KAZAKH KBTU BRITISH
TECHNICAL
UNIVERSITY

Big Data and Machine Learning, and Cloud Security and Compliance on Google Cloud

Author's Name: Kerimbay Kairuddin

Date of Submission: December 04, 2024

Organization Name: Kazakh British Technical University (KBTU)

Almaty 2024

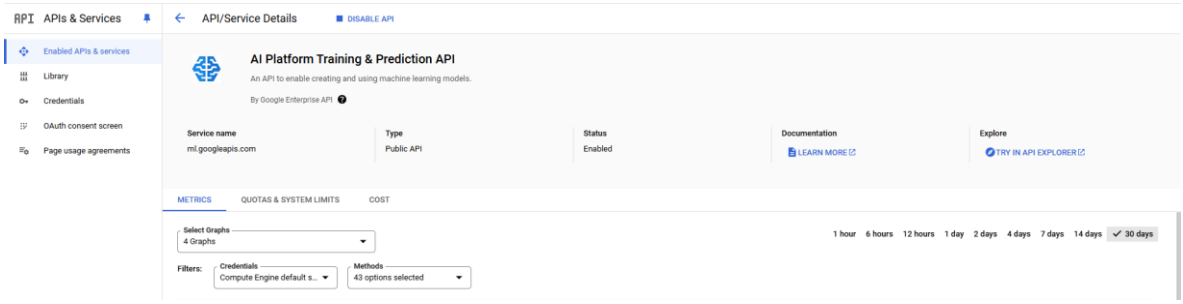
Exercise 1: Big Data and Machine Learning on Google Cloud

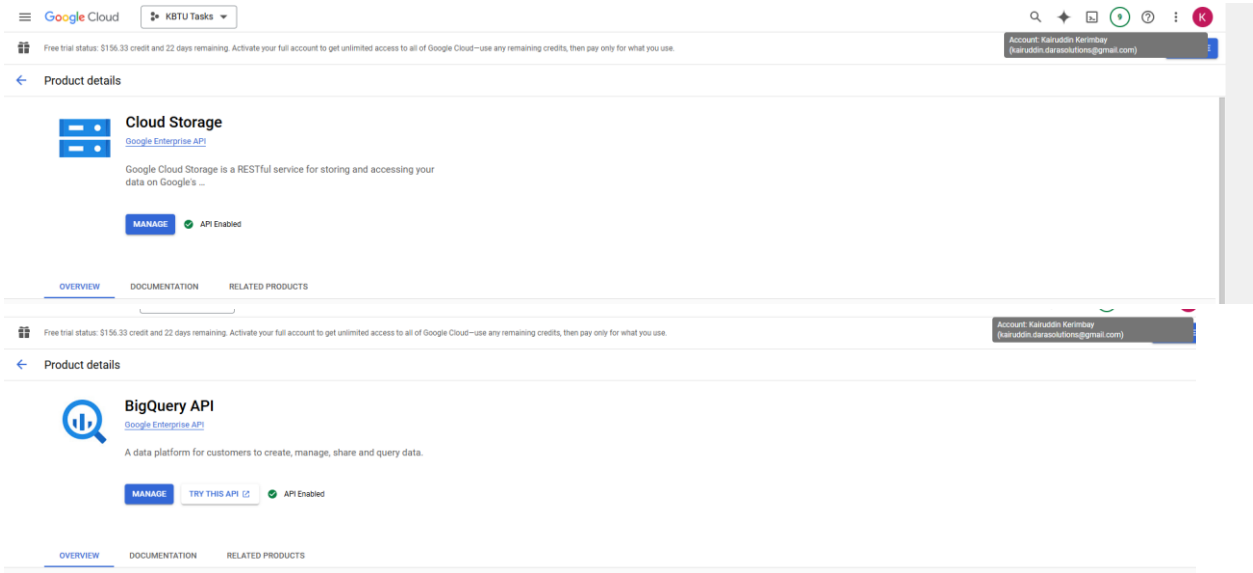
Tasks:

1. **Set Up a Google Cloud Project:**

   I utilized a previously established project from Assignment 1.

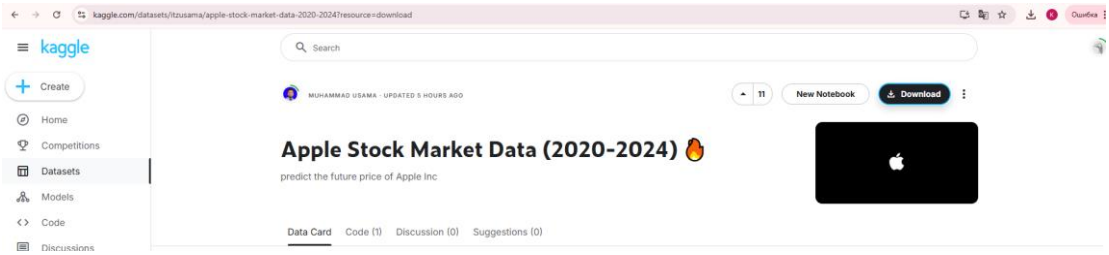
Within the 'APIs & Services' section, under 'Library,' I enabled the following APIs:

