Secure Cloud Retail Analytics with Role-Based Access and Differential Privacy

Table of Contents

Introduction

Today, businesses heavily rely on data collection and analysis. For every business to thrive, a system which can give remote access to useful insights for contributors, while protecting confidential data and exact figures could help. In most cases, an RBAC system is required to ensure who has access to which parts of the sensitive data within a working environment. Recording Login attempts and activity in an immutable way can also help security checks.

Aims and Objectives

In this project we aim to provide a secure cloud analytics dashboard for a retail dataset (Global_superstore2.csv) which is available publicly. The dashboard is hosted on Google BigQuery. It allows different roles within a company to view certain analyses of the dataset online, depending on their position and granted level of access. The system also enforces time-based access (which hours of the day and access duration), JWT authentication (Json Web Token), audit logs for security check, and Differential Privacy (DP) for protecting exact figures.

Objectives

1. Provide a fast response API on BigQuery
2. Deploy the dashboard on a **Google Cloud Virtual Machine (VM)** to allow an easy one-click secure remote access.
3. Provide Role based access control (**RBAC**) with expirable **JWTs** (They have a life time and are provided within certain hours of the day for each role depending on position)
4. Adding **Laplace Noise (Differential Privacy)** to protect single or small number transactions when less trusted users view the data.
5. Present audit logs supervised by the security officers to monitor logins, who logged in, and failed attempts.
6. Offer an interactive dashboard (charts.js) allowing easy comparison between figures.

Concept

Users log in to the app from any IP with their credentials. The Flas app connects and send SQL queries to BigQuery and presents a dashboard with multiple tabs: Top Products, Discount vs. Profit Trend, and Anomaly Overview, and Audit Log (only visible to managers and security officers)**.** There is a DP toggle which can be turned on or off. The toggle is forced on for analysts by default (Differential Privacy is always applied). Managers and security officers are hypothetically more trusted users in authority and may choose to turn the toggle off to see exact analytics or turn it on to add noise when they intend to present the analytics to their audience and protect accurate figures.

Security and Technology Used

- Cloud Platform: **Google Cloud (BigQuery)**
- Deployment: Dashboard is hosted on a **Google Cloud VM** (Virtual Machine) on **port 80**

- Authentication and Authorization: **JWT – HS256** (**JSON Web Token)** ensures when a user logs in, they are assigned a token that proves their identity and role. This token is used for authentication in their requests. HS256 **Hash-based Message Authentication Code** with **SHA-256** is an algorithm used to sign the token. It combines a secret key with the token's data to create a unique signature. This ensures the token hasn't been tampered with.)
- **Password Storing**: **bcrypt** - Passwords are stored as very long strings of characters which are very hard to reverse (**Password Hashes**)
- Privacy: Differential Privacy through Laplace noise addition (DP_EPSILON).
- Safe Audit Log: Audit log can only be added to, not changed or edited in any way.
- **Frontend**: **Bootstrap** and **charts.js,** frontend connects to the backend using **JWT tokens** to authenticate the user and their role.
- **Configuration Setting**: **.env variables** include: **JWT Secret**, **Google Cloud Project ID**, **Retail Dataset and Table**, **DP Epsilon** (set to 0.01 for visible Laplace additions)

Used technology and security summary

	Technology	**Use**
Cloud Platform	Google Cloud Platform (BigQuery + Compute Engine)	Cloud access management
Hosting	Google Cloud Virtual Machine (VM)	Allowing the dashboard on the internet, online for remote access, port 80
Backend	Flask (Python)	JWT Authentication and Authorization, RBAC, audit log trail
Frontend	Bootstrap + Chart.js	Visuals
Authentication	JWT (HS256)	Tokens are signed to prevent from being tampered with
Password Storage	bcrypt	Password hashes (long strings)
Privacy	Laplace Noise Addition (ε - Differential Privacy)	Random noise added to exact numbers
Audit Logging	Append only	monitor failed logins and unusual use
Configuration Security	.env file	For hiding project ID and secrets

Security and Function Requirements

Function Necessities

- **Login**: Checking **credentials (password and email**), if correct granting an **expirable token** depending on role, **showing working hours, token duration, current time** on the webpage.

- **Analytics**: The dashboard provides the following analyses:
    - **Top products** in terms of profit made, grouped by region or category at the user's preference, with optional filters applied
    - **Discount vs profit trend graphs** with the option for the user to choose one or multiple graphs for better comparison, grouped by category and region
    - **Anomalies**: Anomaly counts in each group for further supervision. An anomaly is defined as transactions happened **when profit was less than 100 US dollar or discount was above 50%**, anomaly count percentage of total transactions, along with average profit for anomalous transactions are also displayed.
    - **Audit logs**: last 200 login attempts recorded with details, not editable (Restricted view for certain roles).
- **Dashboard**: allowed tabs for each role is displayed via tokens in the dashboard.

Role	Access Hours	Token Lifetime	Tabs Visible	DP Enforcement
Data Analyst	09:00–17:00	30 s	Top Products, Discount vs Profit Trend Graphs, Anomalies	Always On (Forced)
Data Manager	24 h	10 min	All Tabs	Optional (On/Off)
Security Officer	24 h	10 min	Audit Log only	Not Applicable

Security Necessities

- **RBAC (Role Based Access Control)**: Data Analyst, Data Manager, and Security Officer each have specific permissions to use the dashboard.
- **Time and Duration Based Access**:
    - Analyst Role: Only within working hours (9-17), token expires after 30 seconds
    - Manager and Security Officer Roles: 24/7 access, token expires after 10 minutes.
- **Viewing Tabs Privileges**:
    - Analyst: Top products, Profit vs Discount, Anomalies, along with all filters and drop downs
    - Security: Only the audit log tab
    - Manager: All tabs
- **Differential Privacy**:
    - Analyst: Laplace noise is always added, DP toggle is always forced on.
    - Manager: Optional DP toggle on/off, may choose Laplace noise addition for demos and presentations.
    - Security: DP does not apply to audit log monitored by the security..

- **Auditing**: Every successful or unsuccessful login is recorded, stating time, role, and action.
- **Secrets**: **.env** is used for storing sensitive information, including **JWT Secret** and **Google Cloud Project ID**

Summary of the designed RBAC

Design and Data Flow

Overview of Architecture

Client

Flask

BigQuery

Data Flow

1. User inputs password and email, and the flask checks them.
2. If correct, the flask generates a JWT token, containing role and token lifetime, and sends it back to the user's browser.
3. The browser stores this token in memory and it will be used for future when the user makes requests. It authorizes the user.
4. The user makes a request, e.g. clicking on Top Product tab, and the flask checks the token, if valid, the flask sends a query to Google BigQuery.
5. BigQuery responds and sends the requested data to the flask. In this stage, if the Differential Privacy is enabled, like the forced DP toggle for analyst role, the flask adds a random Laplace noise to the data before sending the data to the user's browser. There is no DP for audit log, as it would only change time records which would be meaningless.

Achievements

- The online dashboard successfully launches from any **public IP remotely on VM Google Cloud**
- Allows **RBAC**, different users, different privileges
- **Temporary access**: Tokens work within certain times of day for each user and expire shortly, requiring re-login, preventing potential attacks
- **Differential Privacy** protects exact individual numbers, the noise range is reasonably selected to avoid misleading graphs and trends, yet protect exact numbers
- Every **log-in attempt**, successful/failed is recorded with details and is **not editable**, allowing supervision by security team
- **BigQuery** used as the storage system
- The dashboard is **interactive** and user friendly with clear **visualizations**
- Insightful analyses are provided **grouped by different metrics**

- A **python decorator (@require_roles)** is used to keep checking the user's token, role, permissions, and access time before allowing any request.
- The dashboard design considers the potential needs of managers or security roles to choose to view actual numbers or hide them for situations like taking screenshots of graphs and presenting them to an audience. (**DP Toggle**)

Limitations

Although **DP** toggle forced on for analysts can help protect accurate numbers, users can calculate a reasonable **average number of blurred figures by repeatedly refreshing/relogging in**. since there are no limits on how many times a user can relog in, a persistent user can get close to the actual numbers.

Determining a log in budget for each user, especially less trusted users might help address this situation. A reasonable numbers of sessions allowed per day/hour according to company's needs or policies could be a better option rather than unlimited logins per day.

Deployment on Virtual Machine for Remote Access

In order to make the flask run on the internet remotely, it is set up on a **Google Cloud VM** (Virtual Machine) .

