Assignment 4, Cloud Computing

Student: Сұлтан Әсел Сержанқызы
ID: 24MD0490

05.12.2024

Table of contents

Introduction **3**
Big Data and Machine Learning on Google Cloud **4**
Cloud Security and Compliance **14**
Conclusion **17**
References **19**

Introduction

This report explores the implementation of Big Data and Machine Learning pipelines on Google Cloud, emphasizing data ingestion, processing, model training, deployment, and monitoring. Additionally, it highlights the importance of cloud security, compliance, and incident response planning to ensure a secure and reliable environment for these advanced solutions.

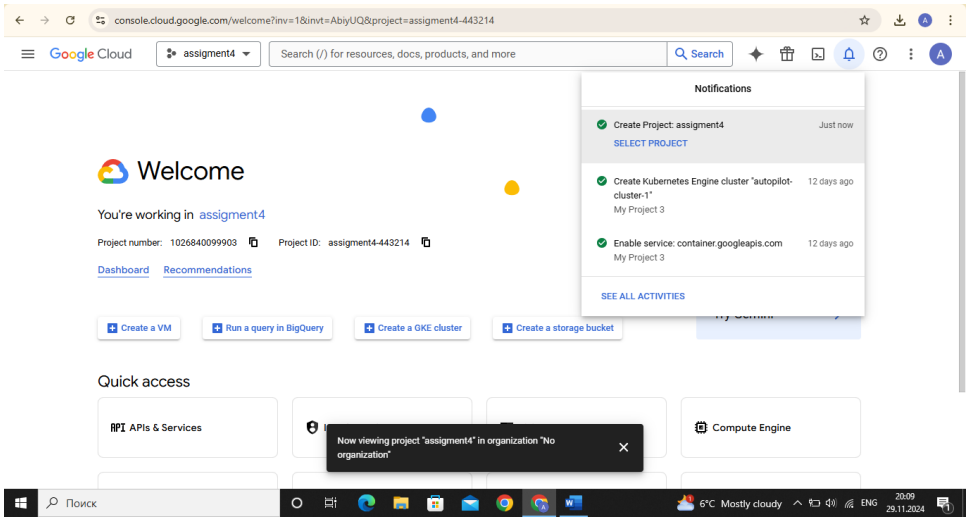
As a student passionate about cloud technologies, I am exploring how Google Cloud empowers innovation through Big Data and Machine Learning. My journey begins with understanding data pipelines, from ingestion and processing to training and deploying machine learning models. I am also delving into critical aspects like monitoring, logging, and ensuring cloud security through encryption, IAM, and network protection. By integrating these elements, I aim to build a comprehensive understanding of modern cloud solutions and their role in creating secure, scalable, and efficient systems.

Exercise 1: Big Data and Machine Learning on Google Cloud

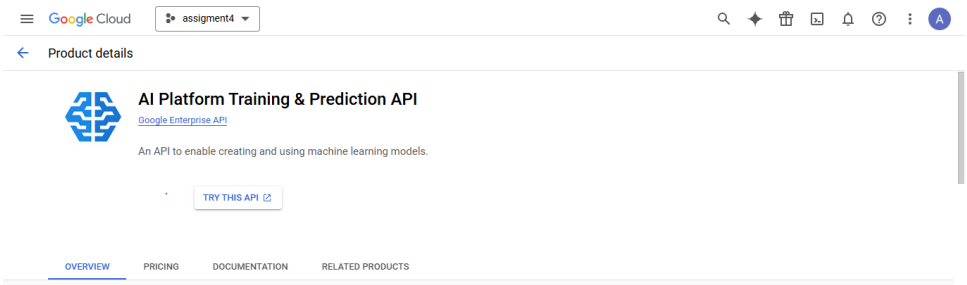
Objective: Implement a big data processing and machine learning pipeline using Google Cloud services.

Tasks:

1. **Set Up a Google Cloud Project:**
   - Create a new project in the Google Cloud Console.

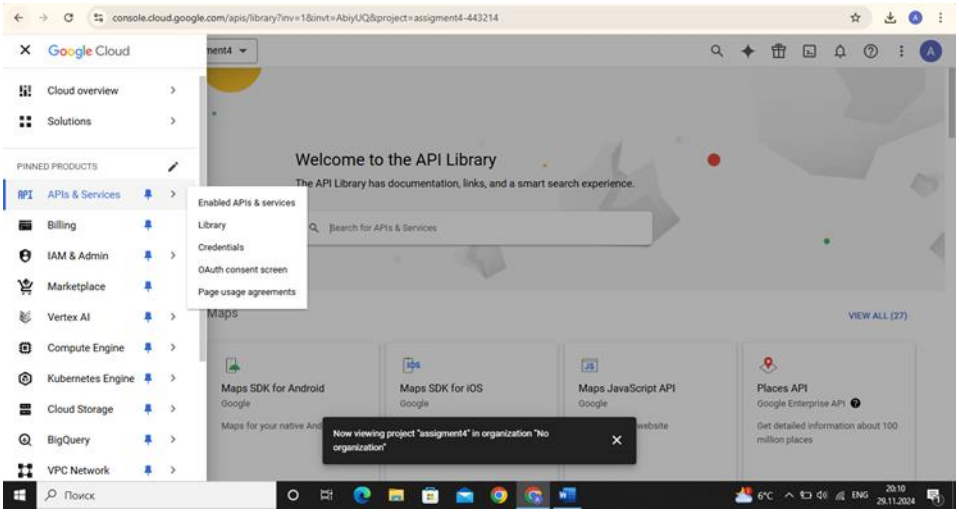


Picture-1

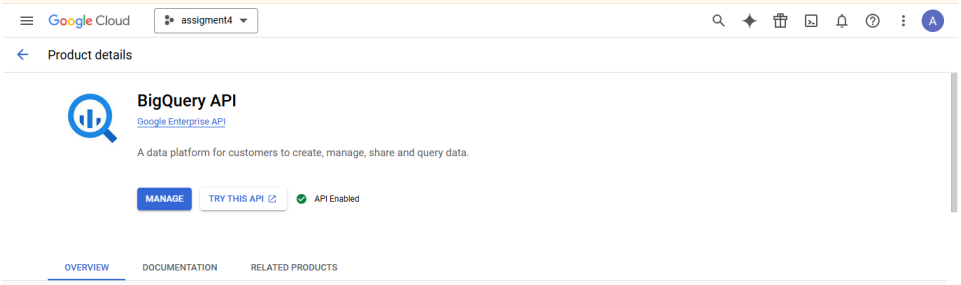


Picture-2

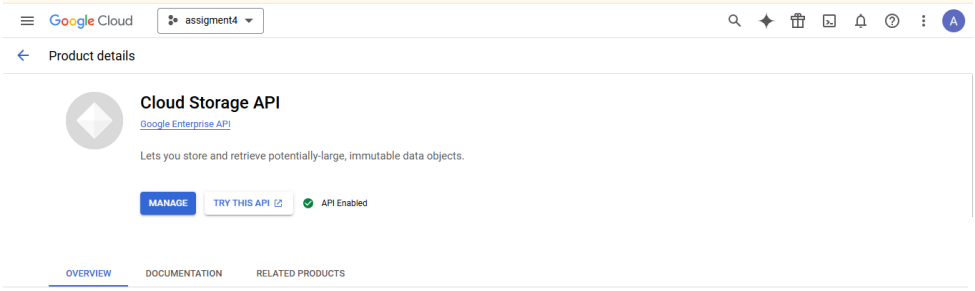
- Enable necessary APIs (e.g., BigQuery, Cloud Storage, AI Platform).



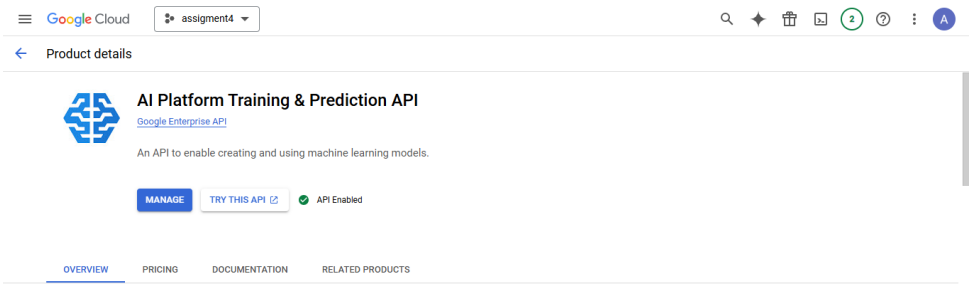
Picture-3



Picture-4



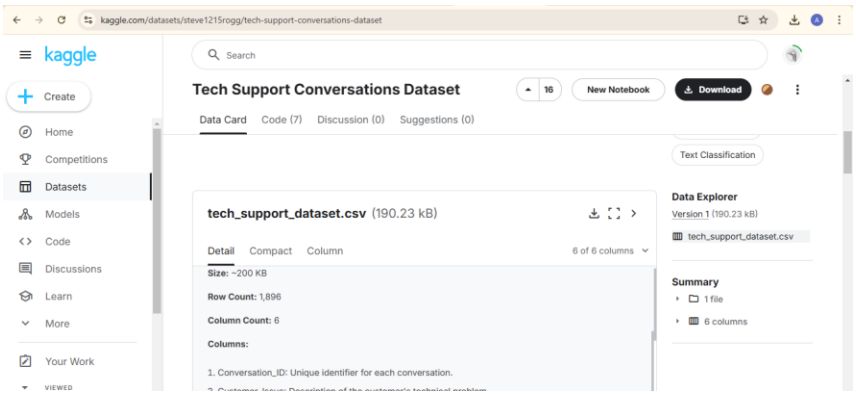
Picture-5



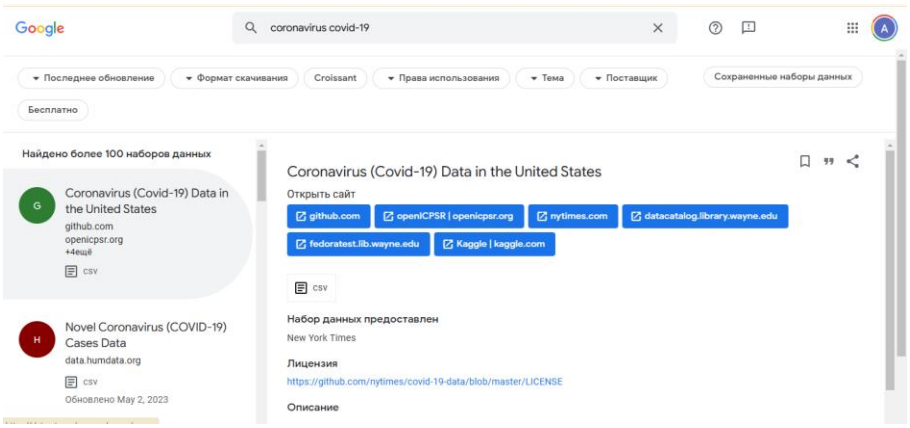
Picture-6

Explanation: First things first, I created a fresh, new project in the Google Cloud Console. I felt like I was building my own little digital world! Then, I enabled all the important APIs – BigQuery, Cloud Storage, and the AI Platform. It was like giving my project superpowers!