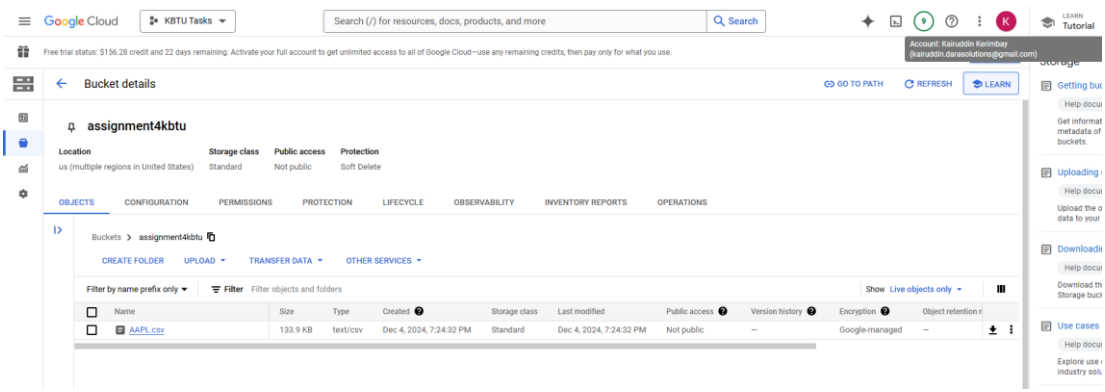




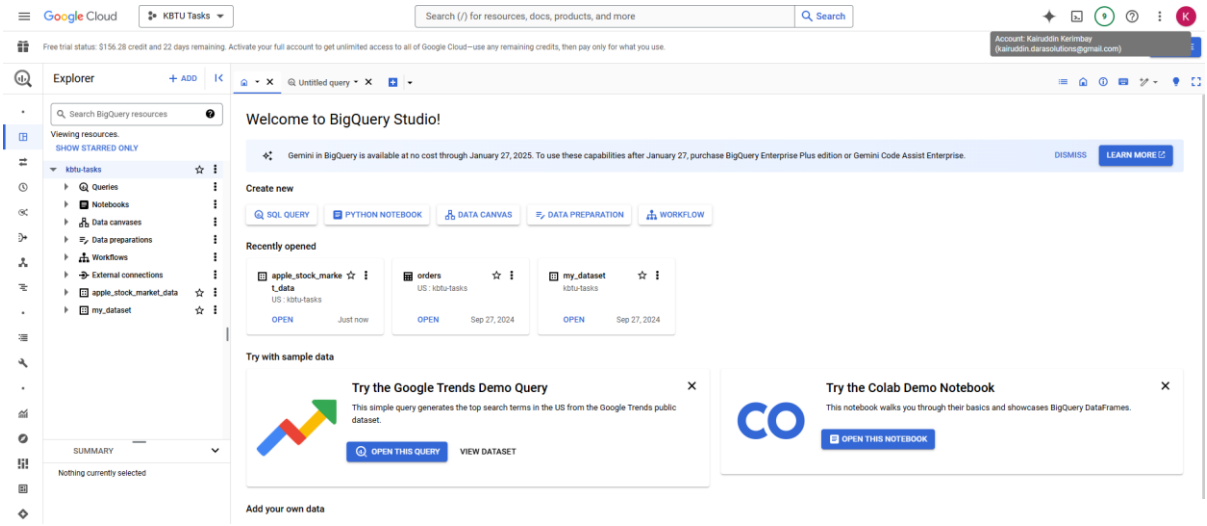
2. **Data Ingestion:**

   I sourced a dataset from Kaggle.com and subsequently uploaded it to a Google Cloud Storage bucket.

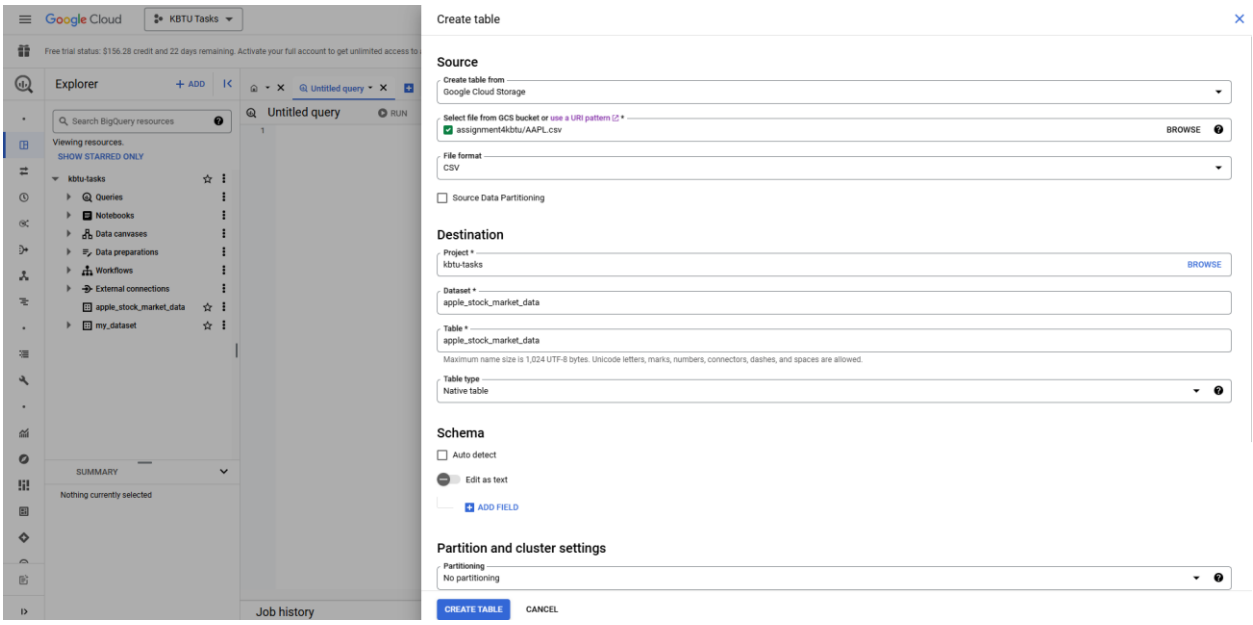


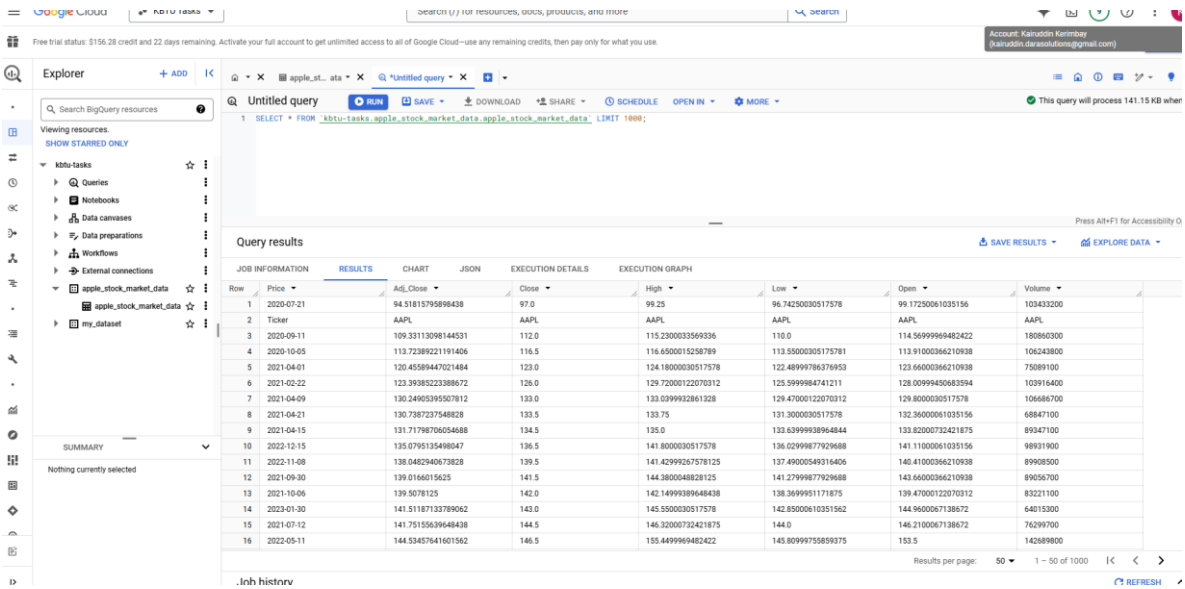


3. **Data Processing with BigQuery:**
In the BigQuery panel of the 'kbtu-tasks' project, I created a new dataset named 'apple_stock_market_data'.

