Assignment 1, Cloud Computing

Exercise 1: Understanding Cloud Computing Models

1. **Objective**: Explore different cloud computing models and understand their key differences.
2. **Steps:**

Step 1:

1. Infrastructure as a Service (IaaS)

IaaS provides users with virtualized computing resources over the internet. Instead of investing in physical hardware, businesses can rent servers, storage, and networking from a cloud provider. We can describe its key features as: full control over their operating systems and applications, users can scale their resources and lastly it gives the opportunity to reduce the need for physical infrastructure. There are several use cases, basically it's web hosting, backup and storage, development and testing.

Examples:

Amazon Web Services (AWS) EC2, Microsoft Azure, Google Cloud Compute Engine.

2. Platform as a Service (PaaS)

PaaS provides a platform that allows developers to build, run, and manage applications without dealing with the underlying infrastructure. We can describe its key features as: development tools, managed services and integration support. There are several use cases, basically it's application development, API development and collaboration.

Examples:

Google App Engine, Microsoft Azure App Service, Heroku.

3. Software as a Service (SaaS)

SaaS delivers software applications over the internet on a subscription basis, meaning users don't need to install or maintain the software locally. We can describe its key features as: accessibility, automatic updates and scalability. There are several use cases, basically it's business applications and collaboration tools.

Examples:

Dropbox, Google Workspace, QuickBooks Online.

Step 2

Table №1 - Comparing table of cloud computing models

Feature	**IaaS**	**PaaS**	**SaaS**
Control	*High control* over infrastructure and applications	*Moderate control*; focus on app development	*Low control*; users interact with pre-built

			software
Flexibility	*Highly flexible*; can customize OS and applications	*Flexible* for app development; limited to the platform's tools	*Limited flexibility;* customization is usually minimal
Use Cases	Hosting websites, data storage, backup solutions, and development/testing environments	Developing and deploying web applications, API management, and collaborative projects	Email services, CRM systems, project management tools, and collaboration applications

Step 3

Here are examples of services offered by Google Cloud Platform (GCP) under each cloud service model:

1. Infrastructure as a Service (IaaS)

- ***Google Compute Engine***: Provides virtual machines (VMs) for running applications.
- ***Google Cloud Storage***: Scalable object storage for unstructured data.
- ***Google Cloud Virtual Private Cloud (VPC)***: Networking capabilities to manage resources securely.

2. Platform as a Service (PaaS)

- ***Google App Engine***: A platform for building and hosting web applications in various languages.
- ***Google Cloud Functions****:* A serverless execution environment for running code in response to events.
- ***Google Cloud Run***: A managed compute platform for deploying containerized applications.

3. Software as a Service (SaaS)

- ***Google Workspace****:* A suite of productivity and collaboration tools, including Gmail, Google Docs, and Google Drive.
- ***Google BigQuery****:* A fully managed data warehouse for analytics and business intelligence.
- ***Google Cloud Identity****:* Identity management for users and devices, providing access to Google services and applications.

3. **Questions**:

    - What are the main differences between IaaS, PaaS, and SaaS?

      Answer: The main difference between these cloud computing models is that IaaS offers the most control and flexibility, PaaS balances between development and management, while SaaS provides ready-to-use software with minimal control.

- Which GCP services fall under each of these models?

  Answer:
  IaaS: *Google Compute Engine*, *Google Cloud Storage*, *Google Cloud Virtual Private Cloud (VPC)*
  PaaS: *Google App Engine*, *Google Cloud Functions, Google Cloud Run*
  SaaS: *Google Workspace, Google BigQuery, Google Cloud Identity*

- Provide a real-world example where each cloud service model might be the most appropriate choice.

  Answer:

  Example (IaaS): A startup needs to host a scalable web application without investing in physical servers. They use Google Compute Engine to deploy VMs that can grow with user demand.

  Example (PaaS): A software development team wants to build and deploy a mobile app quickly. They choose Google App Engine to develop, test, and scale their application without managing the underlying infrastructure.

  Example (SaaS): A company needs a solution for team collaboration and document sharing. They implement Google Workspace to provide email, document editing, and storage for all employees without needing local software installations.

Exercise 2: Exploring Google Cloud Platform's Core Services

1. **Objective**: Get acquainted with the core services provided by Google Cloud Platform.
2. **Steps**:
   - Access the Google Cloud Console and navigate to the list of GCP services.