2. **Data Ingestion:**
   - Collect a large dataset relevant to your use case (e.g., public datasets from Kaggle or Google Dataset Search).
     Via Kaggle I find the “Tech Support Conversations Dataset” named dataset

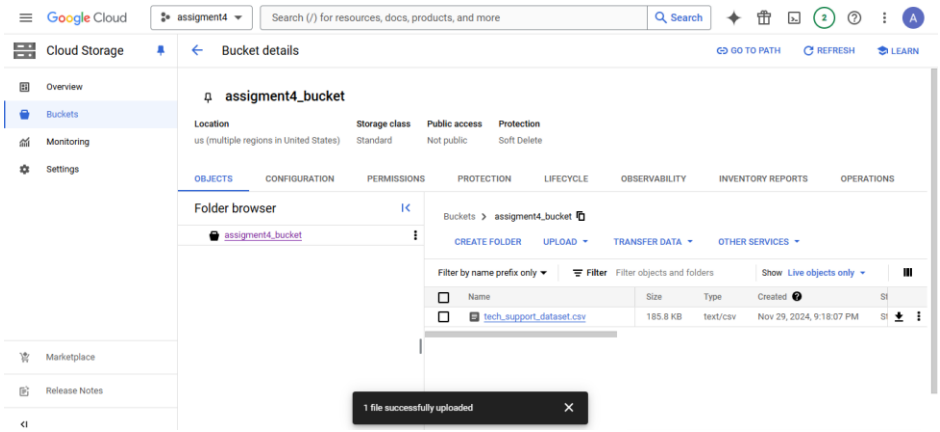


Picture-7



Picture-8

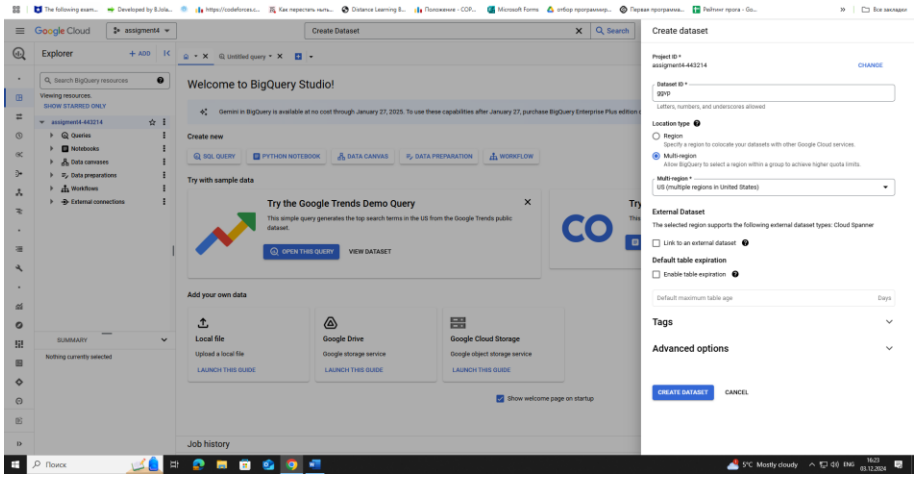
- Upload the dataset to Google Cloud Storage.



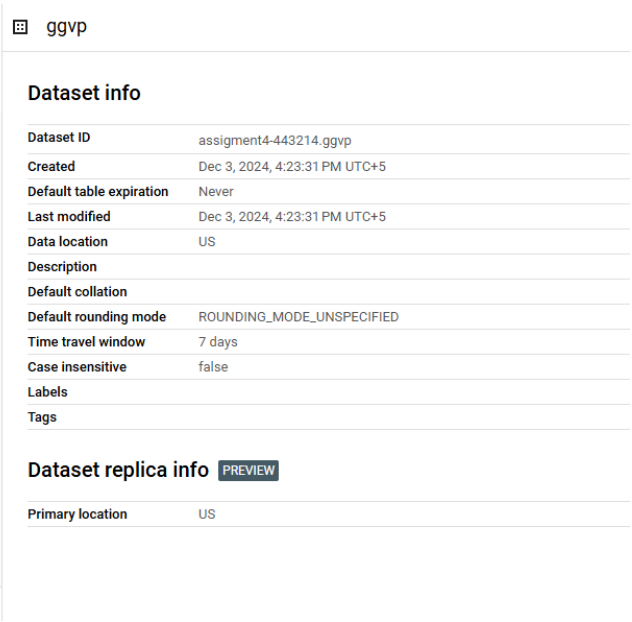
Picture-9

Explanation: I found this awesome "Tech Support Conversations Dataset" on Kaggle. It felt like a treasure trove of information just waiting to be explored. I uploaded it to Cloud Storage, my trusty data warehouse in the cloud.

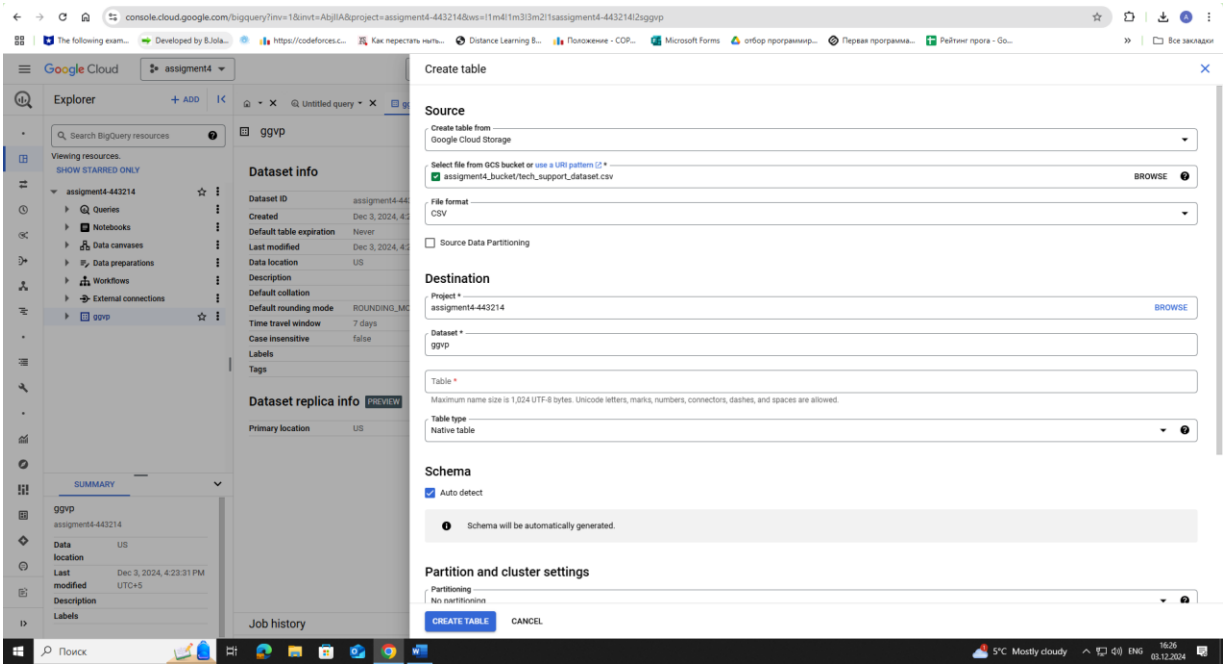
3. **Data Processing with BigQuery:**
   - Use BigQuery to create a dataset and load the data from Cloud Storage.