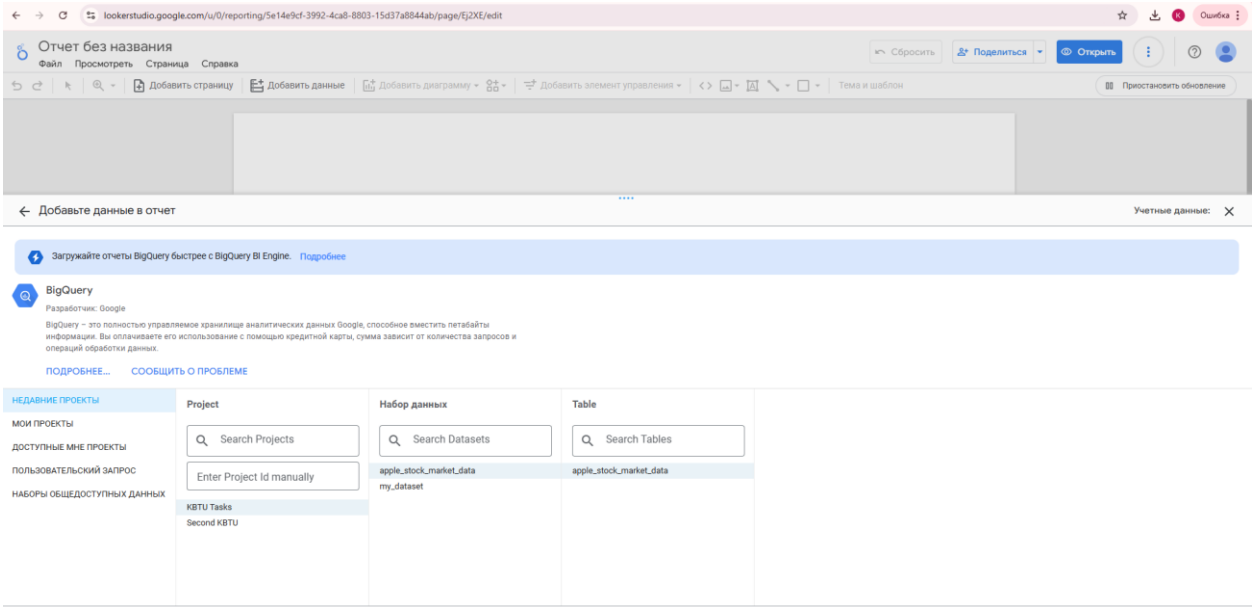


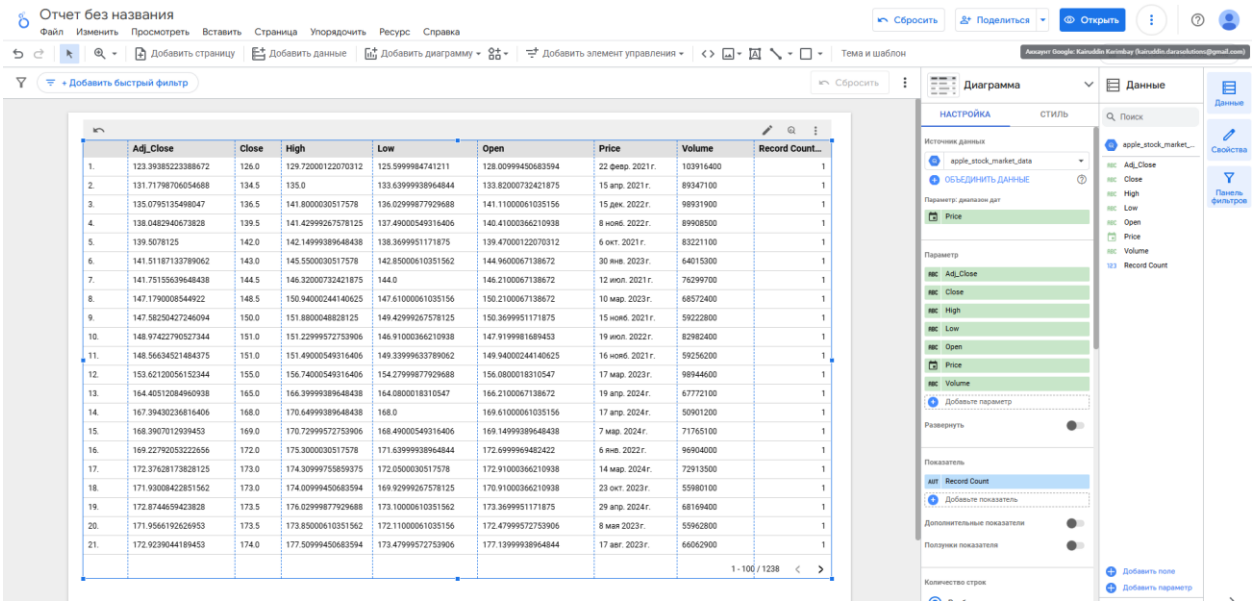
In the BigQuery interface within the 'kbtu-tasks' project, I configured a new table named 'apple_stock_market_data.' This table was set up to import data from a Google Cloud Storage CSV file, leveraging the comprehensive settings available for schema definition and table partitioning.