     Google Cloud Platform services: Compute, Storage, Networking, Data Analytics, Machine Learning, Security and Identity, Management Tools, Developer Tools and IoT

   - Explore and describe the purpose of the following core services:

     Compute Engine

     Provides virtual machines (VMs) that run on Google's infrastructure. Users can customize their VM configurations, install any OS, and scale resources as needed, making it suitable for various applications from simple websites to complex computing tasks.

     Google Kubernetes Engine (GKE)

     A managed Kubernetes service that simplifies the deployment, management, and scaling of containerized applications. GKE automates tasks like cluster provisioning and maintenance, allowing developers to focus on building and running applications without managing the underlying infrastructure.

App Engine

A Platform as a Service (PaaS) that enables developers to build and host web applications without worrying about the underlying infrastructure. It automatically handles scaling, load balancing, and monitoring, making it easy to deploy applications in multiple languages.

Cloud Storage

Offers scalable and secure object storage for unstructured data. Users can store and retrieve any amount of data at any time, making it ideal for backups, data archiving, and serving web content.

BigQuery

A fully managed, serverless data warehouse designed for large-scale data analytics. It allows users to run fast SQL queries on large datasets, enabling real-time analysis and insights without the need for complex infrastructure management.

- For each service, identify a potential use case in a business scenario.

Compute Engine

A financial services company needs to run complex risk analysis simulations on large datasets. They use Compute Engine to provision high-performance VMs that can be scaled up or down based on workload demands.

Google Kubernetes Engine (GKE)

An e-commerce platform wants to deploy a microservices architecture for better scalability and reliability. They use GKE to manage their containerized applications, allowing seamless updates and rollbacks with minimal downtime.

App Engine

A startup develops a mobile application that requires a backend to handle user authentication and data processing. They use App Engine to quickly deploy the backend, benefiting from automatic scaling during high traffic periods.

Cloud Storage

A media company needs to store and distribute large video files for streaming. They use Cloud Storage to store the videos securely and serve them efficiently to users worldwide.

BigQuery

A retail chain wants to analyze customer purchase data to optimize inventory and marketing strategies. They use BigQuery to run fast SQL queries on their large datasets, enabling data-driven decision-making and real-time insights.

3. **Questions**:
    - What is the primary use case of Compute Engine?

      Answer: The primary use case of Compute Engine is to provide scalable virtual machines for running applications, allowing businesses to customize configurations and handle varying workloads, such as hosting websites, running enterprise applications, or performing data processing tasks.

    - How does Google Kubernetes Engine (GKE) simplify the management of containerized applications?

      Answer: Google Kubernetes Engine (GKE) simplifies the management of containerized applications by automating tasks such as cluster provisioning, scaling, and maintenance. It offers built-in load balancing, monitoring, and logging, allowing developers to focus on deploying and managing applications without worrying about the underlying infrastructure.

    - What advantages does Cloud Storage offer for data management?

      Answer: Cloud Storage offers several key advantages for data management such as scalability, durability, accessibility, security, cost-Effectiveness and integration with other systems.

    - Why would a business choose BigQuery for their data analysis needs?