HTTP traffic is allowed for anyone trying to visit the webpage with any IP. The following files are uploaded to VM SSH to complete the **backend, frontend, requirement installation, secrets,** and connecting to **dataset in BigQuery**:

- **App.py**
- **Static/dashboard.html**
- **Requirements.txt**
- **.env**
- **Service-account.json**

After creating a virtual environment and installing the requirements and libraries, a production server **Gunicorn** is installed to allow multiple users and requests more efficiently.

To keep the flask running and keeping the website live **after closing the SSH**, a **systemd** service (a **Linux tool**) is installed to keep the app running and restarting it in case of any potential crash.

In addition, the web server **Nginx** is installed since many networks block **port 5000**, leading to many users not having access to the webpage. **Nginx** communicates with browsers on **port 80**, working anywhere in the world. It takes users' requests and conveys them to **Gunicorn on port 5000 in the VM**. Therefore, all networks can access the webpage remotely without trouble.

Creating the Firewall Rules

- Direction of traffic: **Ingress** (Incoming requests)
- Source IPv4 range: **0.0.0.0/0** (Allowing all Ips from all countries)
- Protocols and Ports: **tcp:80** (standard for HTTP sites)

The app is now **hosted on the VM and starts automatically** or reboots on itself if needed, without keeping **SSH** open. The rules allow the dashboard webpage on the internet accessible to **any IP**, so if any user, analysts, manager, or the security officer is at home or even another country, they can continue having access to the dashboard smoothly all the time.

Running the Program and Testing

In this section we test the program to check if it runs smoothly on the internet, analytics tabs are efficiently loaded, Role-based access control, and Differential Privacy function properly, the interface is user friendly and clear, and presents clear and useful data insights.

How to Access the Dashboard

Please open the following link in any browser, and from any IP address, and enter the credentials below:

http://34.142.89.12/dashboard

- Username: Analyst@retail.local
  - Password: Aida123
- Username: security@retail.local
  - Password: Aida789
- Username: Manager@retail.local
  - Password: Aida456

Please note that access for certain roles is restricted to certain hours of the day, token expiry time, viewing tabs privileges, and DP toggle functions also depend on role. You might fail to login if the time of day does not match the role, or may not see all functions if the role you choose does not have the privileges. The RBAC design summary is stated here again for clarity:

Role	Access Hours	Token Lifetime	Tabs Visible	DP Enforcement
Data Analyst	09:00–17:00	30 s	Top Products, Discount vs Profit Trend Graphs, Anomalies	Always On (Forced)
Data Manager	24 h	10 min	All Tabs	Optional (On/Off)
Security Officer	24 h	10 min	Audit Log only	Not Applicable

Analyst Role Access Validation

Below is a successful and failed login due to the designed day time constraints for the analyst role

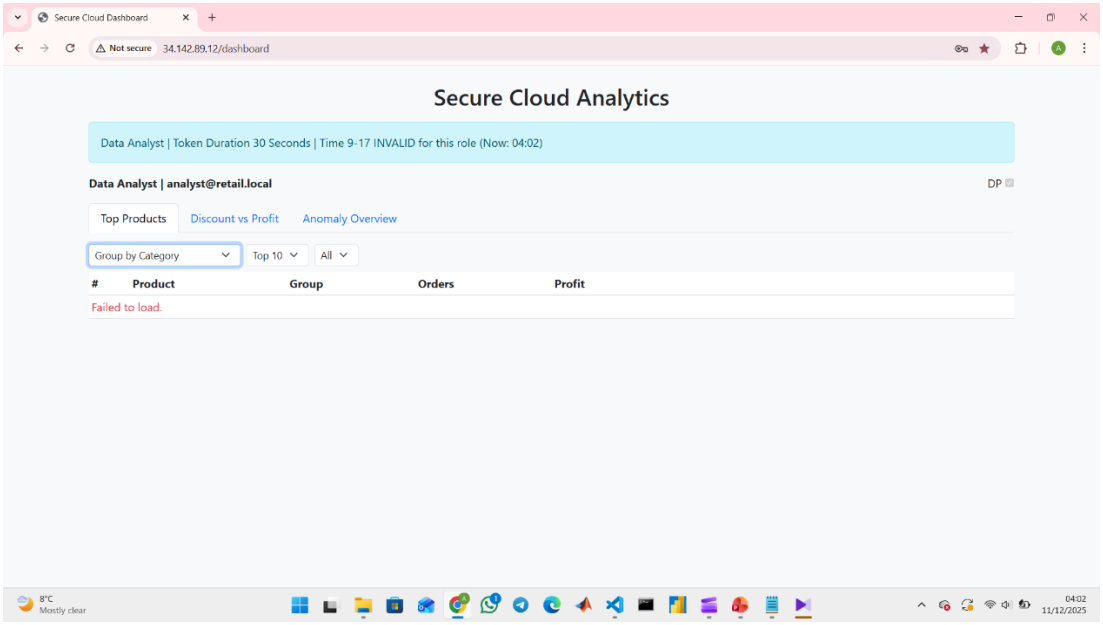


Fig 1: Failed login (outside working hours, not allowed for analysts)

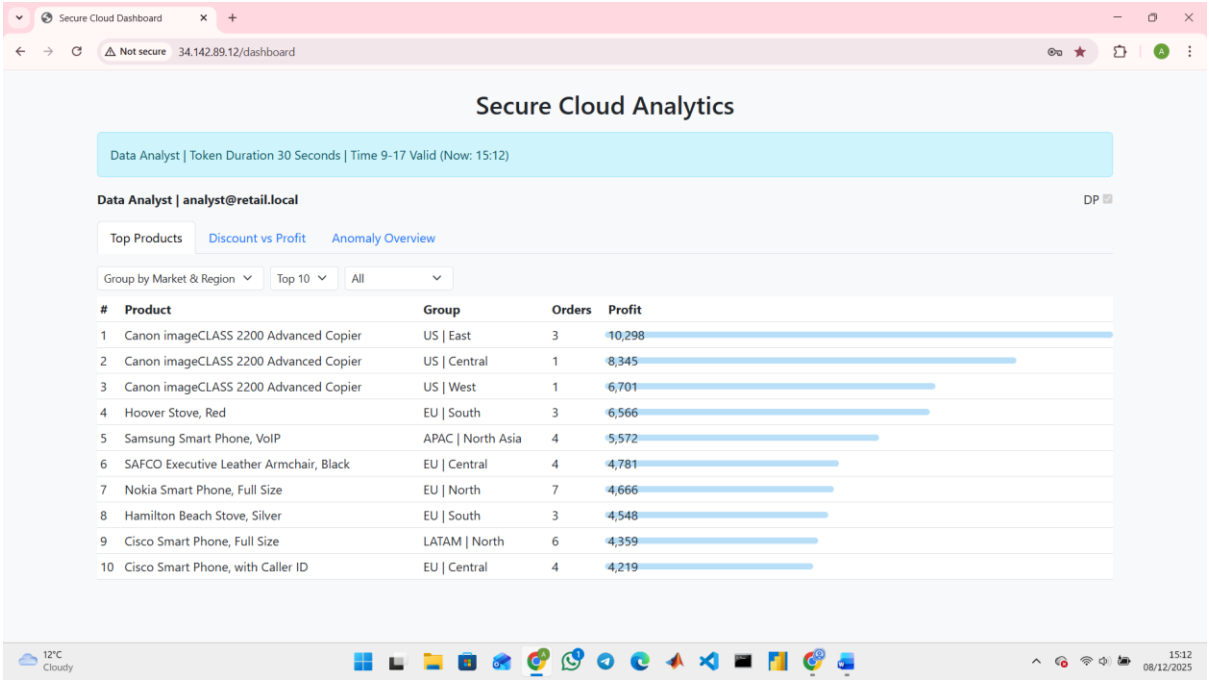


Fig 2: Successful login (withing working hours, allowed for analysts)

Please note the dashboard displays in the **blue ribbon** on the top:

- The role: Analyst
- the time (Now: 15:12)
- Time 9-17 Valid (Access allowed via token during working hours)
- Token Duration: 30 Seconds (For analysts), relog in needed in case the analysts needs to switch between tabs.

Right below the ribbon on the right, is the **DP toggle** (Differential Privacy Activation Button) which is forced on by default for this role and cannot be turned off.

Analyst role has access to three tabs, **Top Products**, **Discount vs Profit**, **Anomaly Overview**

The server is running successfully on the internet. Access control and privacy setting is applied properly for the specific role (Analyst).