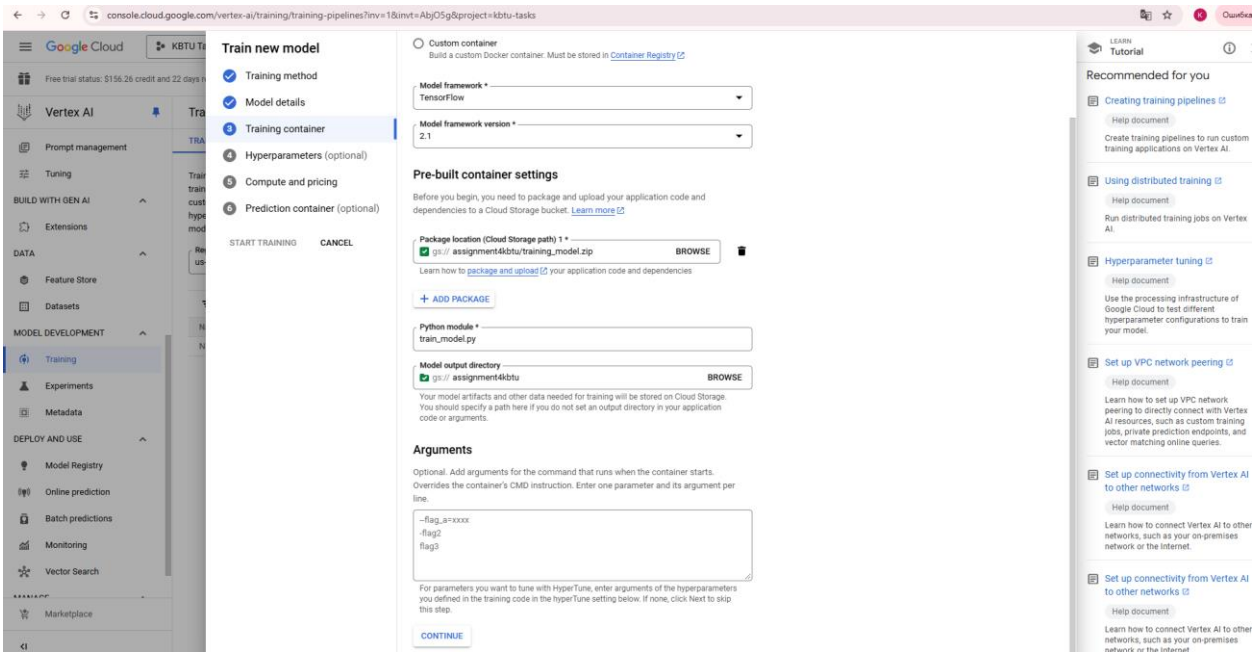


I generated a new report in Google Data Studio, selecting BigQuery as the data source." This version enhances the professionalism and clarity of the statement, making it suitable for a formal document.





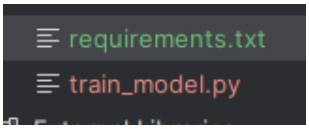
4. Machine Learning Model Training:



Here the python code:

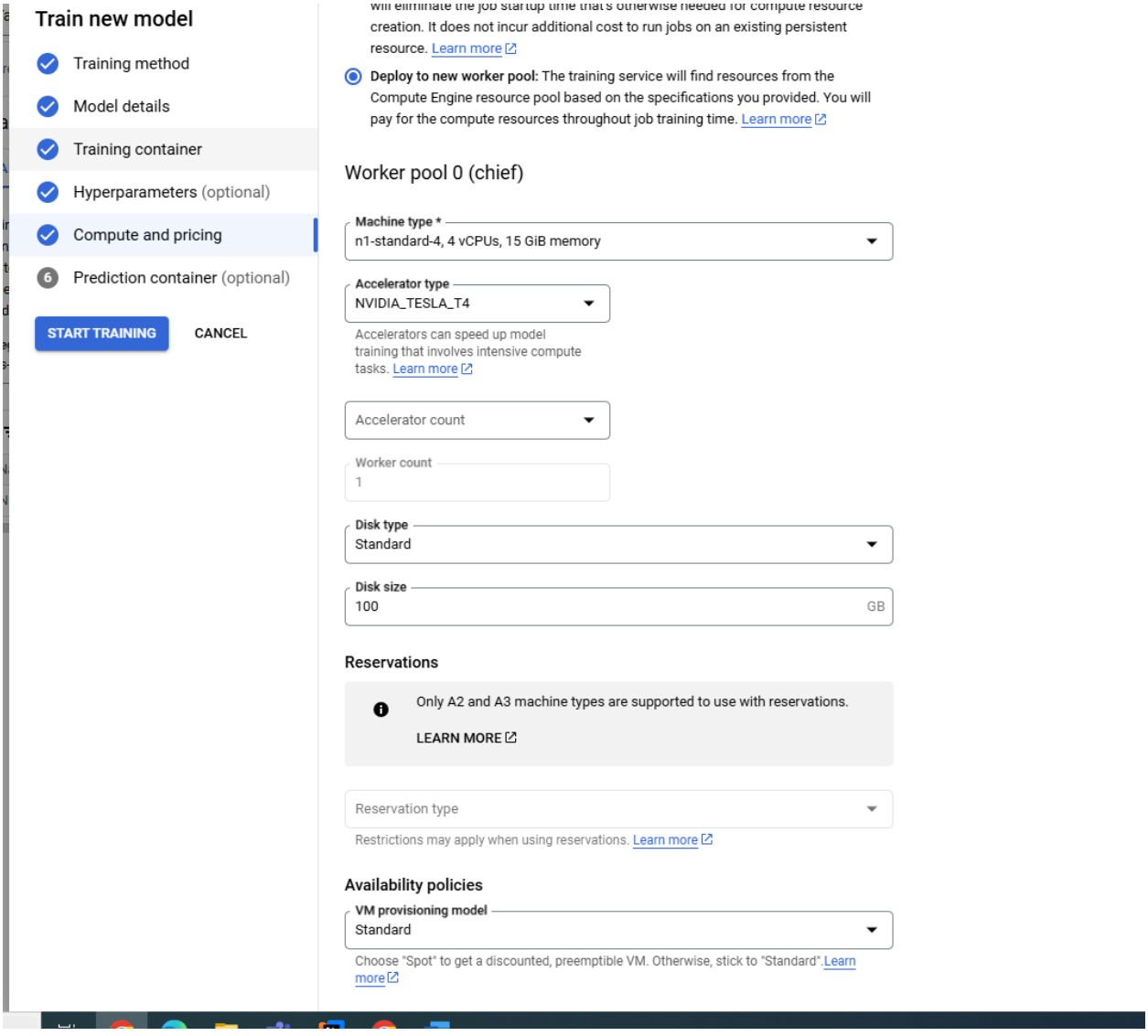
```
import tensorflow as tf
from sklearn.model_selection import train_test_split
import pandas as pd
df = pd.read_csv('gs://assignment4kbtu/AAPL.csv')
df.fillna(0, inplace=True)
X_train, X_test, y_train, y_test = train_test_split(df.drop('target', axis=1), df['target'], test_size=0.2)
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(128, activation='relu', input_shape=(X_train.shape[1],)),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=10, validation_data=(X_test, y_test))
model.save('gs://assignment4kbtu/model_output_directory')
```

