

Metric Type	Detail	Value	Notes
Financials	Total Earnings	₹357.4 million	
	Total Profit	₹83.5 million	
	Average Profit Margin	23.37%	Showing solid financial shape.
Conversion Funnel	Views to Cart	~44.15%	Nearly half add items to their cart.
	Views to Purchase	~20.68%	About 1 out of every 5 people who check a product ends up buying it.
Product	Top Categories	Jewelry, Electronics	Brought in serious cash.
	Top Products	Diamond ring, Treadmill	Showed clear momentum.

Metric Type	Detail	Value	Notes
Regions	Hotspots	New Delhi, Mumbai	Color-coded maps accurately pinpointed sales in big cities.

8.2 Discussion

The project successfully addressed the core problem of delayed insights in traditional e-commerce reporting. By leveraging a serverless AWS architecture, FlashCommerce demonstrated several key advantages:

- **Sudden insights hit faster:** Instead of waiting on old-style daily summaries, linking Amazon Kinesis with AWS Lambda pulled data almost live.
- **Data Quality & Enrichment:** The Silver Layer - powered by AWS Glue - played a key role turning messy event logs into useful insights.
- **Scaling easily while keeping expenses low:** Using serverless tools like Athena, Glue, and Lambda lets the system manage shifting workloads automatically.

8.3 Conclusion

FlashCommerce hit its target - making live sales data easy to access for online store owners. Instead of just gathering numbers, their system sorts them fast through an AWS-powered setup. This means shopkeepers see clear visuals right away, helping them act quickly. Who'd think high-end data setups were only for big firms? With cloud tools, even smaller stores now get smart tech without the headache.

8.4 Future Scope

To boost what the platform can do, here's a look at what might come next:

- **Machine Learning:** Using machine learning to guess future sales - models pull from old data stored in the Gold zone to suggest price changes.
- **Sentiment Analysis:** Pulling in what customers say or write, then checking emotions - adds context to hard numbers from sales.
- **Mobile access:** Build a custom phone interface or plug the dashboard into merchant sites so users can check data from anywhere using their smartphones.