Security Officer Role Access Validation

Below is a display of the dashboard upon a successful login for the security team

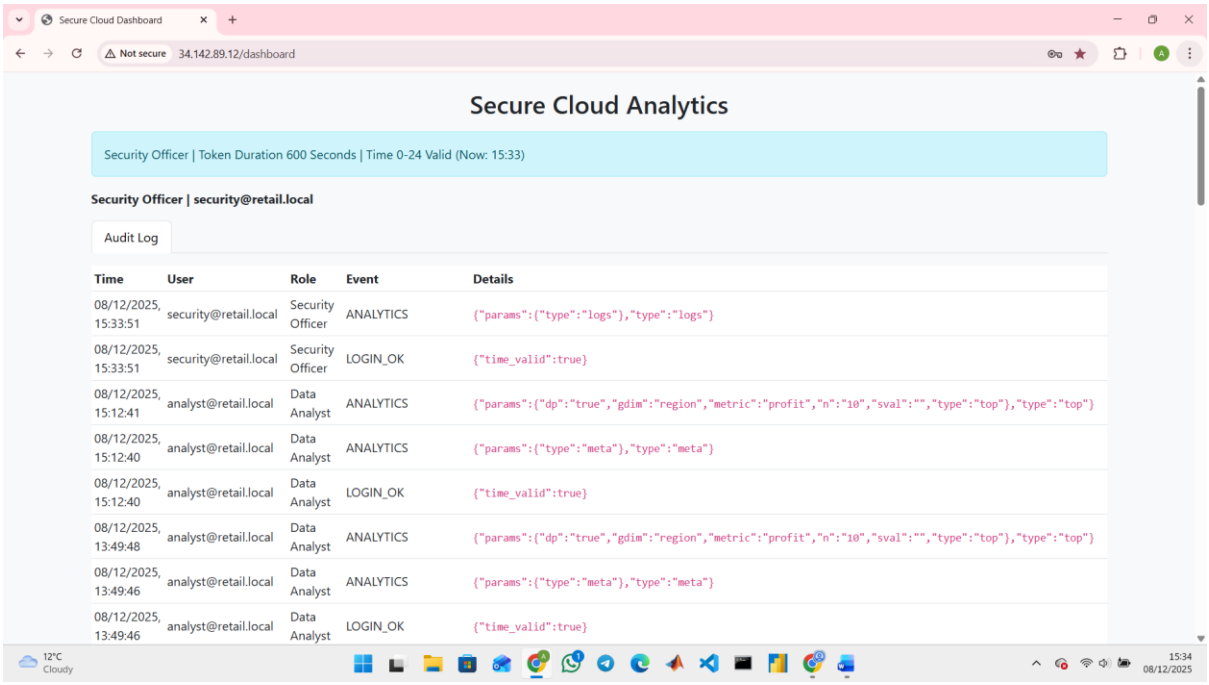


Fig 3: Audit Log tab shown for the security role

Please note the dashboard displays in the **blue ribbon** on the top:

- The role: Security Officer
- the time (Now: 15:33)
- Time 0-24 Valid (Access allowed via token 24/7)
- Token Duration: 600 Seconds (10 minutes for security officers), relogin is needed after 10 minutes

There is **no DP toggle**, as it would only add noise to time records, which is indeed not wanted. No individual data in dataset here is to be protected.

Since the **security team** is solely concerned with **who is accessing the data and when,** and **what action** is being taken, the **Audit Log** is the only tab visible to them. The system does not risk compromising the dataset privacy when it is not needed.

The server is running successfully on the internet. Access control and privacy setting is applied properly for the specific role (Security Officer).

Manager Role Access Validation

Below is a display of the dashboard upon a successful login for the manager:

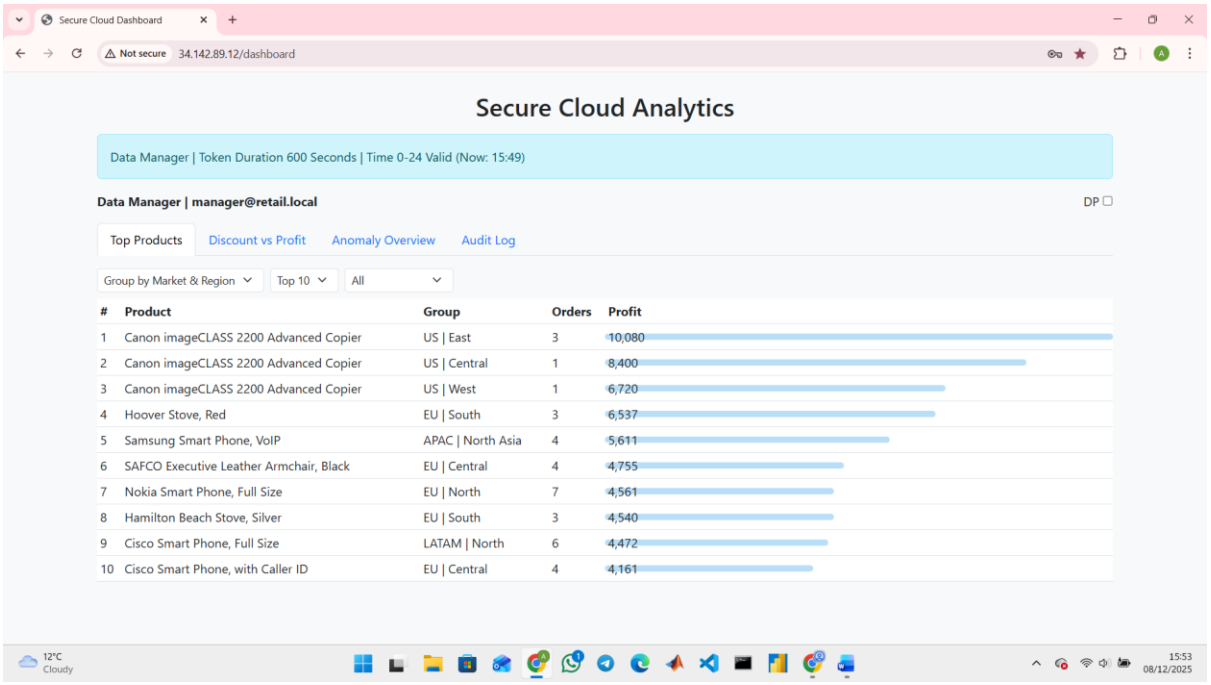


Fig 4: All tabs shown to the manager role

Please note the dashboard displays in the **blue ribbon** on the top:

- The role: Manager (highest position)
- the time (Now: 15:49)
- Time 0-24 Valid (Access allowed via token 24/7)
- Token Duration: 600 Seconds (10 minutes for manager), relogin is needed after 10 minutes

There is an **optional DP toggle**. The **manager,** as the trusted user of the data in authority, may access exact figures for better insights. They have the option **to turn the DP on for presenting the analytics to an audience** while protecting exact numbers.

The **manager** has **full access** to the whole dashboard, including all tabs.

The server is running successfully on the internet. Access control and privacy setting is applied properly for the specific role (Manager).

Differential Privacy Function Testing (DP_EPSILON TESTING)

Below is the exact figures of the profit made, only visible to the manager. We shall compare them with when the DP toggle is on to see the noise variation.