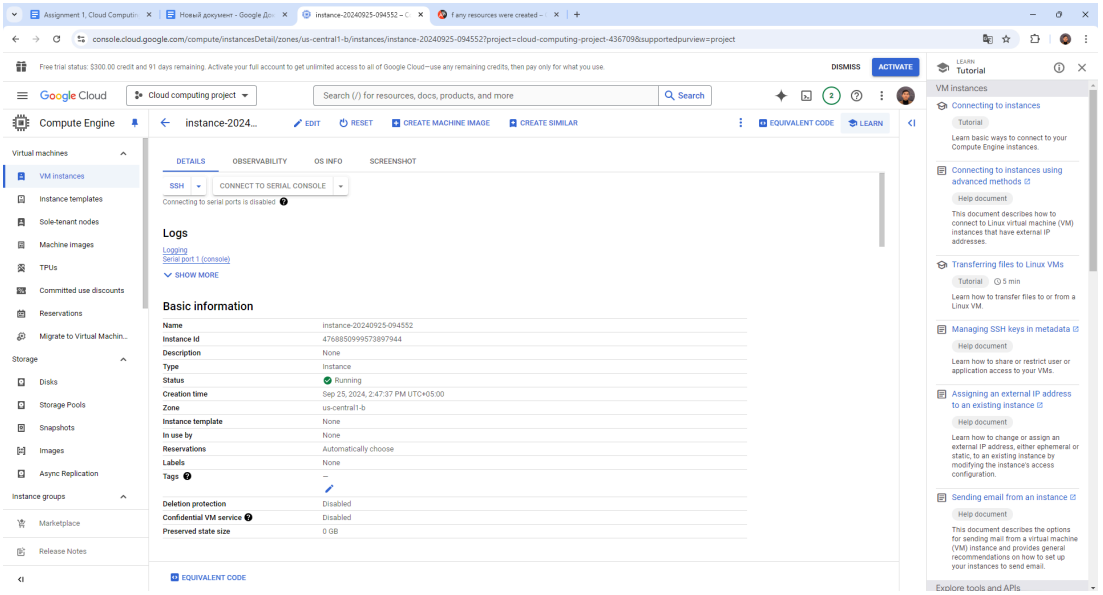
      Answer: A business would choose BigQuery for data analysis due to its ability to handle large-scale datasets efficiently, offering fast SQL queries and real-time analytics. Its serverless architecture eliminates the need for infrastructure management, while built-in machine learning capabilities and seamless integration with other Google Cloud services enhance data insights and decision-making.

Exercise 3: Creating and Managing Virtual Machines with Compute Engine

1. **Objective**: Learn how to create, manage, and interact with virtual machines (VMs) using Compute Engine.
2. **Steps**:
    - In the Google Cloud Console, navigate to Compute Engine and create a new VM instance.

      VM instance name: instance-20240925-094552

Picture 1 - Creating VM instance



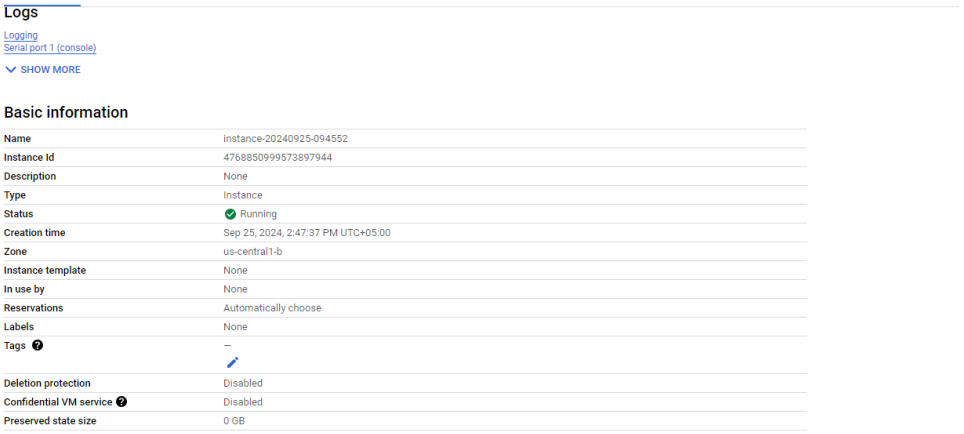
- Configure the VM with specific parameters, such as the machine type, region, and operating system.