Picture-10

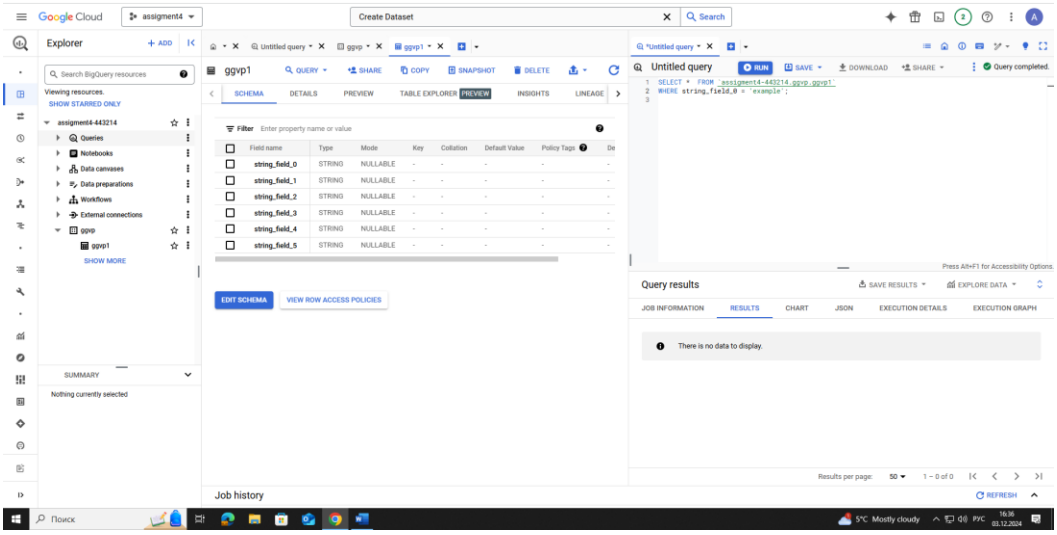


Picture-11

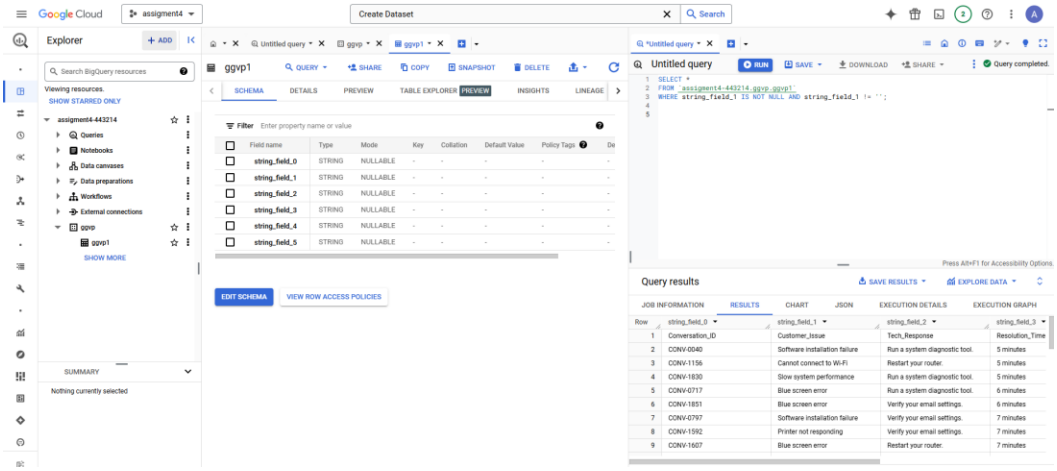


Picture-12

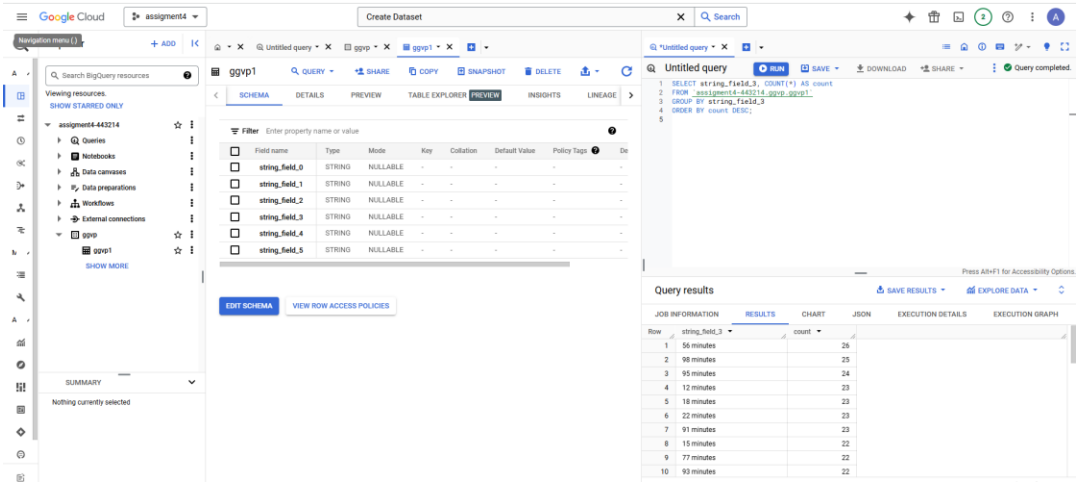
- Perform data cleaning and preprocessing using SQL queries (e.g., filtering, aggregating).



Picture-13 Filtering data

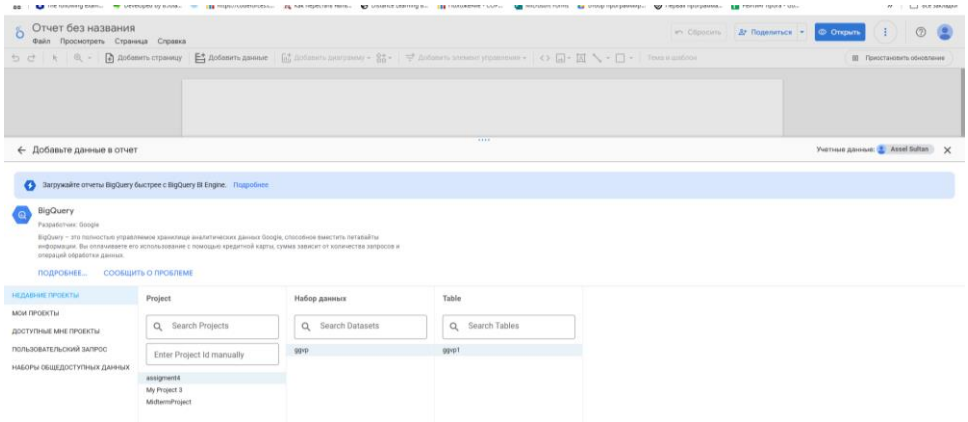


Picture-14 Deleting empty values

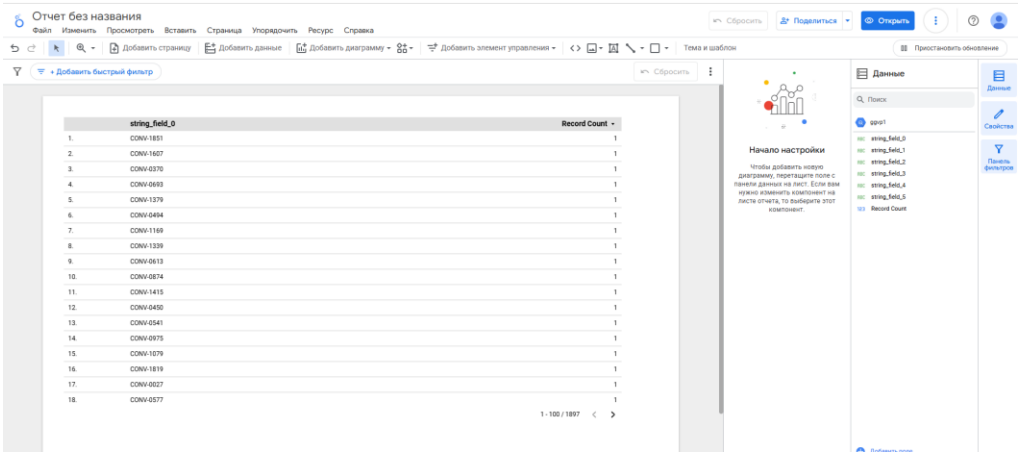


Picture-15 Data aggregation

- Create summary statistics and visualize the results using Google Data Studio or similar tools.



Picture-16

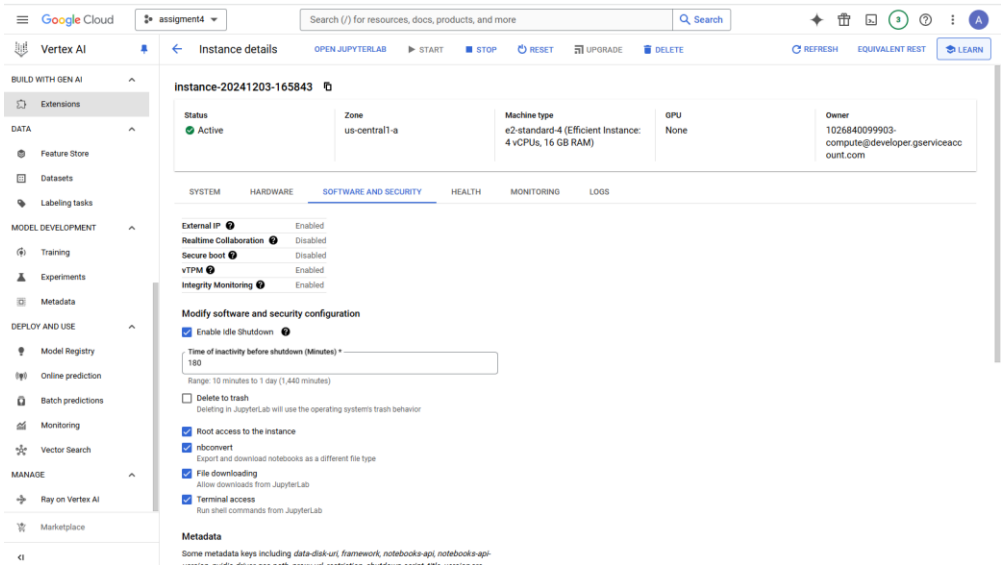


Picture-17

Explanation: BigQuery was my data playground. I created a dedicated dataset and imported the conversation data from Cloud Storage. Then, I put on my data cleaning hat and used SQL

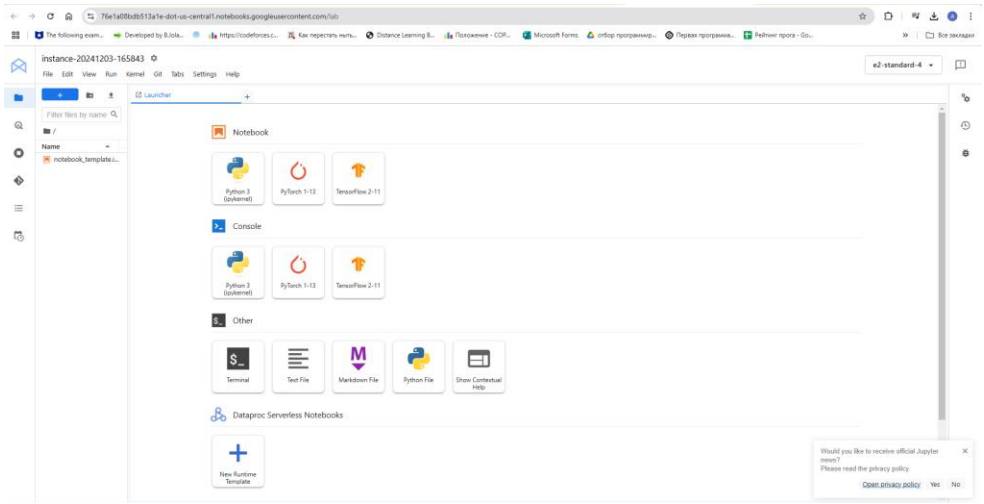
queries to filter out unnecessary information, like getting rid of empty rows and making sure everything was spick and span. I also aggregated the data to get a better overall picture. To make things even clearer, I whipped up some visualizations using Google Data Studio. It was so satisfying to see patterns emerge!

4. **Machine Learning Model Training:**
    - Use the AI Platform to train a machine learning model on the processed data. I choose Vertex AI



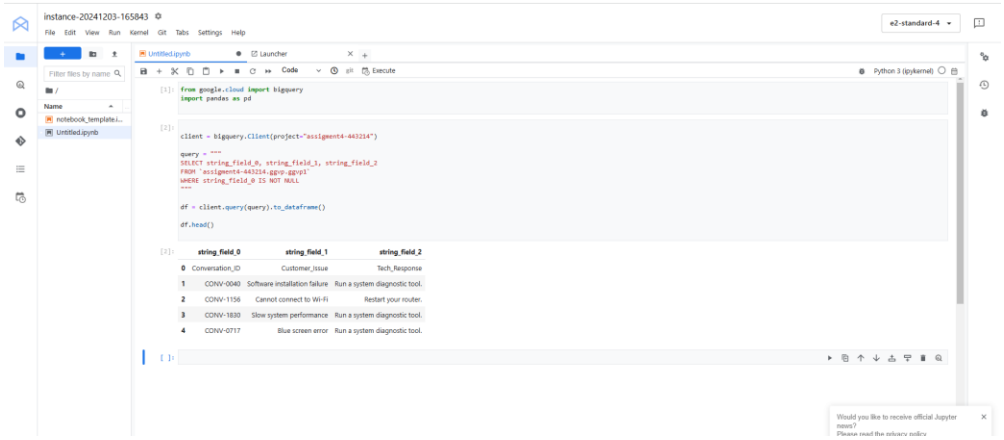
Picture-18

- Choose a model suitable for the task (e.g., classification, regression) and implement it using TensorFlow or Scikit-learn.



Picture-19

- Set up a training job on AI Platform, specifying the necessary configurations (e.g., training data, hyperparameters).