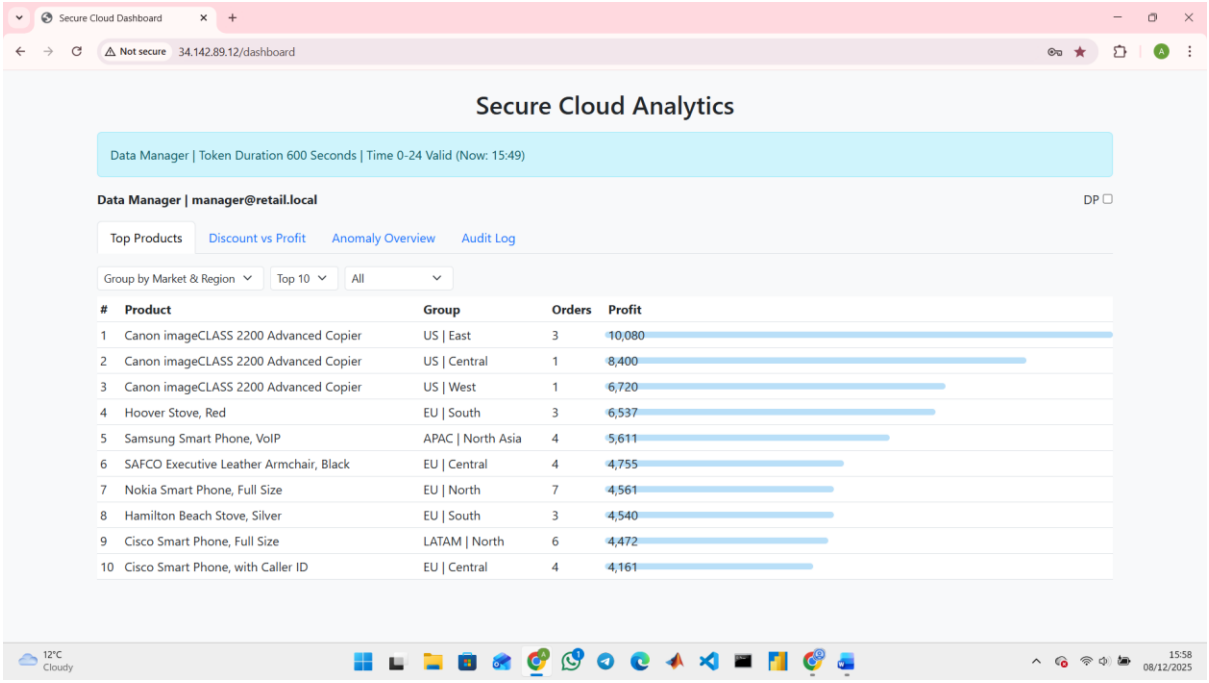


Fig 5: Real numbers for Top Products Profit tab (DP off)

Laplace Noise Mechanism

Now please note below first rows of the figures on random refreshes when the DP is on:

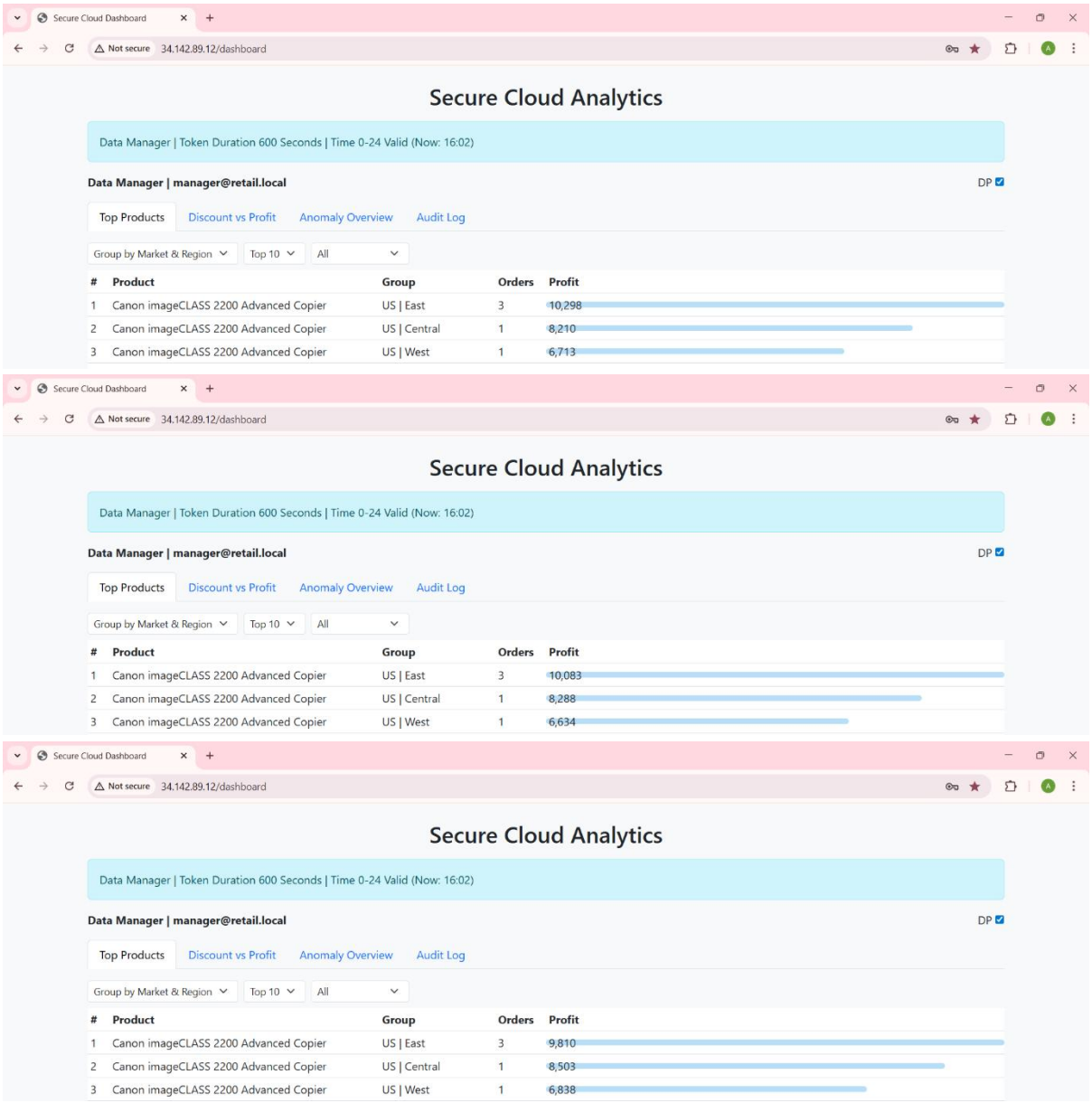


Fig 6: Blurred figures on random refreshes while DP toggle is on

Looking at the first row as an example, the exact number is 10,080 $, compromising the privacy heavily especially since there are only three items sold in this row. It can get more essential for rows two and three where there are only one items sold, revealing the very exact profit amount per product. Now with DP on, we see random numbers on different refreshes, such as 10,298$ (large positive noise), 10,083$ (very little noise), 9,810$ (rather large negative noise), helping to blur the exact profit number.

Since the noise is randomly chosen and can be large or small by chance, it makes it hard to get close to the accurate numbers.

The graphs in Discount vs Profit tab shows stable trends (Utility is preserved)

Selecting the Sweet Spot for Laplace Noise Variation Range

The above screenshots are taken while the dashboard is running on the internet publicly. To test with different Laplace Noise variations, we run the program locally and change the DP_EPSILON setting in

the .env file. As shown below, DP Epsilon is set on 0.01. This is the setting uploaded to Google Cloud VM and running on the internet.