Machine type: e2-medium

Region: us-central1-b

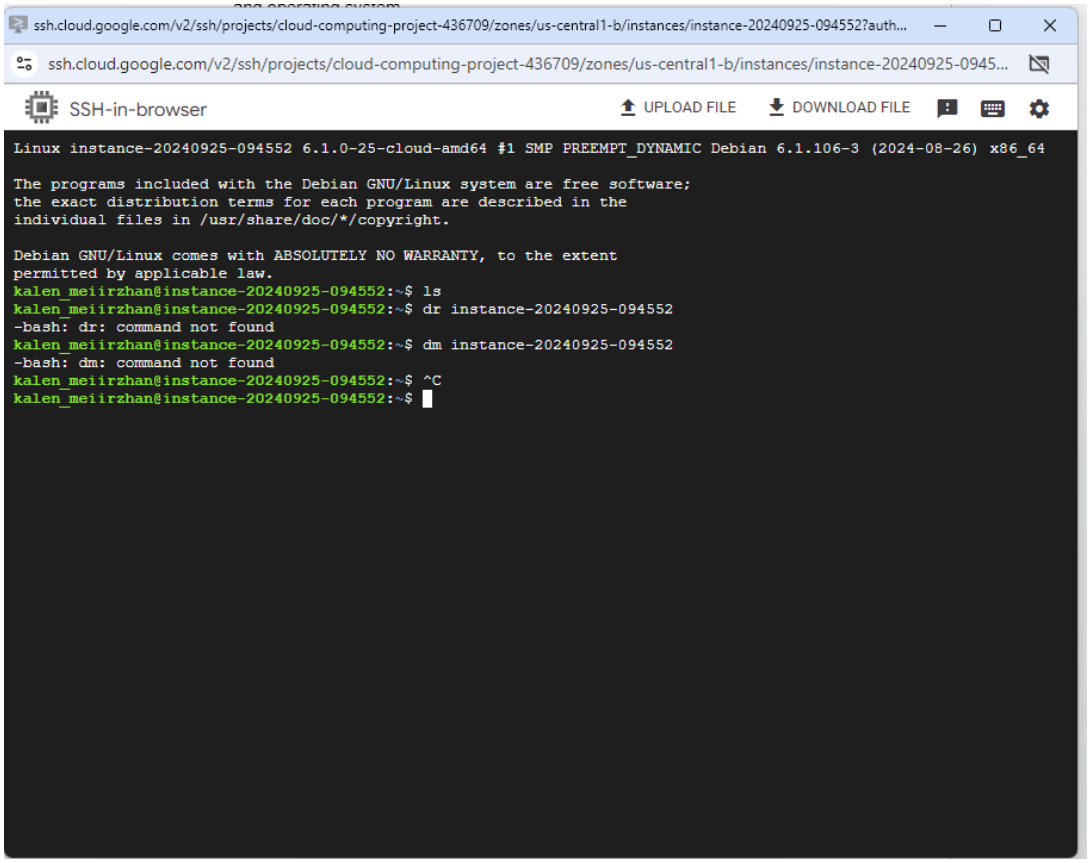
Operating system: Linux instance-20240925-094552

Picture 2 - VM's basic information



- Connect to the VM using SSH and install a basic web server (e.g., Apache or Nginx).