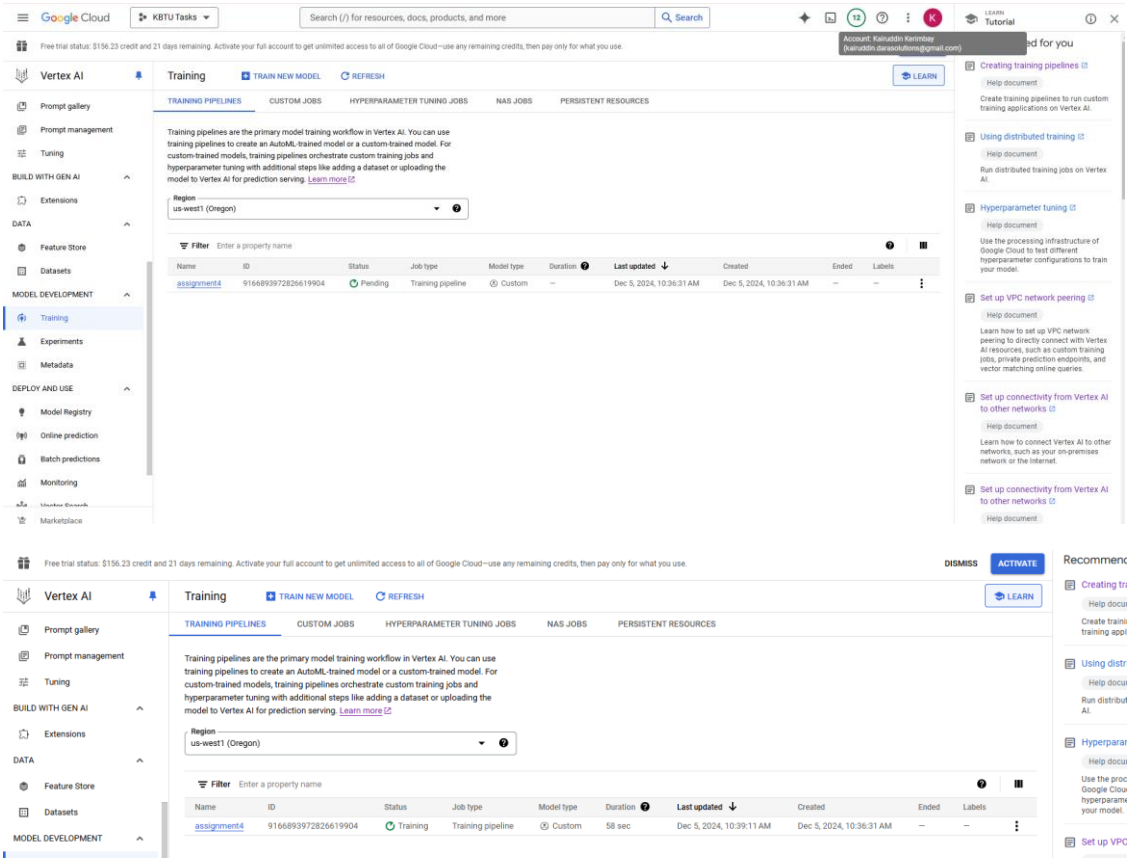
In than zip folder there these files



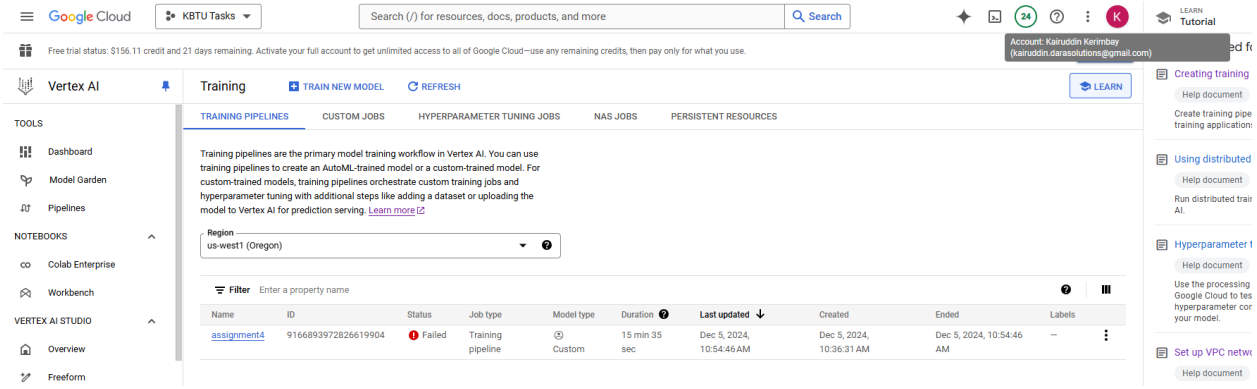
In requirements.txt there is this data, It is necessary to import dependencies:
tensorflow==2.6.0
pandas
scikit-learn

Here I chose basic machine, accelerator and disk types





This is the result. And I tried several times, but only got a glitch.