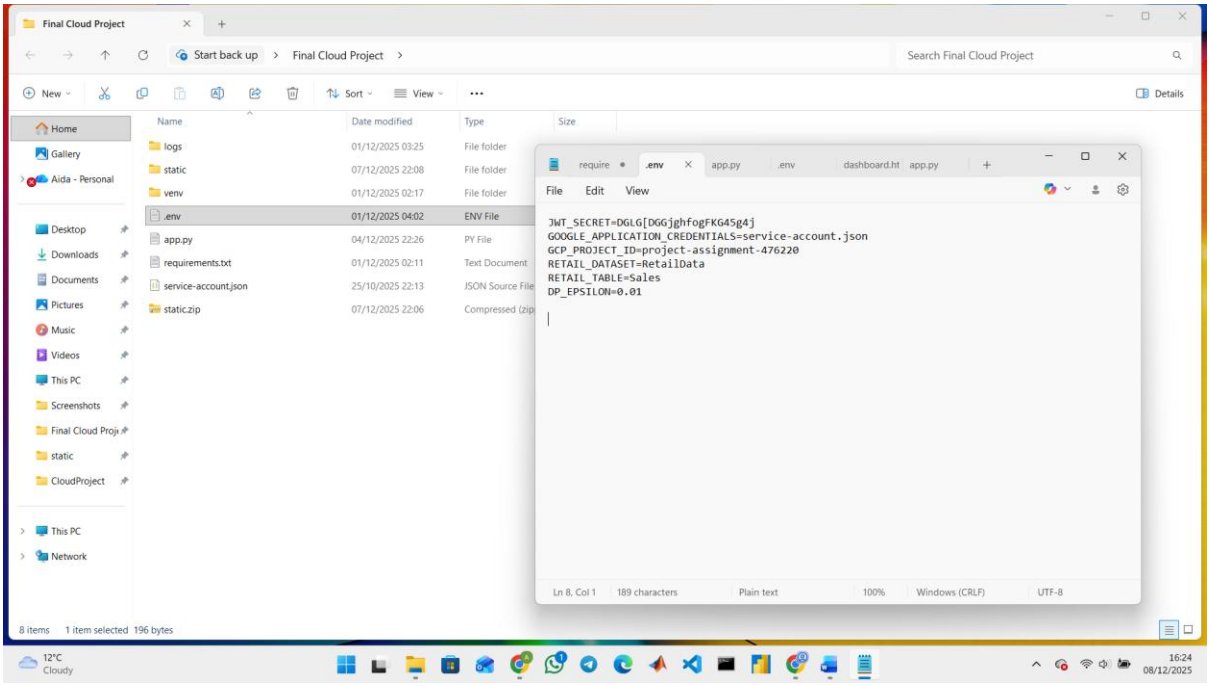
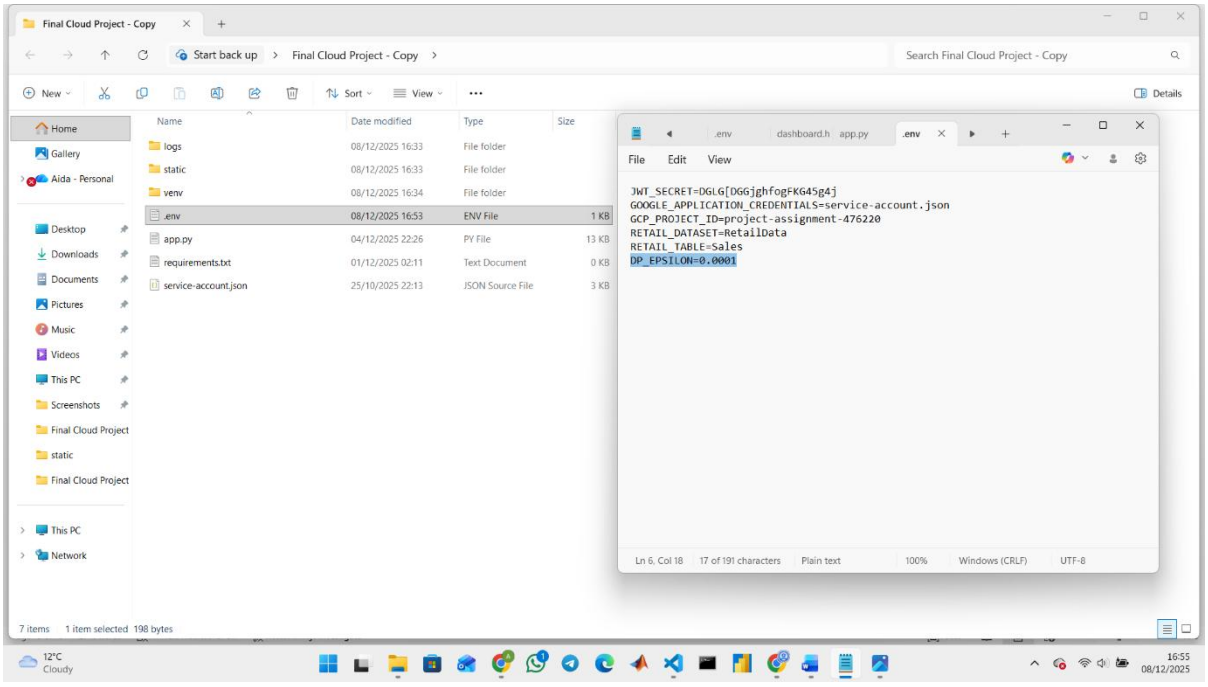


Fig 7: DP_EPSILON setting in the .env file (Local) set on 0.01

Now we change it to and see how the dashboard reacts to more or less noise.

Increasing Noise (Reducing DP_Epsilon)

Below is an exaggerated random noise increase to demonstrate how important trends are distorted and utility is damaged at the cost of privacy if the noise is too much.