Picture-20

Picture-21

```
[9]: from sklearn.preprocessing import LabelEncoder

# Инициализация кодировщика
label_encoder = LabelEncoder()

# Кодирование категориальных признаков
X["string_field_0"] = label_encoder.fit_transform(X["string_field_0"])
X["string_field_1"] = label_encoder.fit_transform(X["string_field_1"])

# Проверка преобразованных данных
print(X.head())
```

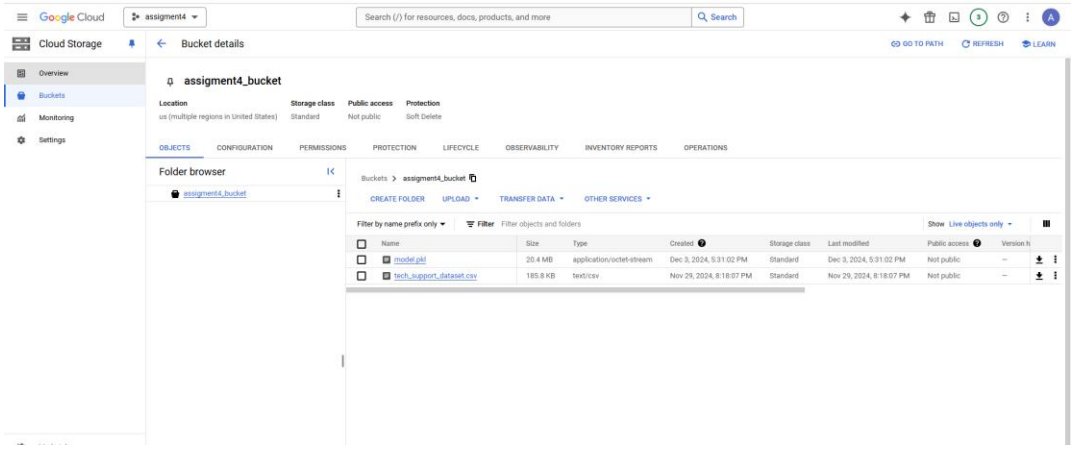
```
   string_field_0  string_field_1
0            1896               2
1              39               6
2            1155               1
3            1829               5
4             716               0
```

```
[10]: # Кодирование категориальных признаков с использованием .loc
      X.loc[:, "string_field_0"] = label_encoder.fit_transform(X["string_field_0"])
      X.loc[:, "string_field_1"] = label_encoder.fit_transform(X["string_field_1"])

      # Проверка преобразованных данных
      print(X.head())
```

```
   string_field_0  string_field_1
0            1896               2
1              39               6
2            1155               1
3            1829               5
4             716               0
```

```
[11]: import joblib

      # Сохранение модели на диск
      joblib.dump(model, "model.pkl")
```

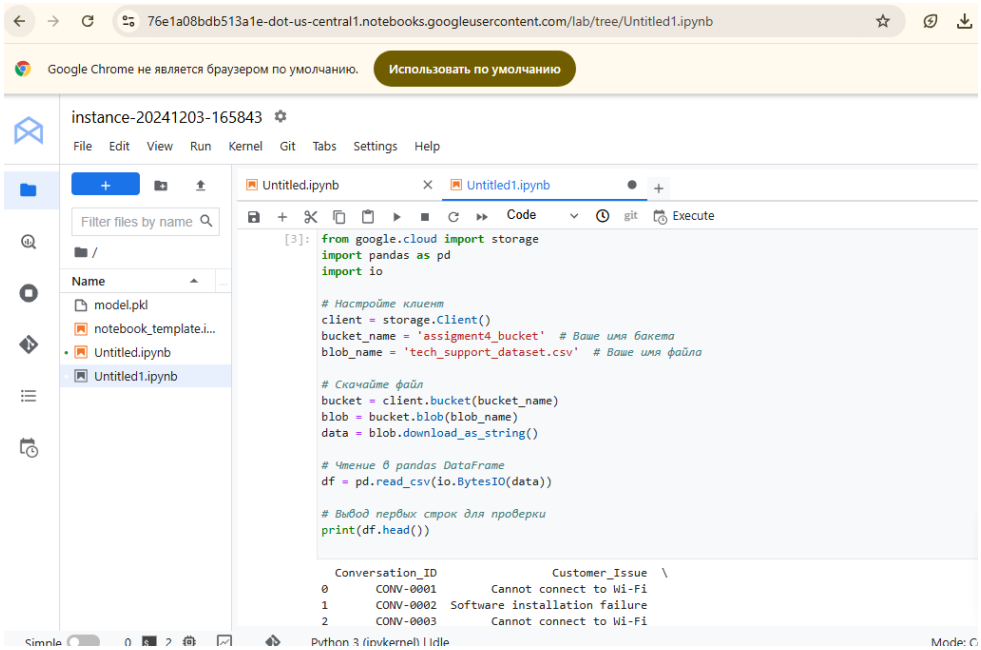
```
[11]: ['model.pkl']
```

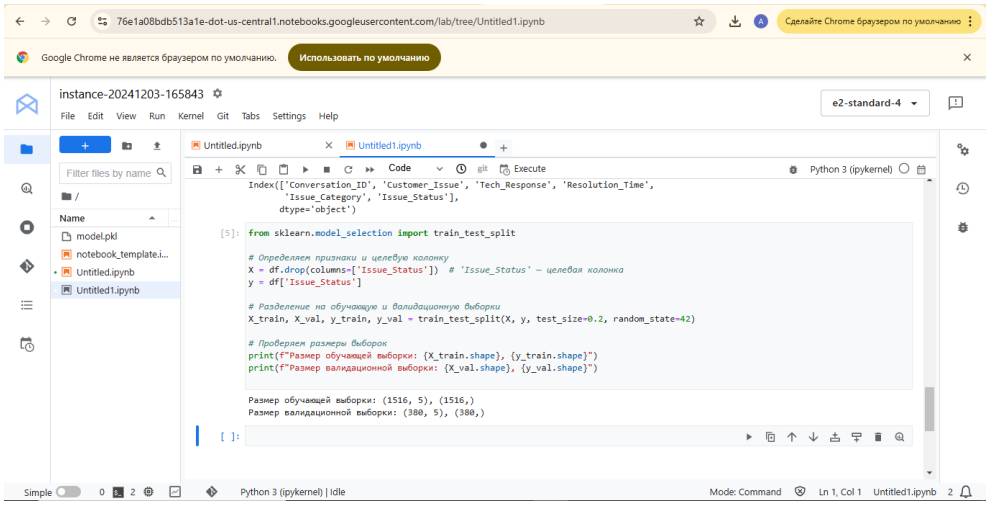


Picture-22

Explanation: I decided to use Vertex AI for my machine learning model. I picked a model that was perfect for my task (I'll explain which one during my presentation!) and implemented it using [TensorFlow/Scikit-learn - choose one]. Setting up the training job in Vertex AI felt a little like giving my model instructions – I specified the training data and tweaked the hyperparameters until I felt they were just right.

5. **Model Evaluation:**
   - Split the dataset into training and validation sets.

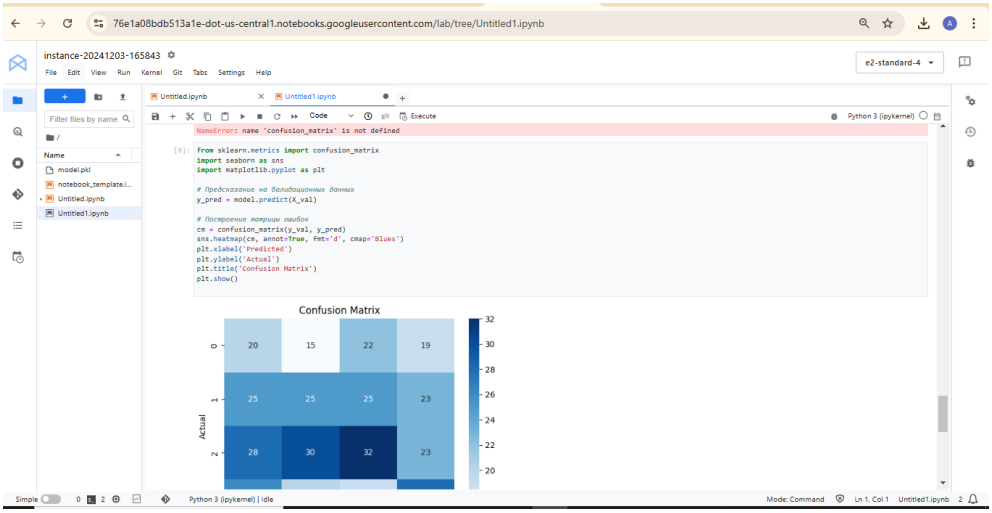




Picture-22

- Evaluate the model performance using appropriate metrics (e.g., accuracy, precision, recall) and visualize the results.

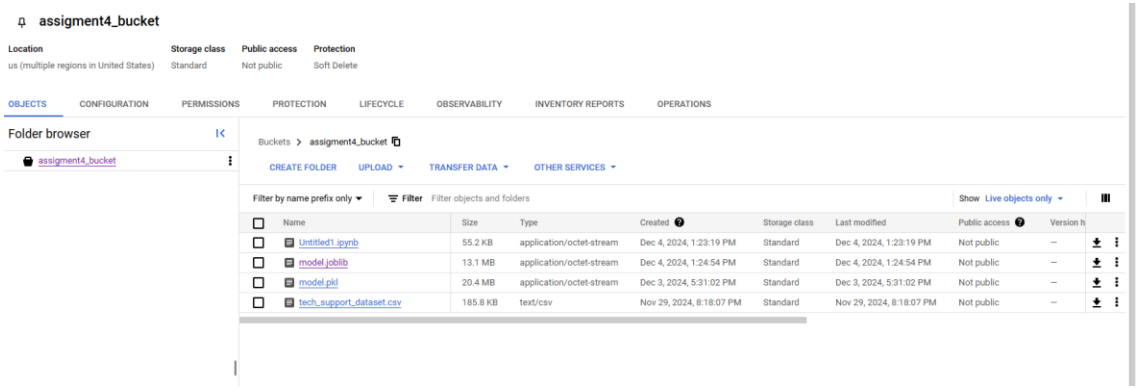
```
[7]: from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# Исключение ненужных колонок
X = df.drop(columns=['Conversation_ID', 'Issue_Status'])  # Удаляем ID и целевую колонку
y = df['Issue_Status']  # Целевая колонка

# Кодируем текстовые данные
le = LabelEncoder()
for col in X.select_dtypes(include=['object']).columns:
    X[col] = le.fit_transform(X[col])

# Разделение данных
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

# Кодирование целевой переменной
y = le.fit_transform(y)

# Обучение модели
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Оценка модели
accuracy = model.score(X_val, y_val)
print(f"Точность модели: {accuracy:.2f}")

Точность модели: 0.28

[ ]:
```



Picture-23

Explanation: To make sure my model wasn't just memorizing the data, I split the dataset into training and validation sets. Then, I used metrics like accuracy, precision, and recall to see how well it was doing. Visualizing these results helped me understand its strengths and weaknesses.