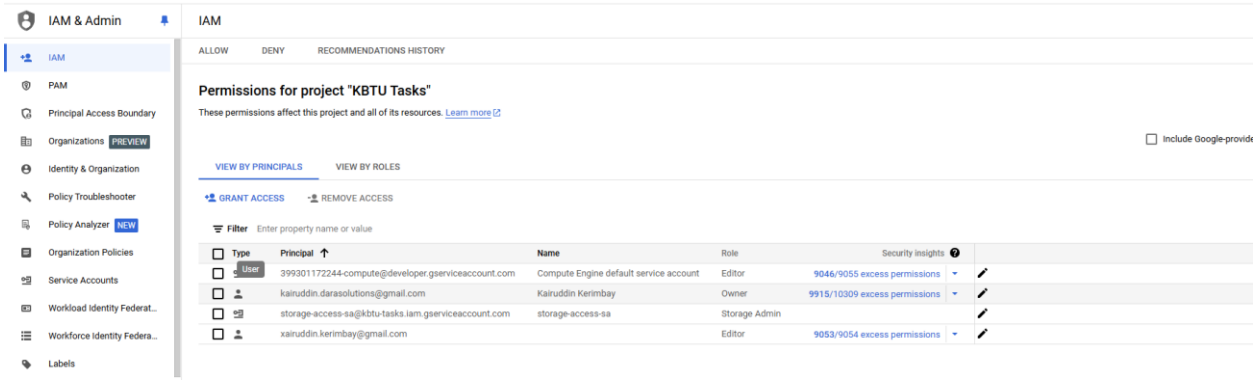


Exercise 2: Cloud Security and Compliance

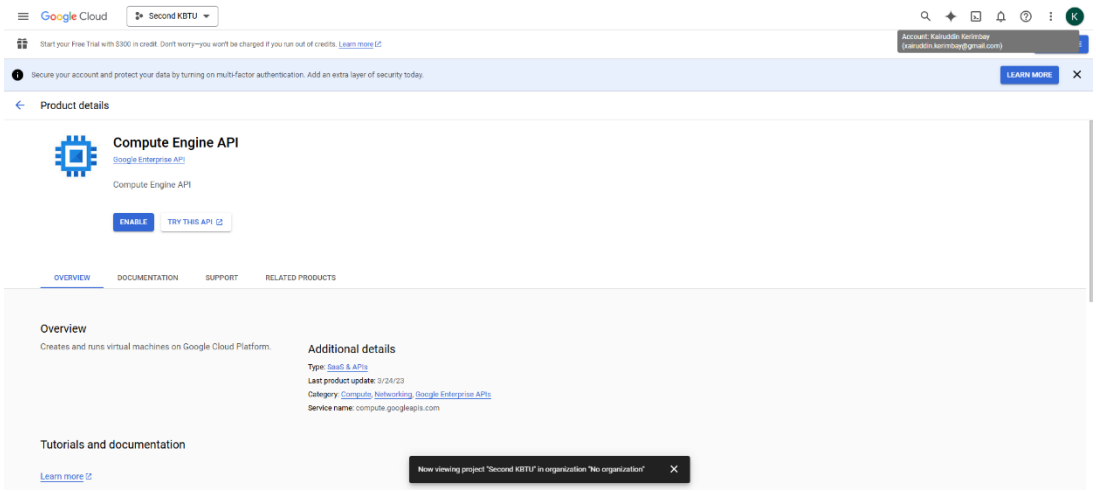
1. Identity and Access Management (IAM):

In the IAM & Admin section of the Google Cloud Console for the 'KBTU Tasks' project, I configured permissions for specific users. This involved assigning tailored roles to each user to control their access levels within the project. For example, one user was given the 'Editor' role, allowing them read-write access across most services, while another was assigned the 'Storage

Admin' role, focusing their permissions on cloud storage management.



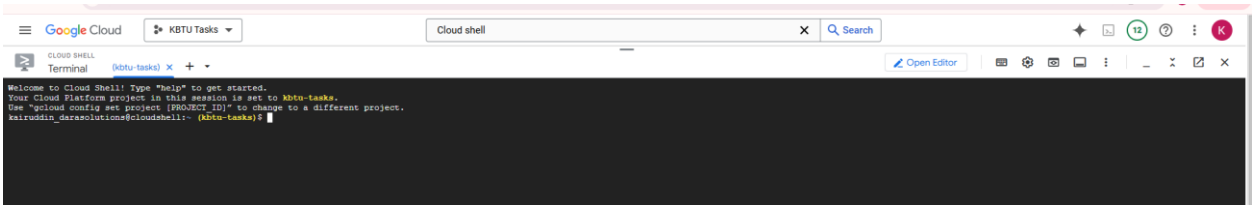
This screenshot depicts the user interface of the Google Cloud Platform as viewed from the account 'xairuddin.kerimbay'. The display shows the Compute Engine API overview page, indicating that this account has access to manage and view details related to Compute Engine services.



For more focused access management, there is a service account specifically dedicated to handling cloud storage operations: 'storage-access-sa@kbtu-tasks.iam.gserviceaccount.com'. This account is configured to limit its permissions strictly to cloud storage services, ensuring a higher level of security and compliance with the principle of least privilege.

2. **Data Encryption:**

To do these tasks I decided to use Google Cloud Shell



In the process of setting up data encryption, I utilized the Google Cloud Shell from within the 'kbtu-tasks' project environment to activate necessary services. As illustrated in the screenshot, I executed the command gcloud services enable cloudkms.googleapis.com to

enable the Google Cloud Key Management Service (KMS). This action was successfully completed as confirmed by the operation status message displayed in the terminal.

After enabling the necessary services, I proceeded to configure encryption keys using Google Cloud Key Management Service (KMS). Initially, I created a key ring named 'my-keyring' in the global location, as demonstrated by the command:

gcloud kms keyrings create "my-keyring" \ --location "global"

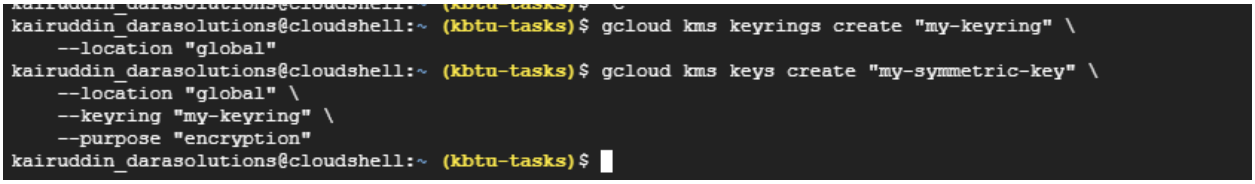
Following this, I created a symmetric encryption key named 'my-symmetric-key' within this key ring, designated specifically for encryption purposes. The command executed was:

**gcloud kms keys create "my-symmetric-key" **

**--location "global" **

**--keyring "my-keyring" **

--purpose "encryption"



These commands were successfully run in the Google Cloud Shell, setting up the foundational security infrastructure for data encryption.

Created a plaintext file named 'data.txt' containing the string 'my-contents' using the echo command in the Google Cloud Shell. This file was then encrypted using the Google Cloud Key Management Service. The encryption process was executed with the following command sequence:

Create a plaintext file:

$ echo "my-contents" > ./data.txt

Encrypt the file:

**$ gcloud kms encrypt **

**--location "global" **

**--keyring "my-keyring" **

**--key "my-symmetric-key" **

**--plaintext-file ./data.txt **

--ciphertext-file ./data.txt.enc

The output shows that the file 'data.txt.enc' was successfully created and lists its properties, confirming the encryption process was executed correctly. Subsequent verification commands display the

contents of both the plaintext and the encrypted file to demonstrate the successful encryption.

```
kairuddin_darasolutions@cloudshell:~ (kbtu-tasks)$ echo "my-contents" > ./data.txt
kairuddin_darasolutions@cloudshell:~ (kbtu-tasks)$ gcloud kms encrypt \
    --location "global" \
    --keyring "my-keyring" \
    --key "my-symmetric-key" \
    --plaintext-file ./data.txt \
    --ciphertext-file ./data.txt.enc
kairuddin_darasolutions@cloudshell:~ (kbtu-tasks)$ ls -l ./data.txt.enc
-rw-rw-r-- 1 kairuddin_darasolutions kairuddin_darasolutions 93 Dec  4 18:35 ./data.txt.enc
kairuddin_darasolutions@cloudshell:~ (kbtu-tasks)$ cat ./data.txt.enc

[?2004hkairuddin_darasolutions@cloudshell:~ (kbtu-tasks)$ sudo cat ./data.txt
my-contents
```