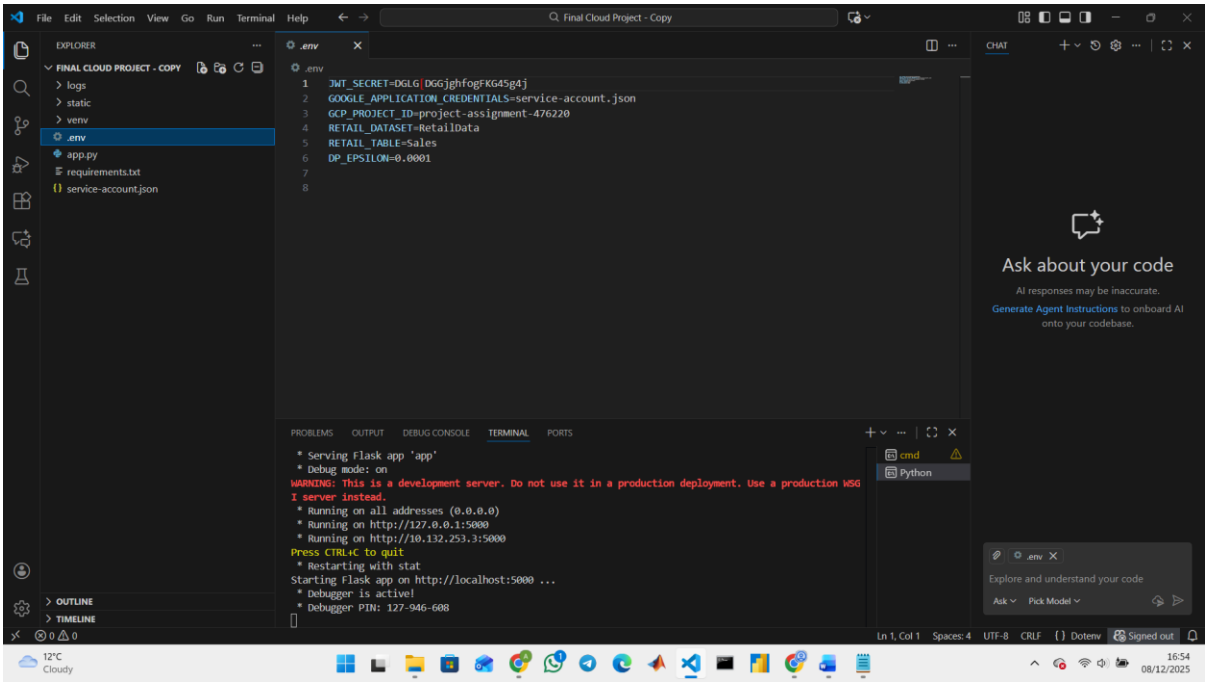
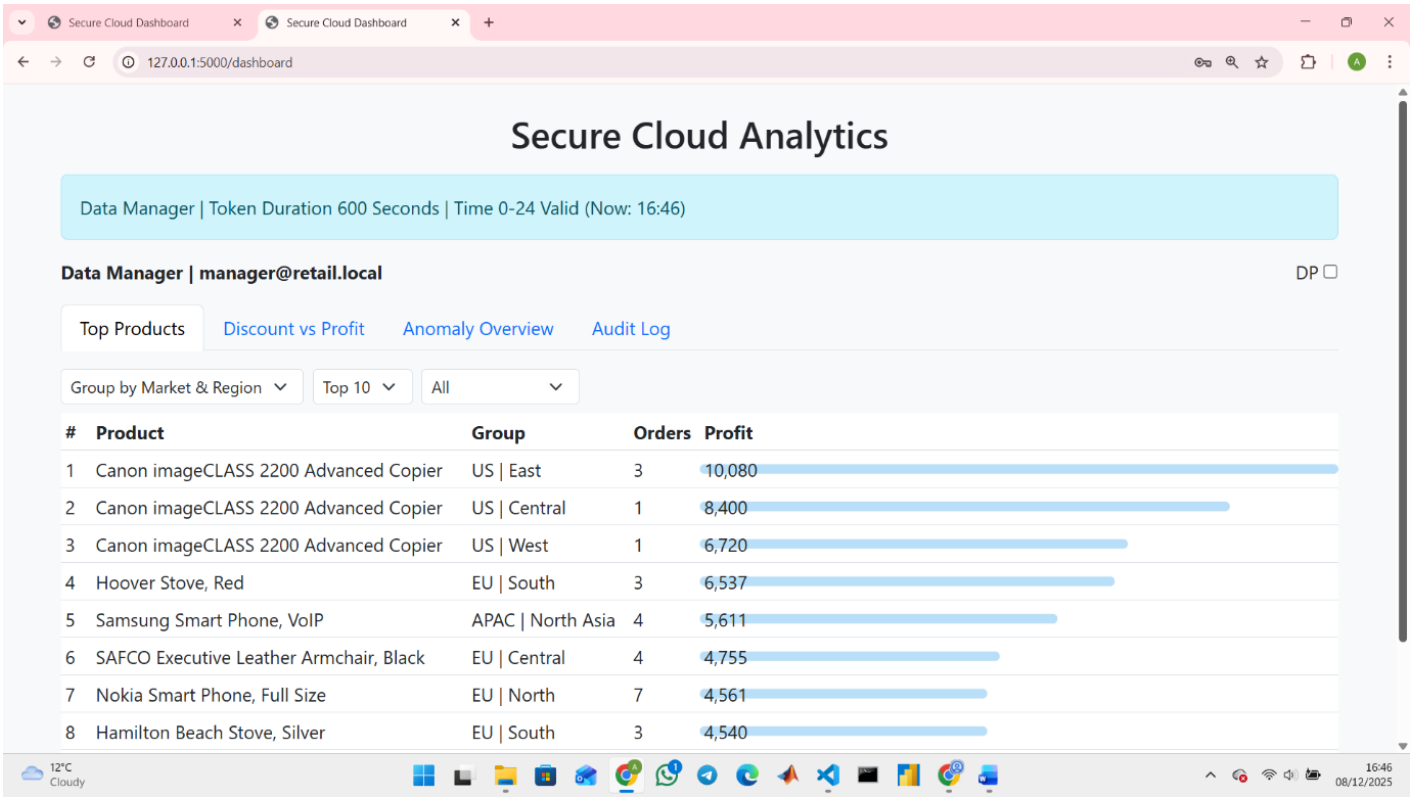
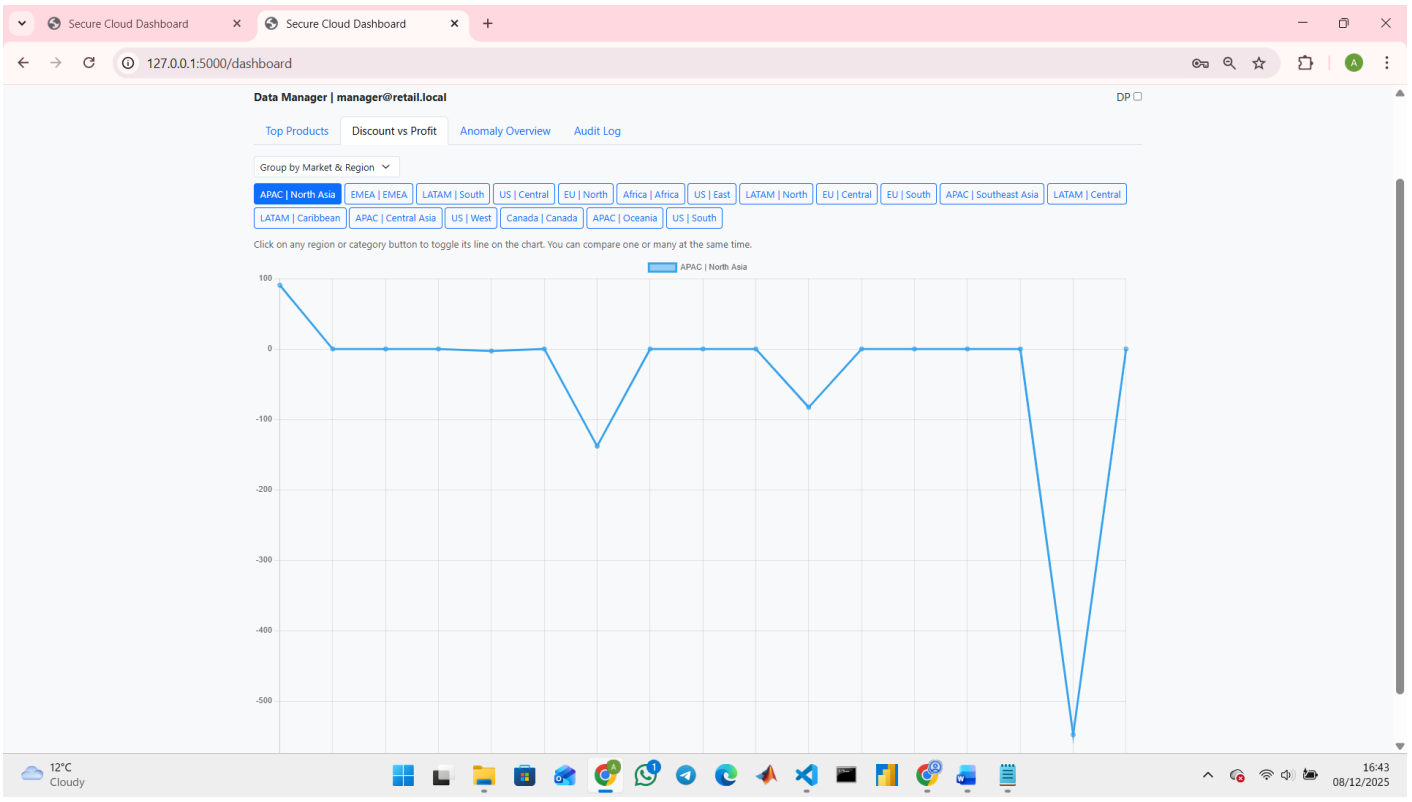
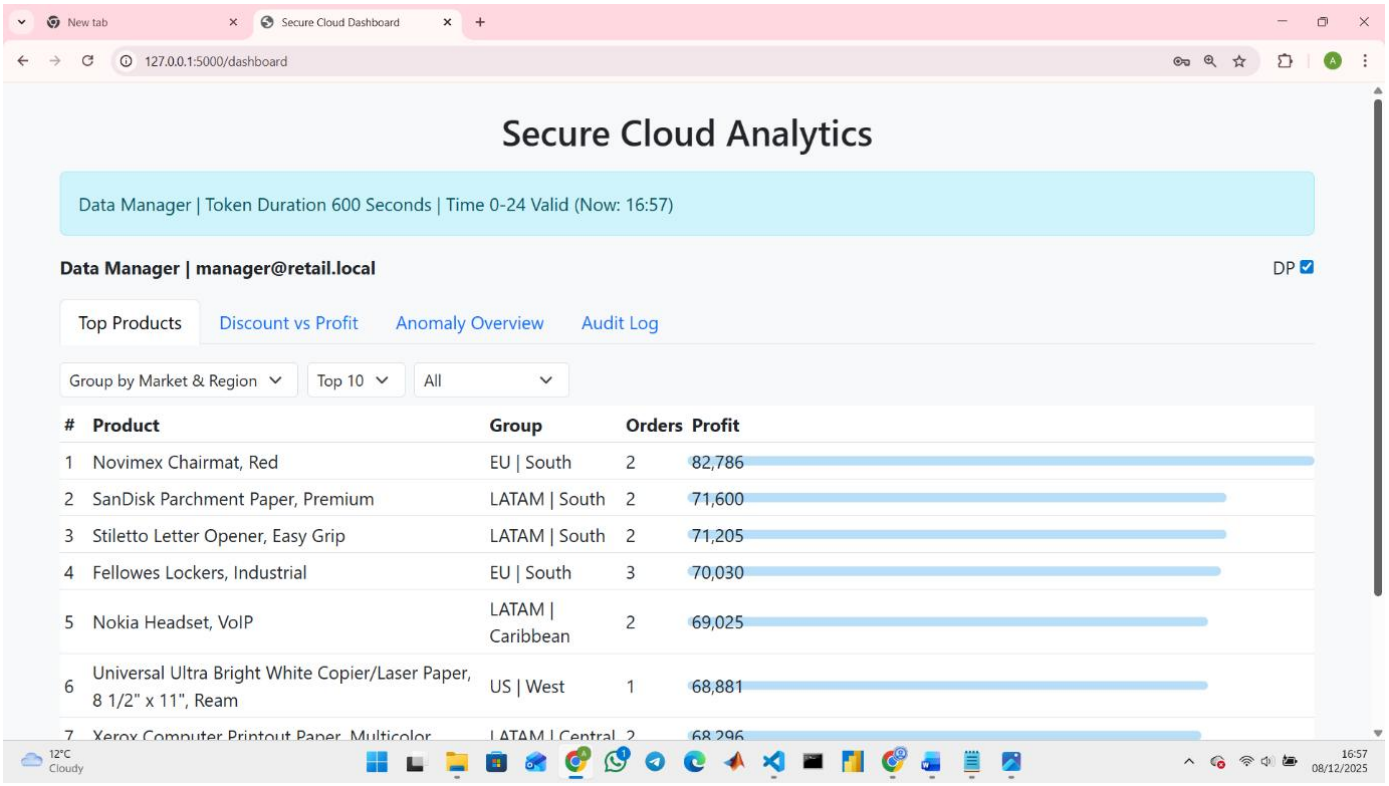


Fig 8: DP_EPSILON set to 0.0001, and the flask is run again locally, very noisy results are expected

The results below show terribly misleading trends and analytics, completely useless for business insights.