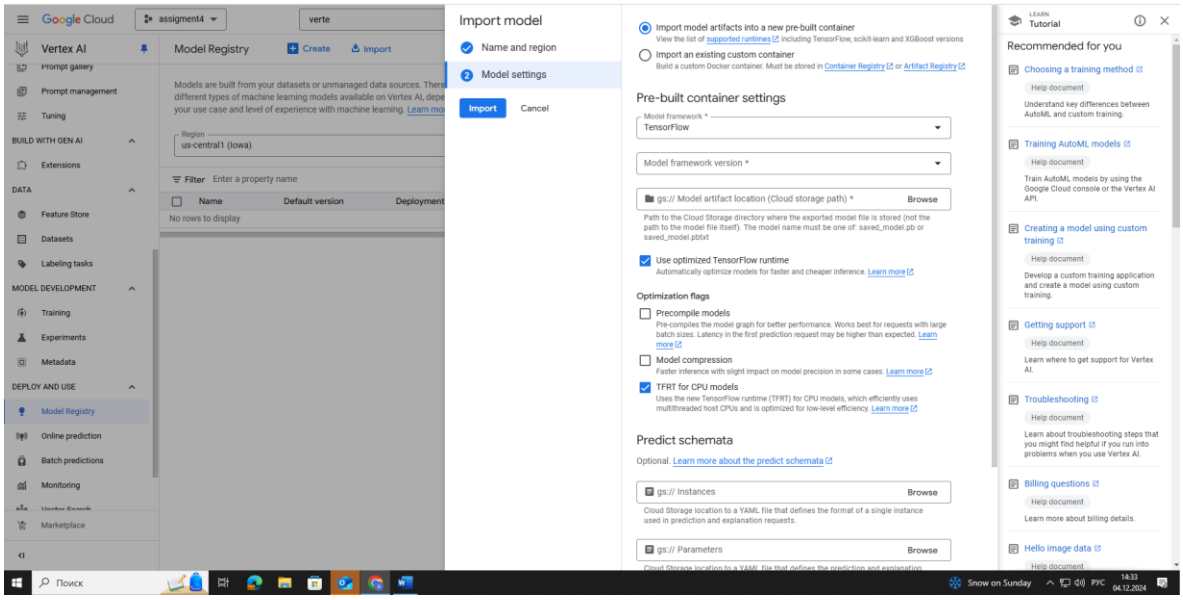
6. **Model Deployment:**
    - Deploy the trained model using AI Platform's serving capabilities.




```
aselek_m_s@cloudshell:~ (assigment4-443214)$ gsutil cp model.joblib gs://assigment4_bucket/model.joblib
CommandException: No URLs matched: model.joblib
aselek_m_s@cloudshell:~ (assigment4-443214)$
```

Picture-24

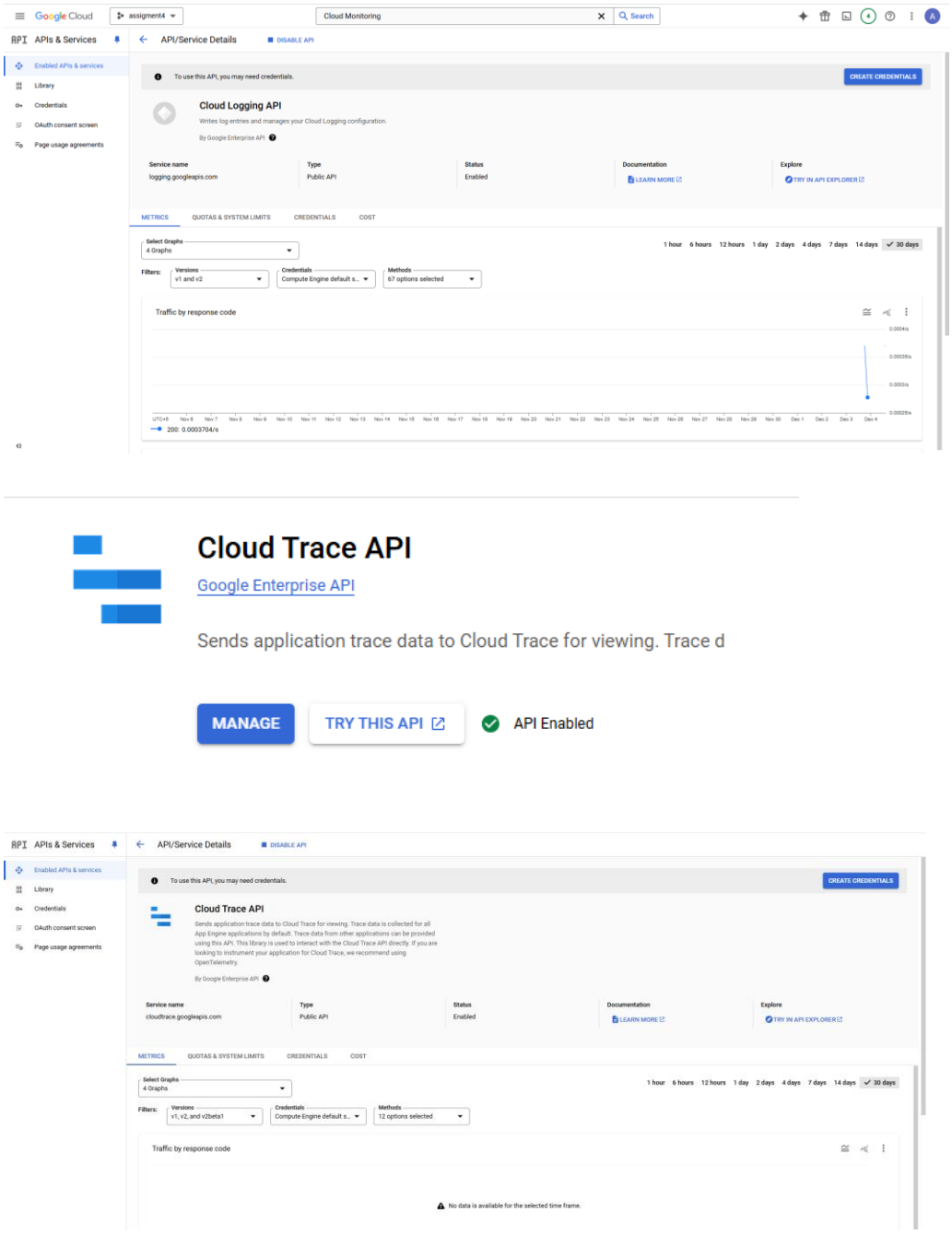
- Create an API endpoint for predictions.



Picture-25

Explanation: Time to share my creation with the world! I deployed my trained model using Vertex AI's serving features. I created a handy API endpoint so anyone could send data and get predictions back.

7. **Monitoring and Logging:**
    - Set up logging and monitoring for the deployed model to track usage and performance metrics.



Picture-26

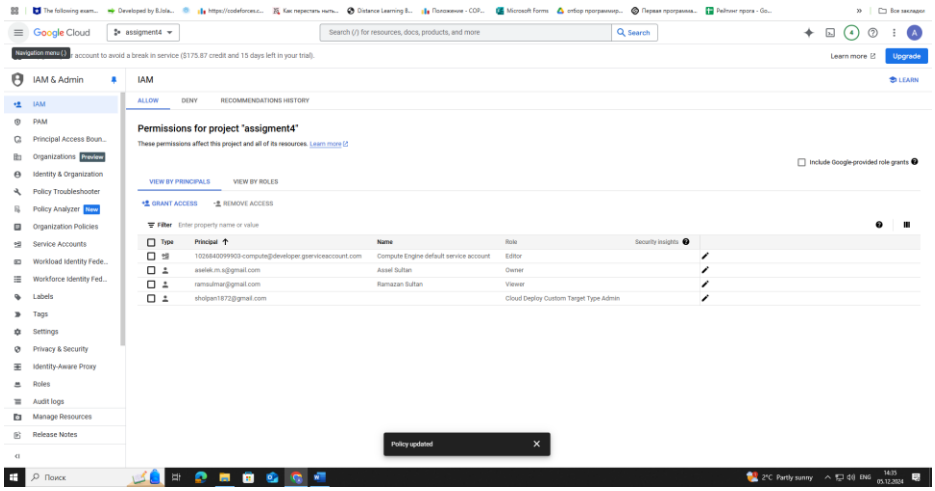
Explanation: I kept a close eye on my deployed model with logging and monitoring tools. It was important to me to track usage and performance – like making sure my little shop was running smoothly!

Exercise 2: Cloud Security and Compliance

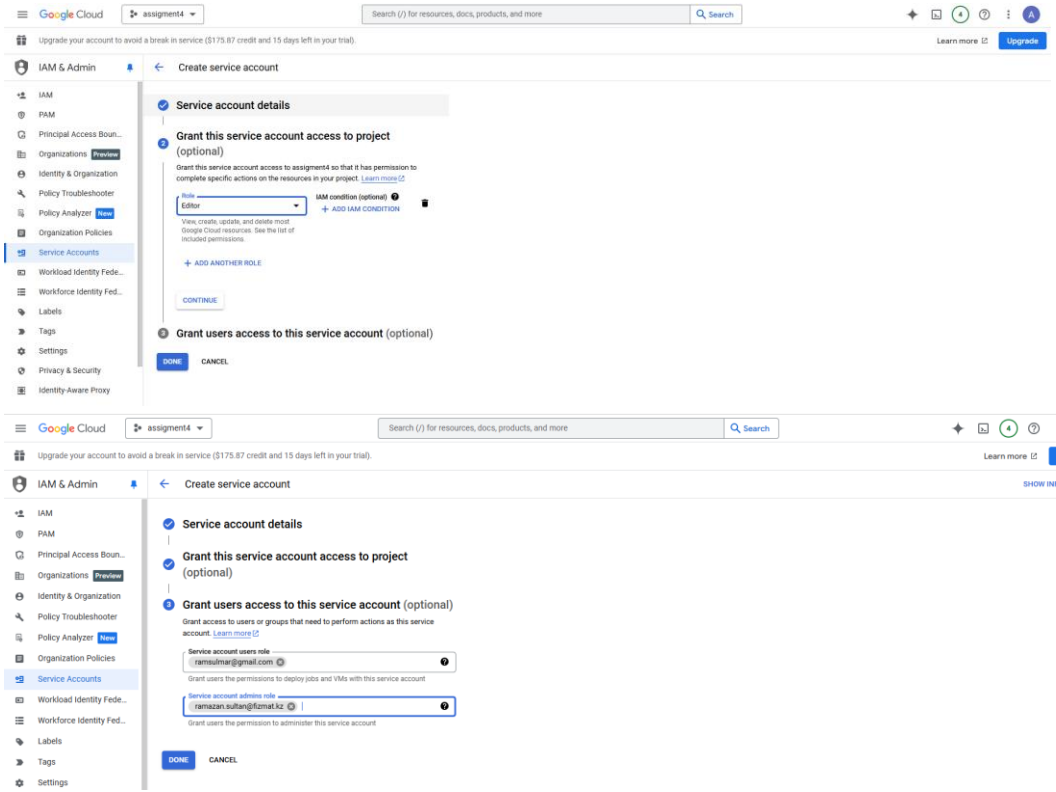
Objective: Implement security best practices and compliance measures for a Google Cloud project.