Following the encryption of the data, I proceeded to decrypt the file to verify the integrity and success of the encryption process. This was achieved using the Google Cloud Key Management Service with the command shown below

gcloud kms decrypt \

--location "global" \

--keyring "my-keyring" \

--key "my-symmetric-key" \

--plaintext-file - \

--ciphertext-file ./data.txt.enc

```
kairuddin_darasolutions@cloudshell:~ (kbtu-tasks)$ gcloud kms decrypt \
    --location "global" \
    --keyring "my-keyring" \
    --key "my-symmetric-key" \
    --plaintext-file - \
    --ciphertext-file ./data.txt.enc
my-contents
```

The command specifies the encrypted file (data.txt.enc) and outputs the decrypted contents directly to the console, confirming that the original data ('my-contents') was successfully restored. This demonstrates the effective use of encryption and decryption processes managed through Google Cloud KMS.

3. Network Security:

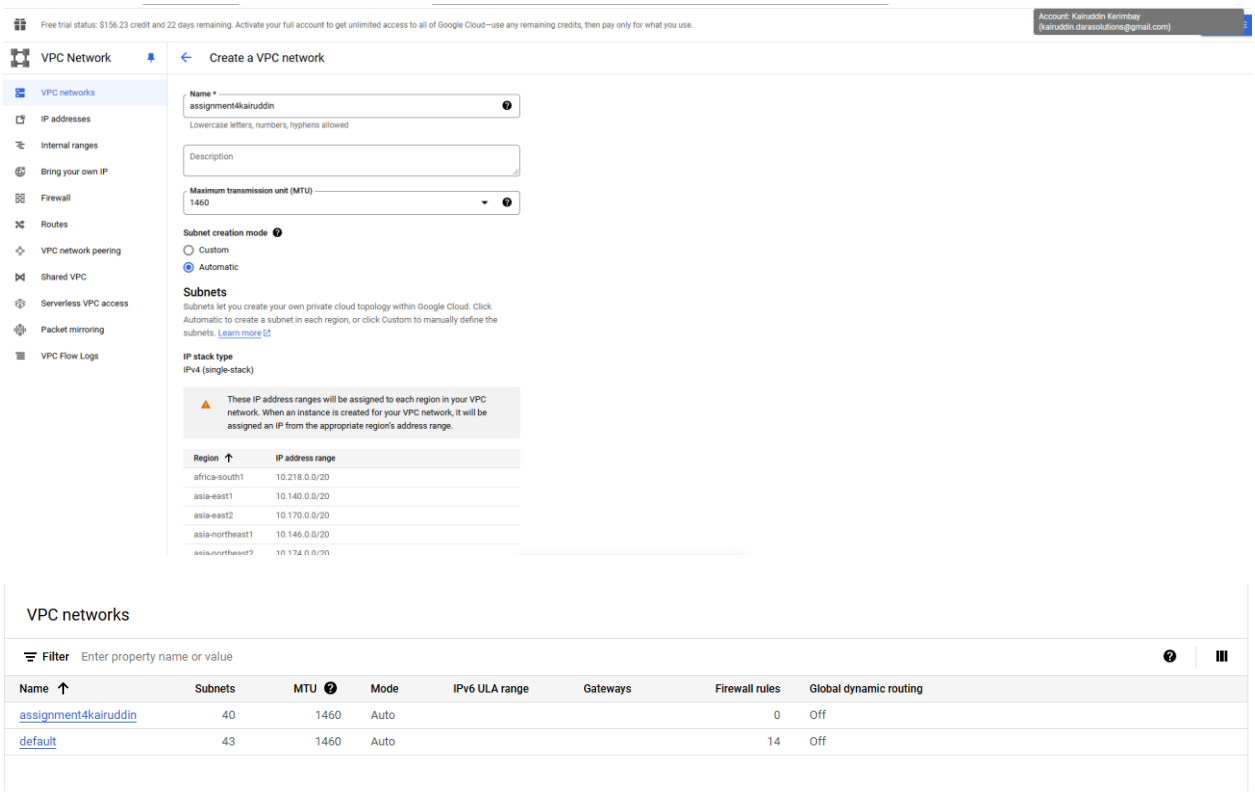
In the setup of network security for the project, I configured a new Virtual Private Cloud (VPC) network named 'assignment4kairuddin' in Google Cloud. This configuration included setting up various subnets to ensure a structured and secure network environment across multiple regions, as shown in the screenshot.

The VPC was set to custom mode, allowing for the explicit definition of subnets rather than using automatic subnet creation. This approach gives precise control over the IP ranges and locations, enhancing the security and efficiency of network resource allocation.

Key details of the setup included:

- Name of the VPC: assignment4kairuddin
- Subnet creation mode: Custom, allowing for tailored subnet management.
- IP ranges: Configured to align with regional requirements, ensuring optimal performance and compliance with data residency regulations.
- MTU settings: Set at 1460, the default for Google Cloud, ensuring effective data transmission.

This configuration underscores the project's commitment to robust network security practices by segmenting network traffic and applying region-specific settings to meet operational and compliance needs.