Secure Cloud Dashboard
Secure Cloud Dashboard
127.0.0.1:5000/dashboard
Secure Cloud Analytics
Data Manager | Token Duration 600 Seconds | Time 0-24 Valid (Now: 16:46)
Data Manager | manager@retail.local
DP
Top Products
Discount vs Profit
Anomaly Overview
Audit Log
Group by Market & Region
Top 10
All
Product Group Orders Profit
1 Canon imageCLASS 2200 Advanced Copier US | East 3 10,080
2 Canon imageCLASS 2200 Advanced Copier US | Central 1 8,400
3 Canon imageCLASS 2200 Advanced Copier US | West 1 6,720
4 Hoover Stove, Red EU | South 3 6,537
5 Samsung Smart Phone, VoIP APAC | North Asia 4 5,611
6 SAFCO Executive Leather Armchair, Black EU | Central 4 4,755
7 Nokia Smart Phone, Full Size EU | North 7 4,561
8 Hamilton Beach Stove, Silver EU | South 3 4,540
12°C
Cloudy
16:46
08/12/2025


Secure Cloud Dashboard
Secure Cloud Dashboard
127.0.0.1:5000/dashboard
Data Manager | manager@retail.local
DP
Top Products
Discount vs Profit
Anomaly Overview
Audit Log
Group by Market & Region
APAC | North Asia
EMEA | EMEA
LATAM | South
US | Central
EU | North
Africa | Africa
US | East
LATAM | North
EU | Central
EU | South
APAC | Southeast Asia
LATAM | Central
LATAM | Caribbean
APAC | Central Asia
US | West
Canada | Canada
APAC | Oceania
US | South
Click on any region or category button to toggle its line on the chart. You can compare one or many at the same time.
APAC | North Asia
100
0
-100
-200
-300
-400
-500
12°C
Cloudy
16:43
08/12/2025



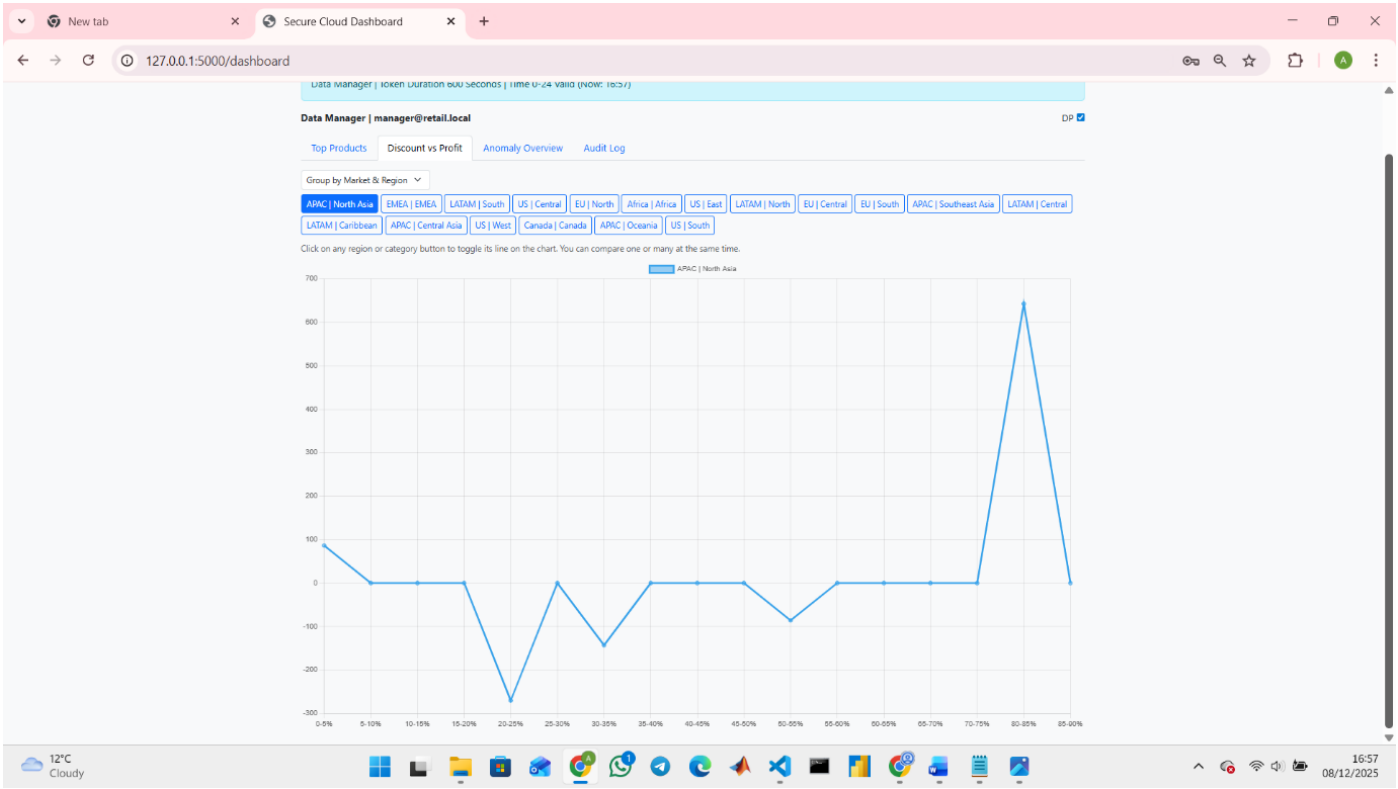
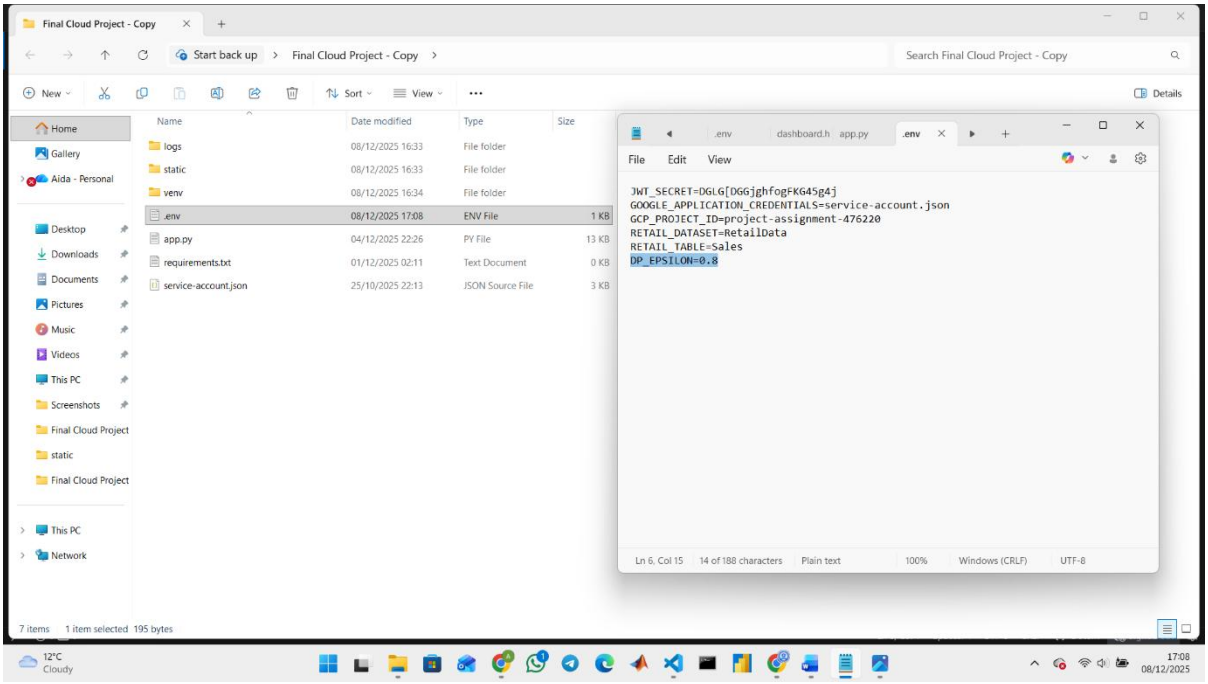