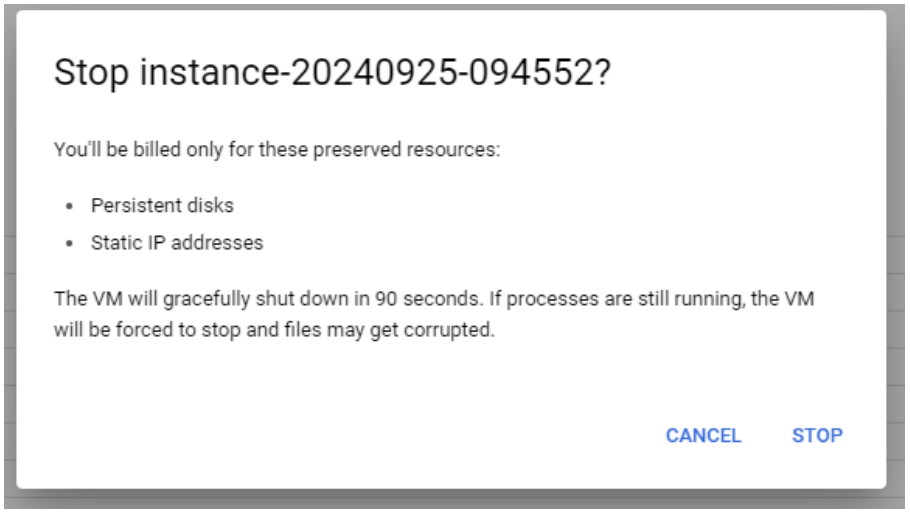
Picture 3 - SSH in browser



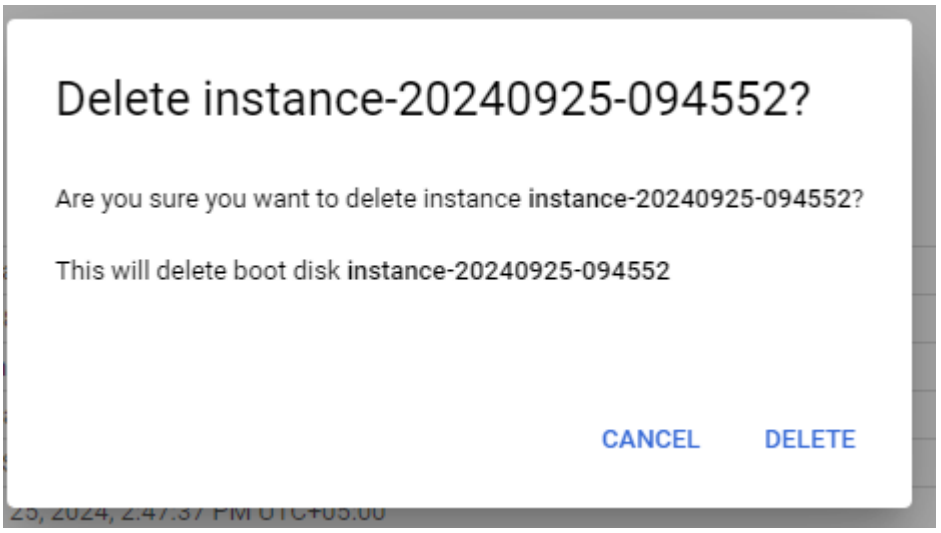
- Stop, start, and delete the VM through the console.

  In the pictures 4 and 5 we can see that our current VM instance was started, stopped then deleted.

Picture 4 - Stop notification from GCS



Picture 4 - Delete notification from GCS



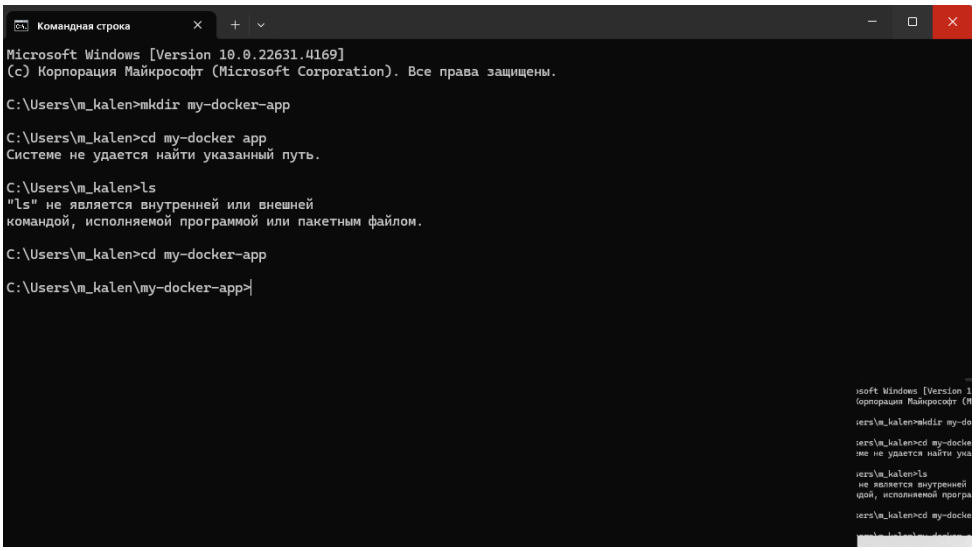
3. **Questions**:
    - What steps did you follow to create the VM?
        1. Login into Google Cloud Console.
        2. Added payment card and started free trial
        3. Navigate into Compute engine
        4. Created instance
        5. Configure VM settings
        6. Review and created
        7. Connected to VM
        8. Started, stopped then deleted instance
    - How did you connect to the VM, and what commands did you use to install the web server?
        - In the VM instance page I clicked the SSH button next to it. It's opened terminal window directly in my browser
    - What happens to the VM and its data when it is stopped versus when it is deleted?
        - When we stop the VM all the processes in VM will be just stopped, but when we delete it all the processes in VM will be deleted and we will not be able to start VM again.