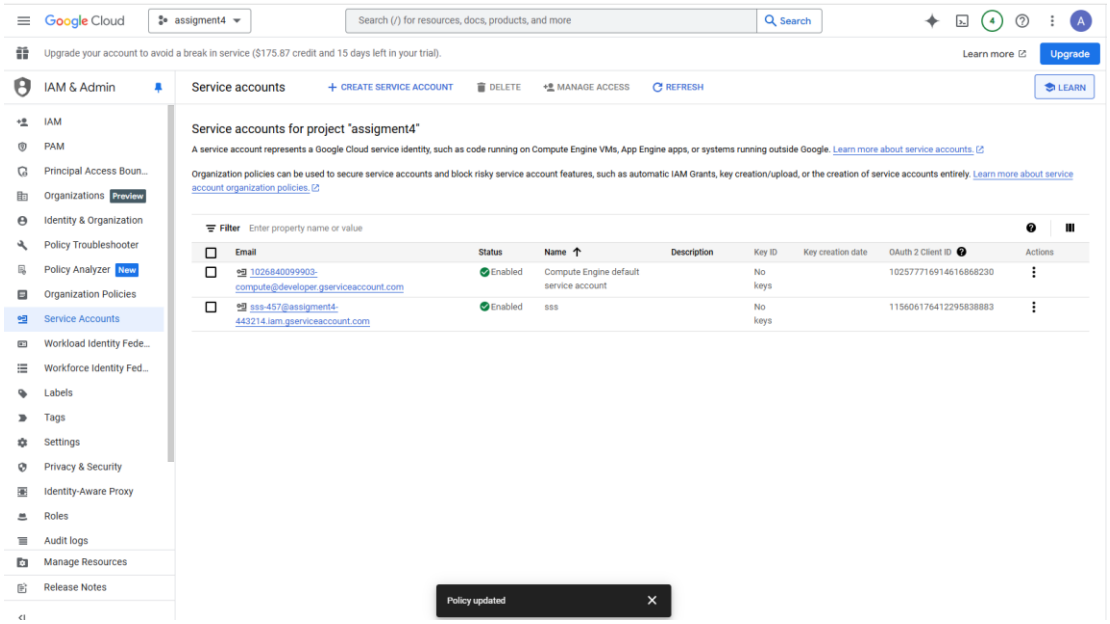
Tasks:

1. **Identity and Access Management (IAM):**
   - Configure IAM roles and permissions for different users in your project.



   - Implement the principle of least privilege for service accounts and users.

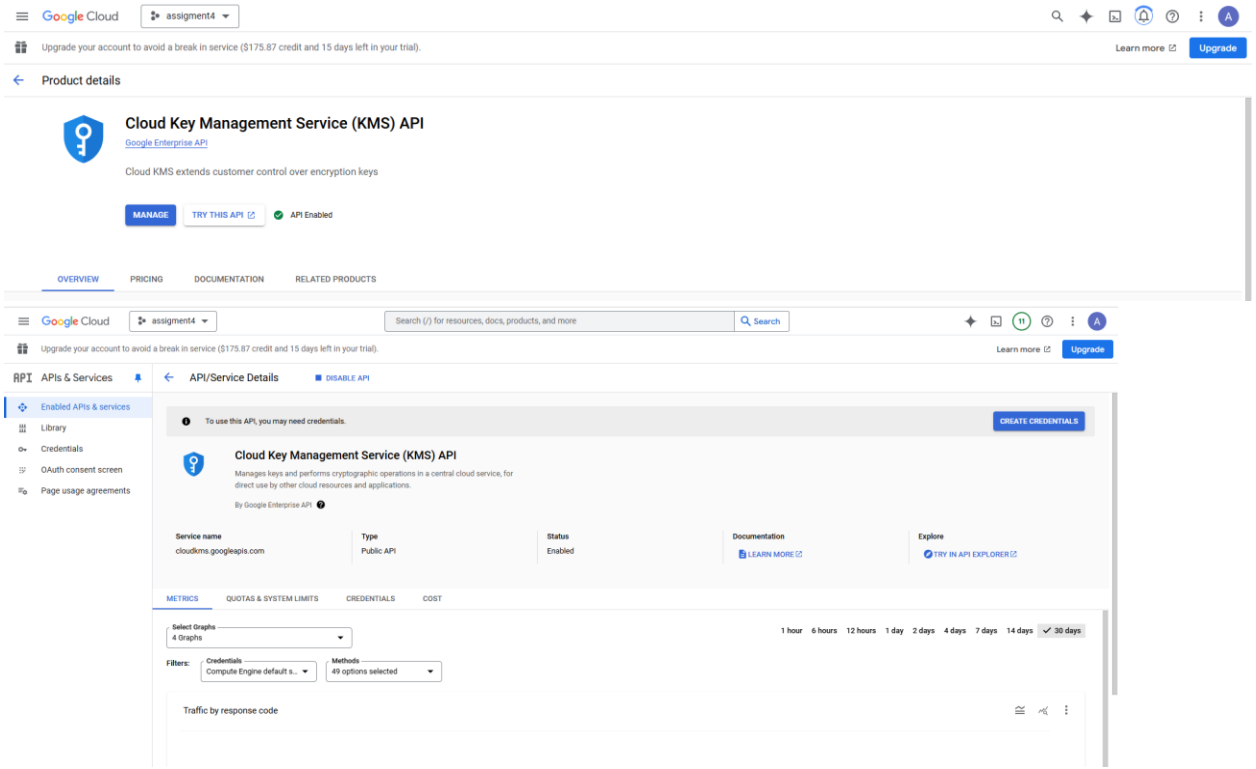




Picture-27

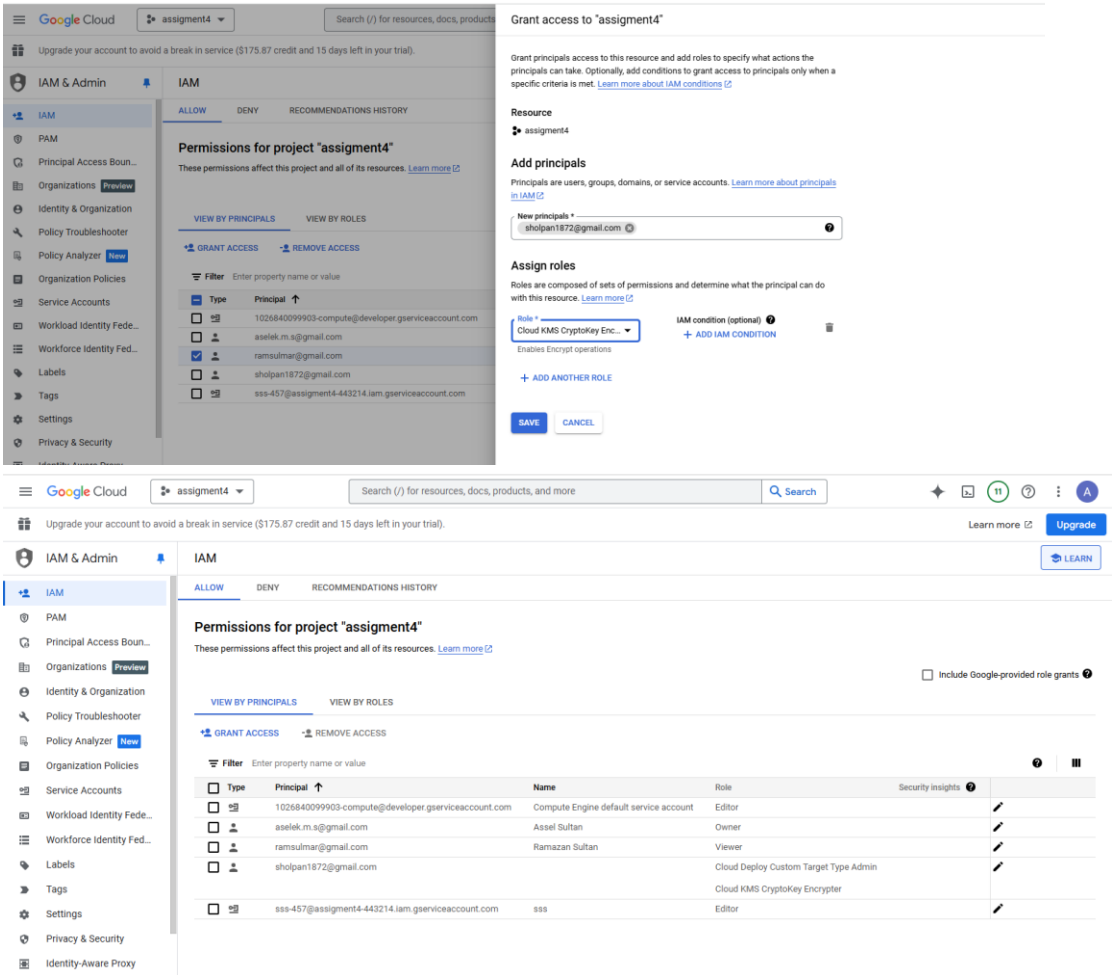
Explanation: I played security guard and carefully configured IAM roles and permissions. I followed the principle of least privilege, making sure everyone only had access to what they absolutely needed.

2. **Data Encryption:**
   - Ensure that data is encrypted at rest and in transit.