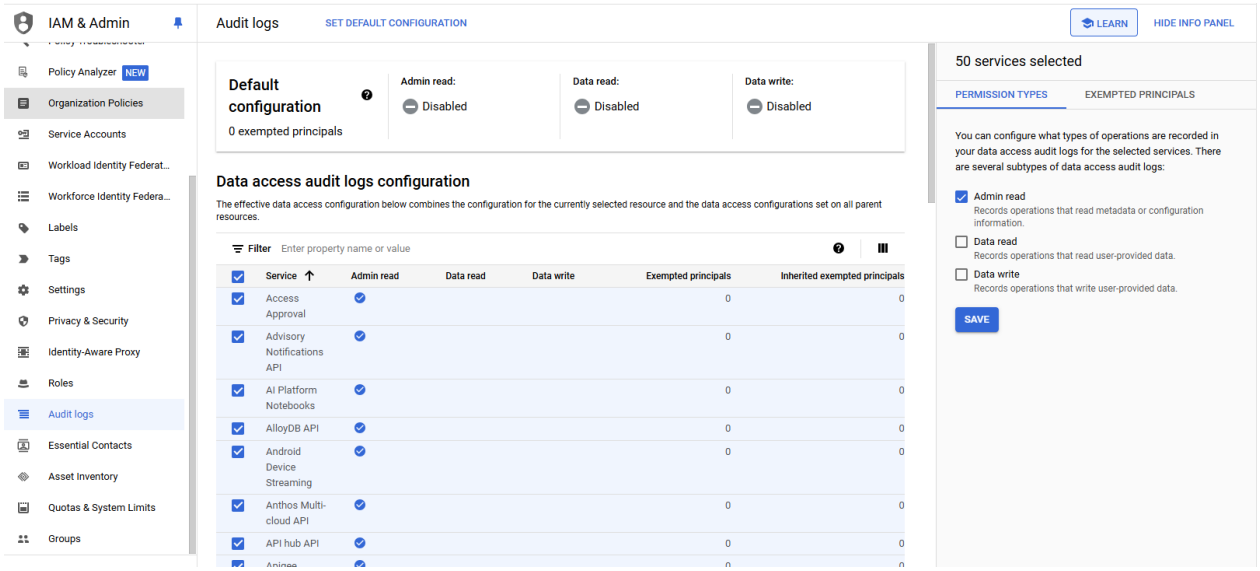


4. Audit Logging:

In this segment of the project, I focused on setting up and managing Audit Logging to enhance the security and compliance measures of our Google Cloud environment. The screenshot illustrates two main aspects:

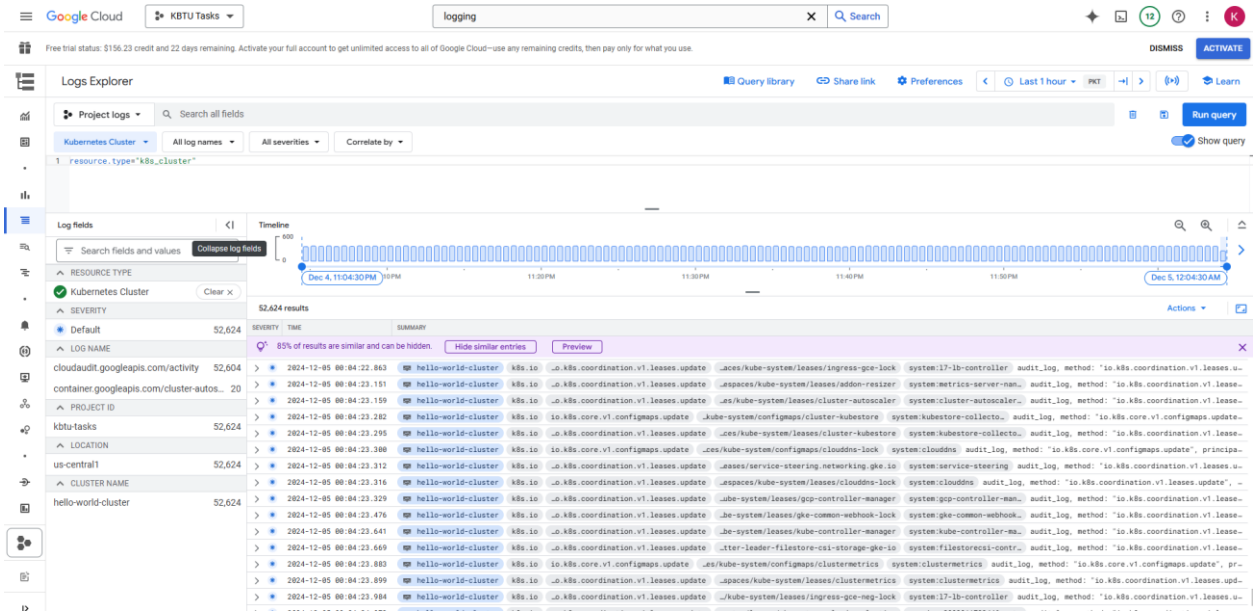
Audit Logs Configuration:

Log Management and Review:



The IAM & Admin > Audit Logs panel displays the default configuration settings for various services. I tailored the logging settings to ensure comprehensive coverage across administrative read, data read, and data write activities, which are essential for maintaining rigorous oversight.

Log Management and Review:



In the Logs Explorer section, you can see the activity logs filtered for a particular project. This includes logs related to Kubernetes Engine and other critical components, providing a granular view of operations within the platform.

These configurations and the proactive review of audit logs are crucial for detecting potential security incidents early and providing audit trails necessary for compliance with **industry standards.**

Executive Summary

In this assignment, I embarked on a comprehensive journey to understand and implement key technologies within Google Cloud, focusing on Big Data, Machine Learning, and robust security measures. Here's a concise overview of what I have learned and achieved:

Big Data and Machine Learning: I developed a hands-on understanding of how to construct and manage a data processing pipeline using Google Cloud's BigQuery and Cloud Storage. This experience allowed me to appreciate the complexities of managing large datasets and the power of cloud computing in processing and analyzing data at scale. I also engaged in building and training machine learning models, which enhanced my skills in applying theoretical knowledge to practical, real-world data problems.

Security Practices: I learned the critical importance of securing data and applications in the cloud environment. Through the implementation of IAM roles, data encryption, and network security measures such as VPC configurations and firewall rules, I gained practical insights into the layered security approach necessary for protecting sensitive information and ensuring compliance with data protection standards.

Audit and Compliance: Setting up and managing audit logs helped me understand the mechanisms for monitoring and recording activities within cloud environments, which is crucial for compliance and security governance.

This assignment not only enhanced my technical skills but also deepened my understanding of the operational and security challenges in cloud computing, preparing me for future roles in technology and data management.

References

- Google Cloud Documentation. (n.d.). *Creating a training pipeline with Vertex AI*. Retrieved from https://cloud.google.com/vertex-ai/docs/training/create-training-pipeline
- Kaggle. (n.d.). *Models*. Retrieved from https://www.kaggle.com/models
- HsjTIrCiAmQc. (n.d.). *YouTube*. Retrieved from https://www.youtube.com/watch?v=HsjTIrCiAmQc
- Looker Studio. (n.d.). *Document editing page*. Retrieved from https://lookerstudio.google.com/u/0/reporting/5e14e9cf-3992-4ca8-8803-15d37a38844b/page/Ej2XE/edit
- Google Cloud Documentation. (n.d.). *Audit logs*. Retrieved from https://cloud.google.com/logging/docs/audit