Exercise 4: Deploying a Containerized Application on Google Kubernetes Engine (GKE)

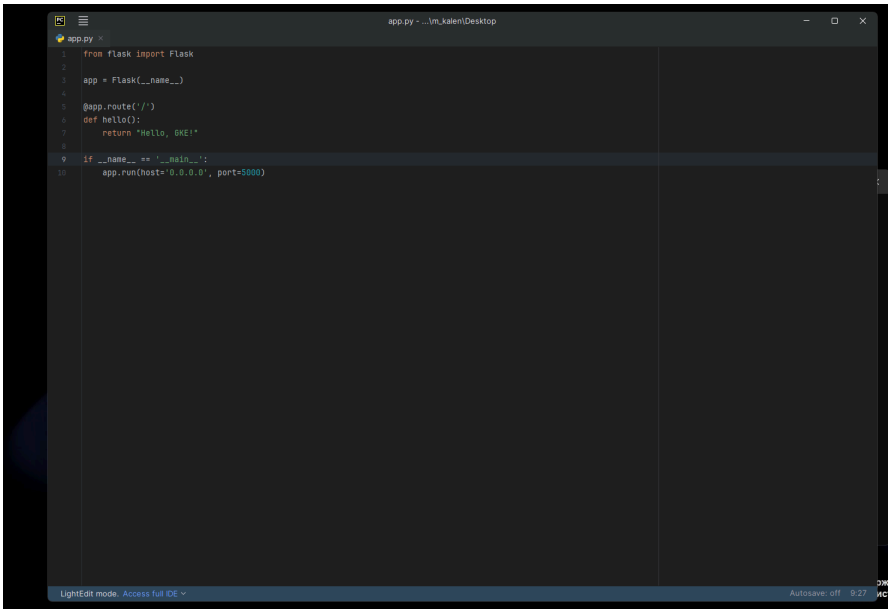
1. **Objective**: Understand how to deploy and manage containerized applications using Google Kubernetes Engine.
2. **Steps**:

    Creating a dir:

Picture 5 - Creating directory



Picture 6 - Simple application on python



Picture 7 - Created a dockerfile

```
FROM python:3.9-slim
WORKDIR /app
COPY app.py .
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
EXPOSE 5000
CMD ["python", "app.py"]
```

Build the docker image:

- docker build -t my-docker-app .

Push the Docker Image to Google Container Registry

- docker tag my-docker-app gcr.io/[autopilot-cluster-1]/my-docker-app

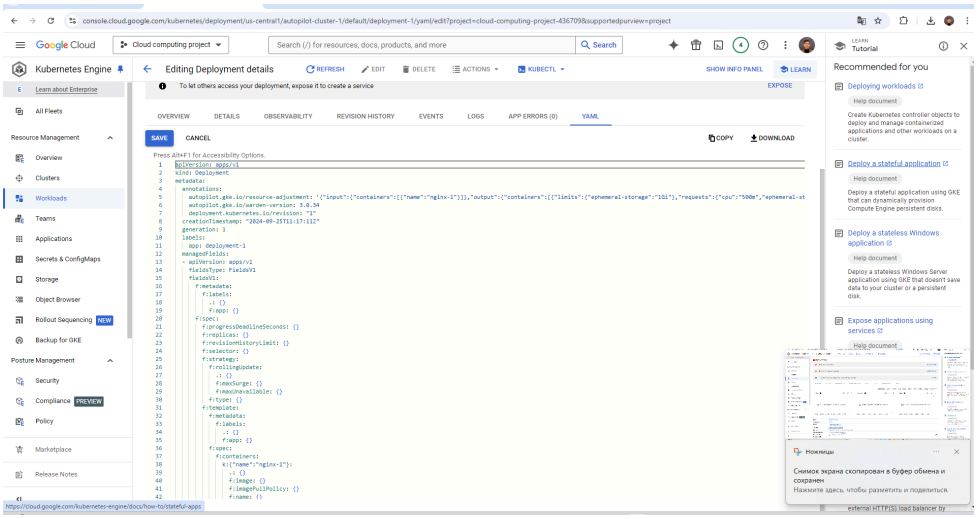
Log in to google container Registry

- gcloud auth configure-docker

Push the Docker image:

- docker push gcr.io/[autopilot-cluster-1]/my-docker-app

Picture 8 - Deployment GKE

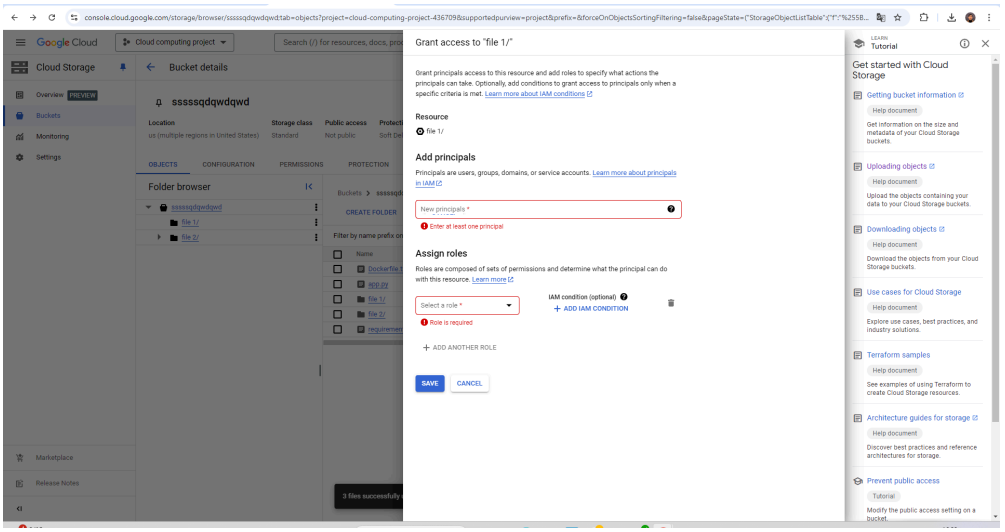


3. **Questions**:
    - How did you create and push the Docker container to GCR?
    - **Answer: Create Application Files, Create a Dockerfile: Build the Docker Image: Tag the Image: Push the Image to GCR:**
    - What steps were involved in setting up the GKE cluster?
    - **Answer: Access Google Cloud Console, Navigate to GKE, Create a Cluster, Verify**
    - How did you verify that your application was successfully deployed and accessible?
    - **Answer:** I verified it by checking workloads, by getting external IP and by getting access in browser

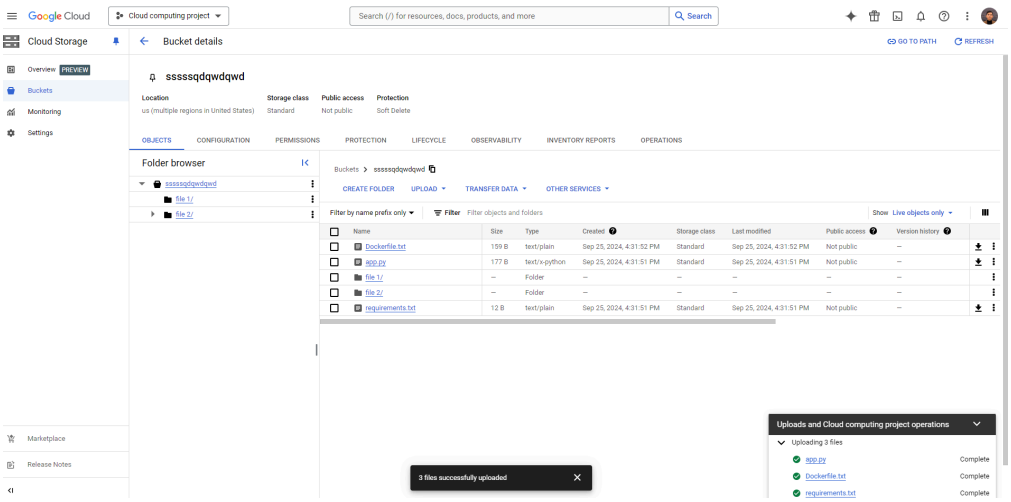
Exercise 5: Storing and Accessing Data in Google Cloud Storage

1. **Objective**: Learn how to store, manage, and access data using Google Cloud Storage.
2. **Steps**:
    - In picture 9 I granted access to the public and also for giving permissions for someone I needed a Google account, so there's only this form where you can grant it. But the logic is the same.
    - In picture 10 there's a list of uploaded files of different types and created folders from bucket