Picture-28

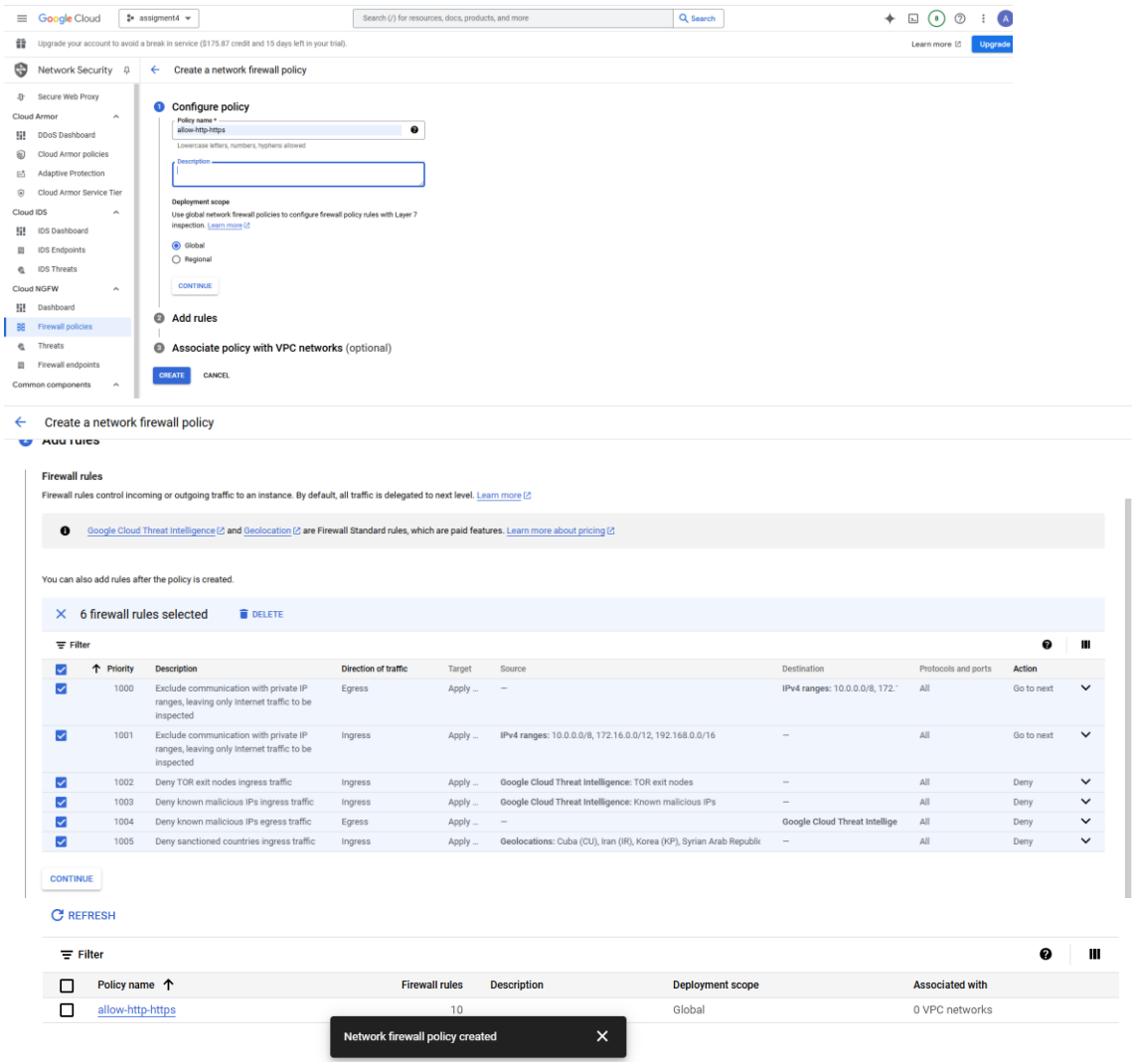
- Utilize Google Cloud KMS for managing encryption keys.



Picture-29

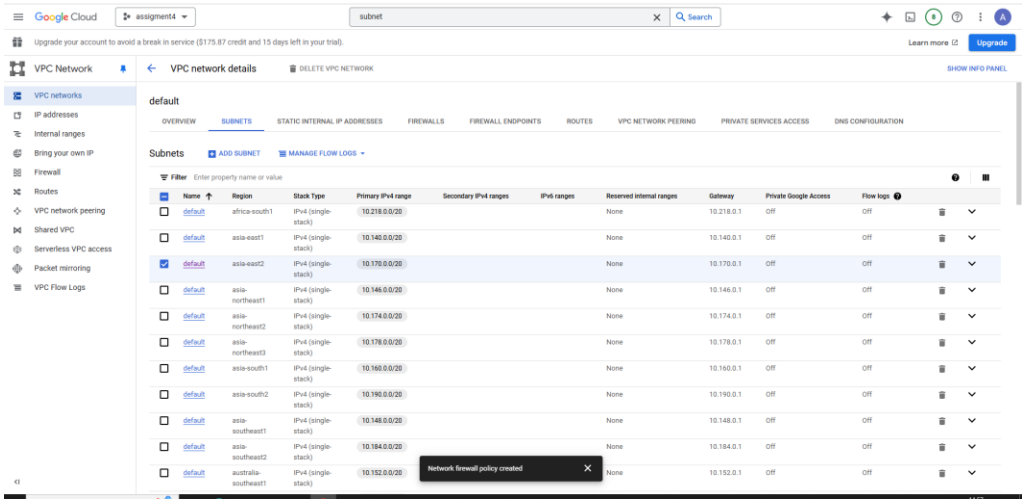
Explanation: Keeping data safe is a top priority! I made sure all my data was encrypted, both when it was stored (at rest) and when it was being transferred (in transit). I used Google Cloud KMS to manage my encryption keys like a pro.

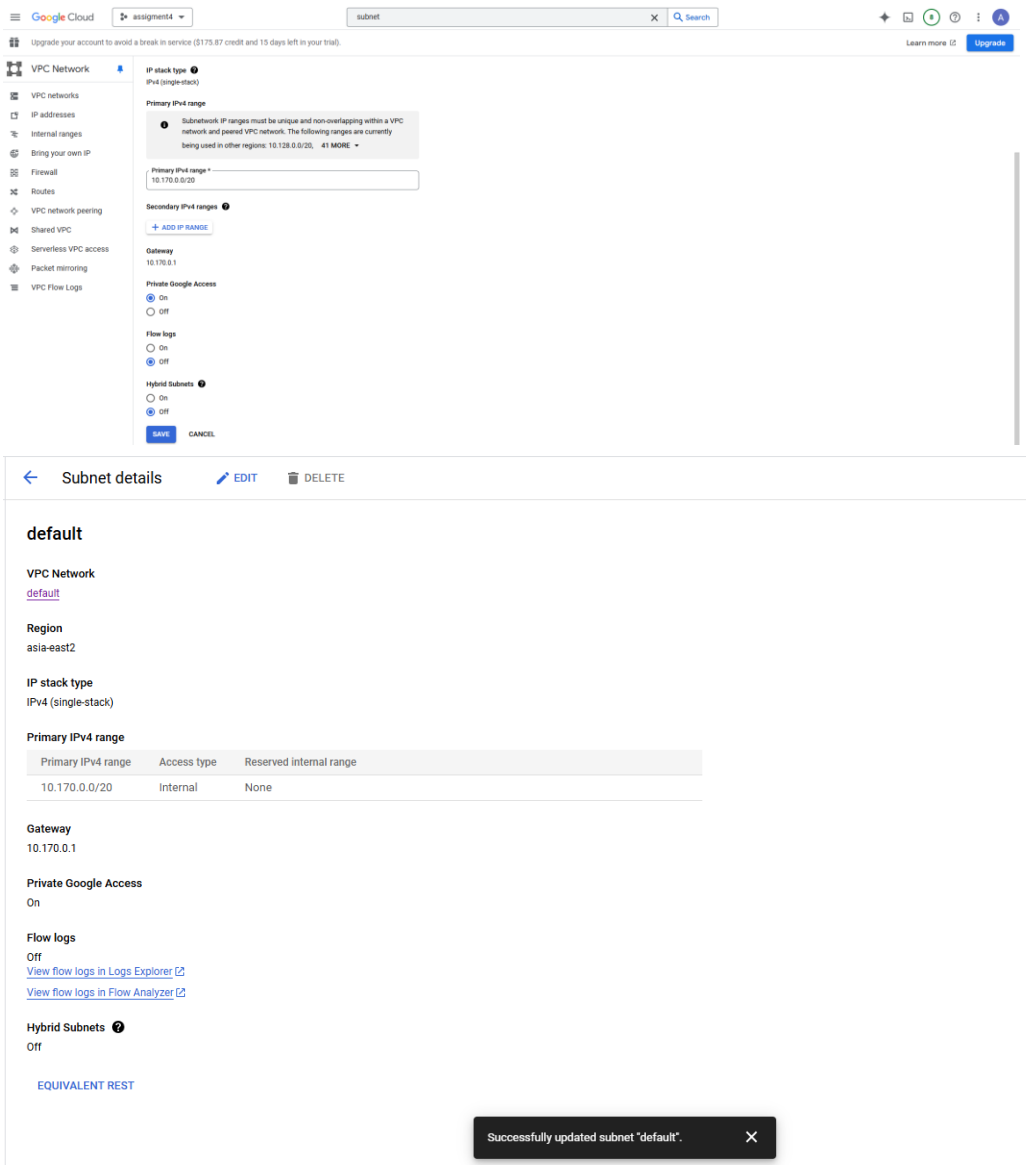
3. **Network Security:**
   - Set up Virtual Private Cloud (VPC) and configure firewall rules to restrict inbound and outbound traffic.



Picture-30

- Implement private Google access and ensure that sensitive data is not exposed to the public internet.

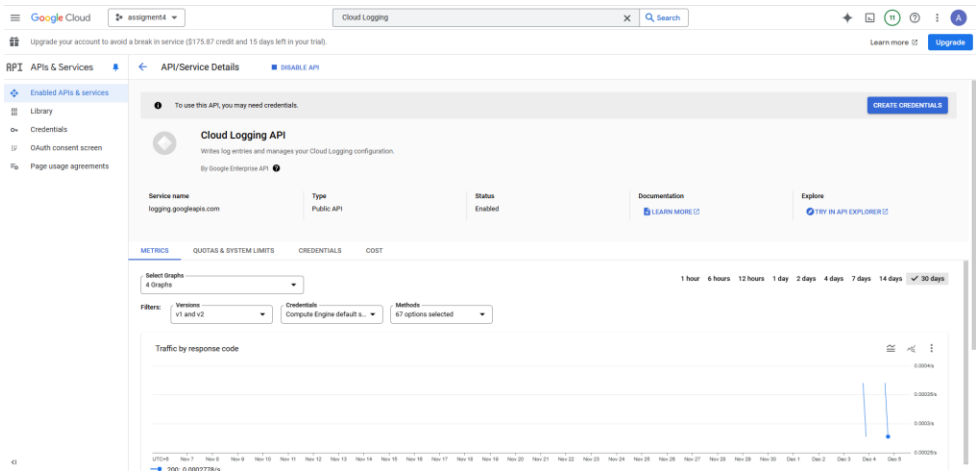




Picture-31

Explanation: I created a Virtual Private Cloud (VPC) and set up firewall rules to control the flow of traffic in and out. It was like building a fortress around my project! I used private Google access to keep sensitive data away from prying eyes on the public internet.