Fig 9: Differential Privacy off (real figures shown in the first two screenshots) vs Differential Privacy On (DP_EPSILON= 0.0001), very large noise, shown in the second two screenshots

Now we test with DP_EPSILON=0.8, small noise



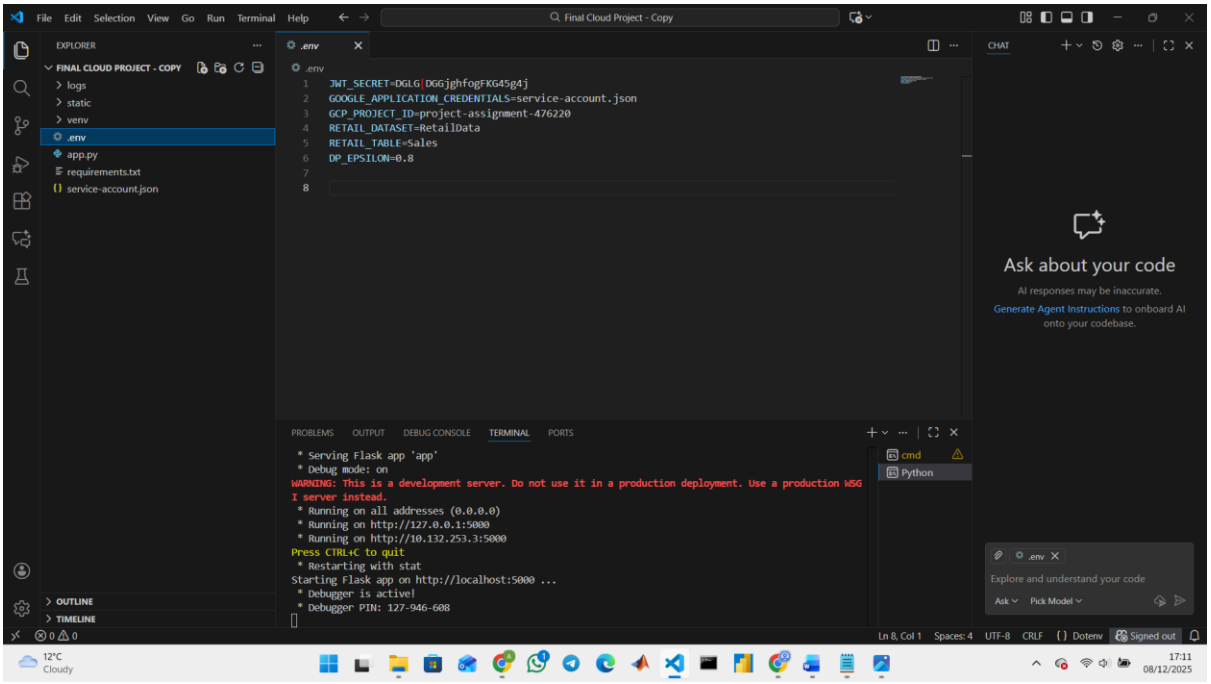


Fig 10: DP_EPSILON set to 0.8, and the flask is run again locally, small noise is applied

Below are the results of the selected noise:

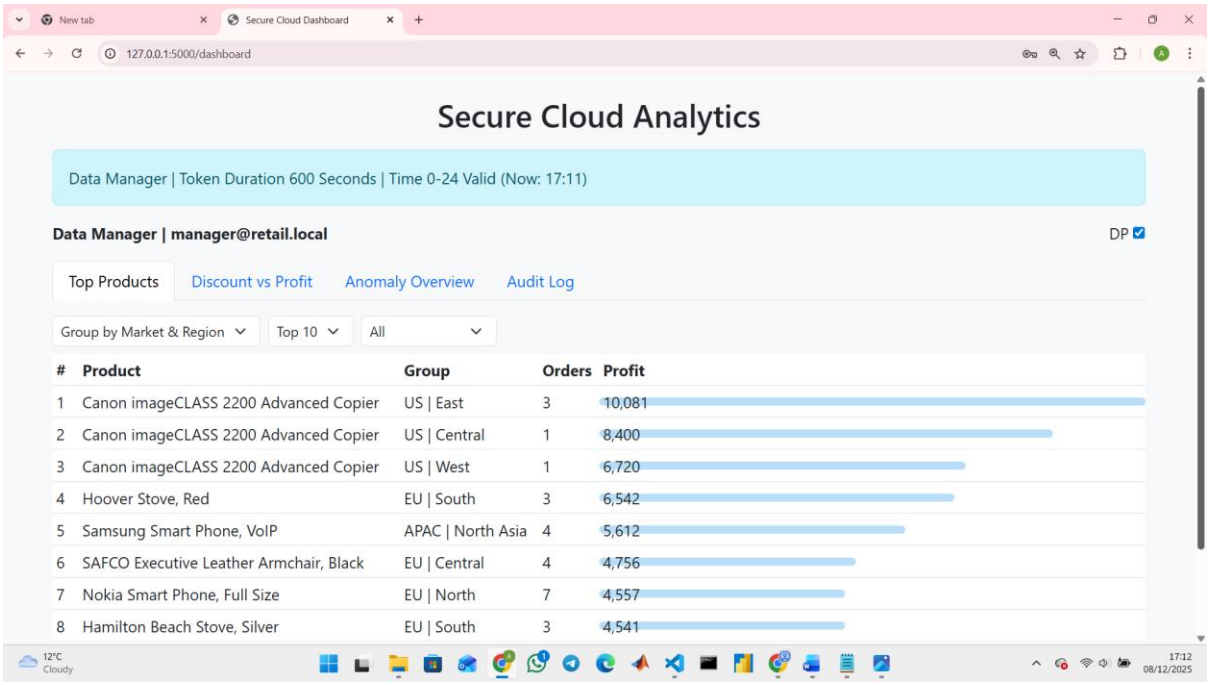


Fig 11: Results when Differential Privacy On (DP_EPSILON= 0.8)

This shows very little difference between the exact and blurred figures, e.g. 10,081 instead of 10,080. This perfectly preserves utility and trends are almost untouched. But there is almost no privacy.

Choosing DP_EPSILON Conclusion

Choosing DP_EPSILON=0.01 as the setting (.env) seems to be a decent choice to balance privacy and utility. This is the final setting chosen and uploaded to Google VM for remote access on the internet.

The Reasoning Behind the Designed Privacy Policies for Users

This project aims to act as a sample solution suggestion for businesses who want analysis, access control, and privacy altogether. Certain designed policies and control settings might be unrealistic in real world businesses and are implemented for demo only. They suggest that based on business orientation and needs, settings can be adjusted for best performance. For example,

- The dataset used here is not sensitive, and publicly available
- The token duration is a suggestion to force relog in to prevent from cyber attacks in case credentials are compromised while users are online. They should be adjusted realistically. (30 second token duration for analyst is unrealistic and chosen for demo purposes here, though it can be adjusted in real businesses depending on their policies)
- The working hours restriction for data access is another suggestion to avoid unnecessary login and token compromise when a user does not intend to actually work on the dashboard. This again can be more realistically adjusted in real world situations.

What the analyses Achieve

This sample project does not go into details of analytics. The four tabs designed, Top Products Proft, Discount vs Profit, Anomalies, and the Audit Log are common analytics used in many businesses. Here is a very brief sample interpretation and solution only for sample suggestions of how such systems can actually help businesses:

- **Top Products Profit**: demonstrates high profitability for Canon copiers
- **Top Category Profit:** demonstrates higher profitability for technology products overall
- **Regional Performance**: shows the US markets as more profitable
- **Discount vs Profit Trend**: The trend is for the profit to drop as discount increases. We see steeper losses especially when discount goes beyond 30-40%.
- **Anomalies by Region**: EMEA (Europe, Middle East, Asia) and Africa regions show remarkable anomalous transactions, could be due to pricing or operational issues, and might need extra supervision
- **Anomalies by Category**: Over half the anomalies happen in the category of office supplies, monitoring is advised.
- **Average Anomalous Transaction Profit**: stays negative across all groups. It is noticeable, though that the most negative figure belongs to Central Asia region. Supervision might be needed.