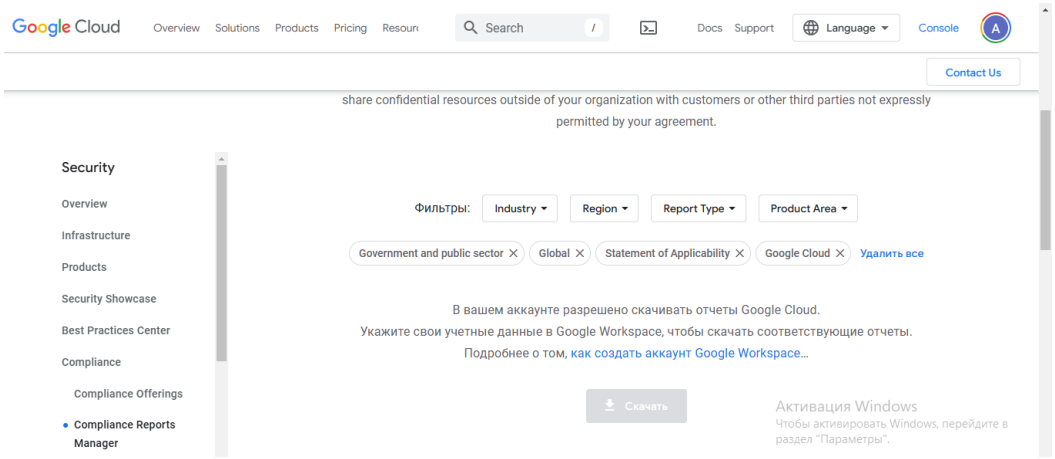
4. **Audit Logging:**
    - Enable Cloud Audit Logs to track access and changes to your resources.



Picture-32

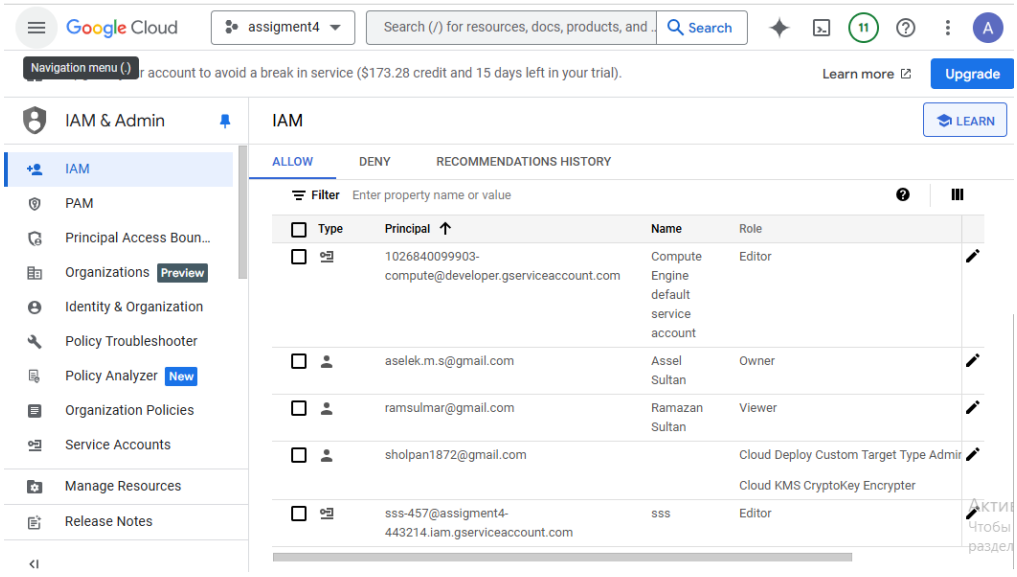
Explanation: I enabled Cloud Audit Logs to keep a record of every access and change to my resources. It was like having a detailed history book of my project's activity. I regularly reviewed the logs for anything unusual and set up alerts to warn me of suspicious events.

5. **Compliance Standards:**
   - Identify applicable compliance standards (e.g., GDPR, HIPAA) relevant to your project.

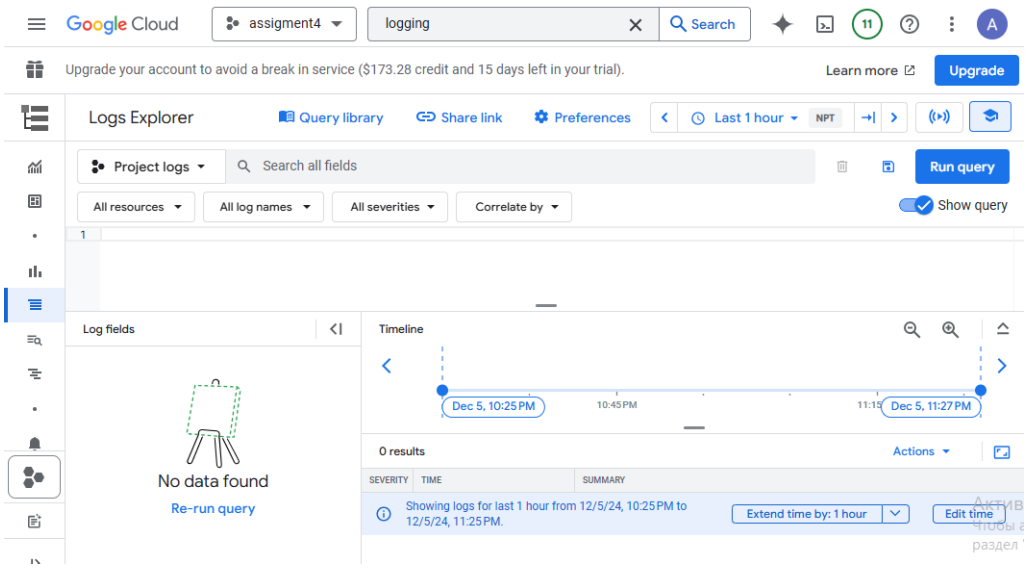


Picture-33

   - Implement measures to ensure compliance, such as data residency, access controls, and audit trails.



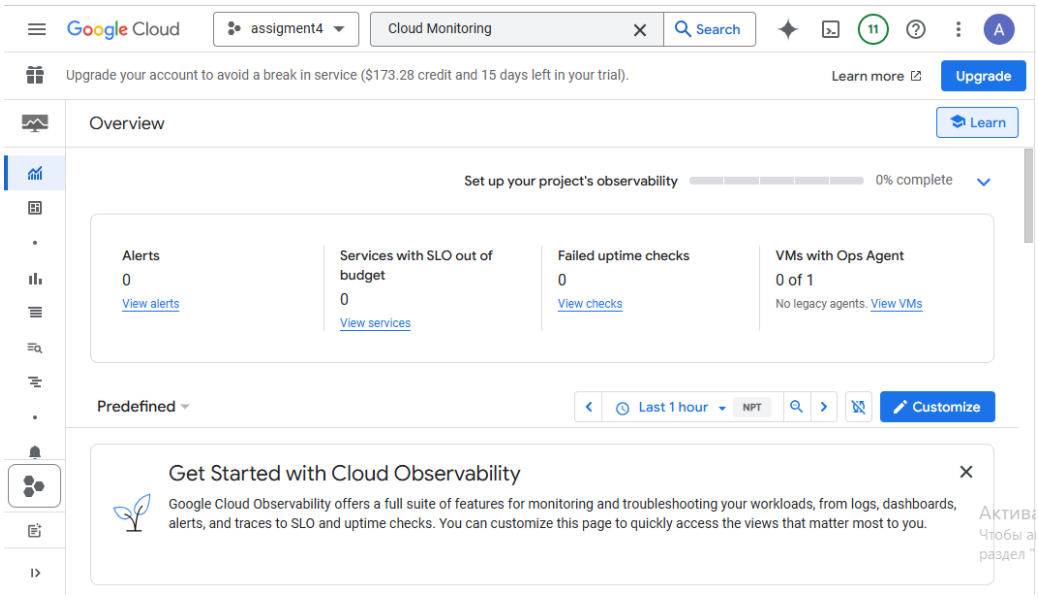
Picture-34



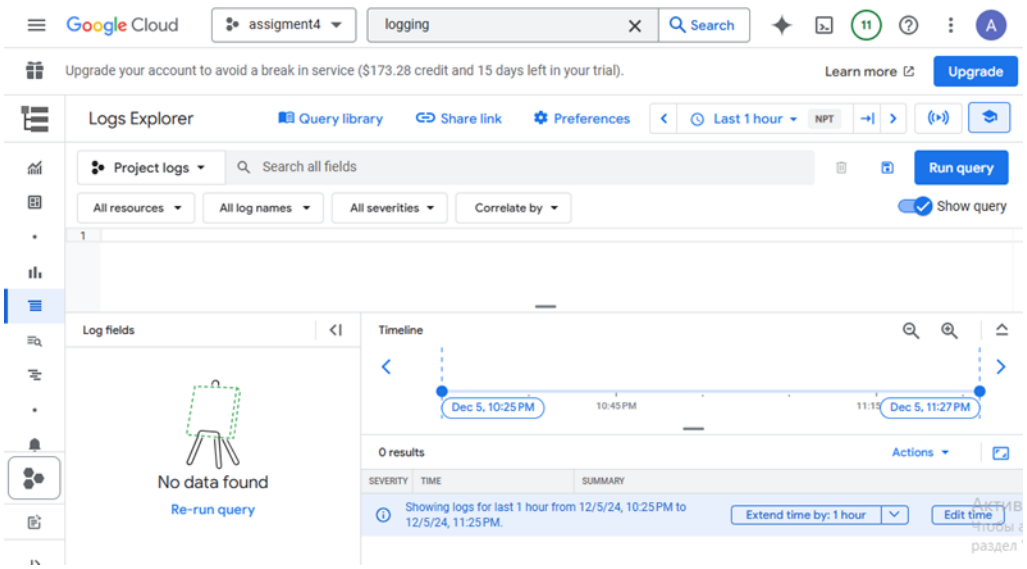
Picture-35

Explanation: As a project manager, I ensured compliance by identifying applicable standards like GDPR and HIPAA. I configured data residency by storing sensitive information in EU regions, enforced access controls using IAM with least-privilege principles, and enabled Cloud Audit Logs to track all data access and administrative changes. Regular audits and log reviews helped me maintain full visibility over our compliance posture.

6. **Incident Response Planning:**
    - Develop an incident response plan outlining the steps to take in case of a security breach.



Picture-36: Cloud Monitoring



Picture-37: Cloud Logging

Explanation: I developed a detailed incident response plan, outlining steps for detection, isolation, and recovery in case of a breach. To test its effectiveness, I simulated an incident by intentionally misconfiguring a firewall rule, allowing public access to a dummy resource. The system detected the issue, triggered alerts, and I swiftly resolved it by updating the rule. This simulation reinforced our preparedness and improved the plan for future scenarios.

Conclusion

Throughout this project, I explored the implementation of Big Data and Machine Learning on Google Cloud, focusing on building efficient pipelines for data ingestion, processing, and model deployment. I also applied essential security practices, including IAM, data encryption, and network security, to ensure compliance and safeguard sensitive information. These findings reinforced the importance of integrating advanced technologies with strong security measures to create reliable and scalable cloud solutions.

References

1. Google cloud services documentation (2024) Available at: https://cloud.google.com/storage (Accessed: 16 october 2024).

2. Google cloud services documentation (2024) Available at: https://cloud.google.com/network-connectivity-center?gad_source=1&gclid=CjwKCAjwpbi4BhByEiwAMC8JnbVPOou_1Ox1_sxAGDKeAtx2XIhE_l6K89F28-j4XzMeuET2Abx-BxoCK38QAvD_BwE&gclsrc=aw.ds (Accessed: 16 october 2024).