Picture 9 - Giving permissions and access



Picture 10 - Loading/Creating files and folders in bucket

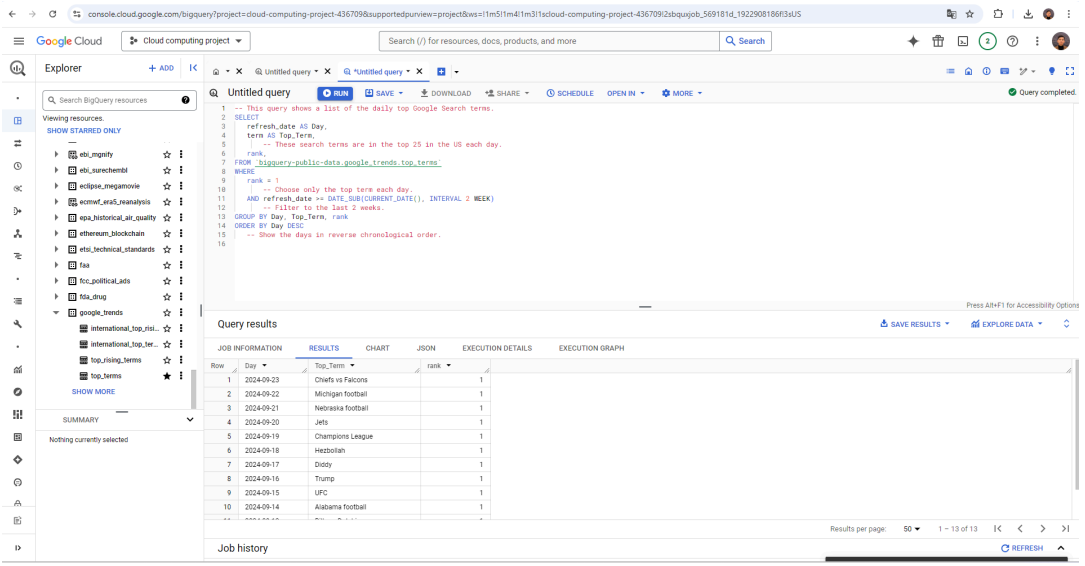


3. **Questions**:
    - I created a bucket in Google Storage service, there is a list of different optional parameters.
    - We set a bucket to public, then everyone has access to it; private it will be accessible only for us and users with whom we share access.
    - We can manage access permissions for individual files in a bucket by clicking on files/folder then setting up the permissions for google accounts of users.

Exercise 6: Analyzing Data with BigQuery

1. **Objective**: Perform data analysis tasks using BigQuery.
2. **Steps**:
    - In picture 11 displayed training on dataset given by Google

Picture 11 - Dataset provided by Google

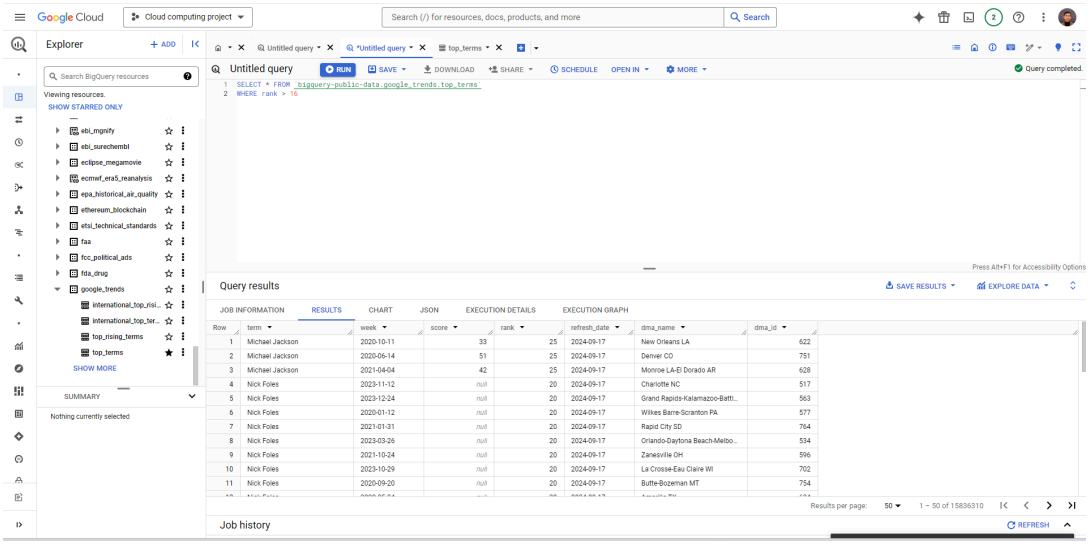


- Write and execute SQL queries to perform basic data analysis, such as filtering, aggregation, and sorting.

In picture 12 displayed the filtering top_term table by the rank which is higher that 17. SQL code: `SELECT * FROM` `` `bigquery-public-data.google_trends.top_terms` ``

```
WHERE rank > 16
```

Picture 12 - Filtering the data

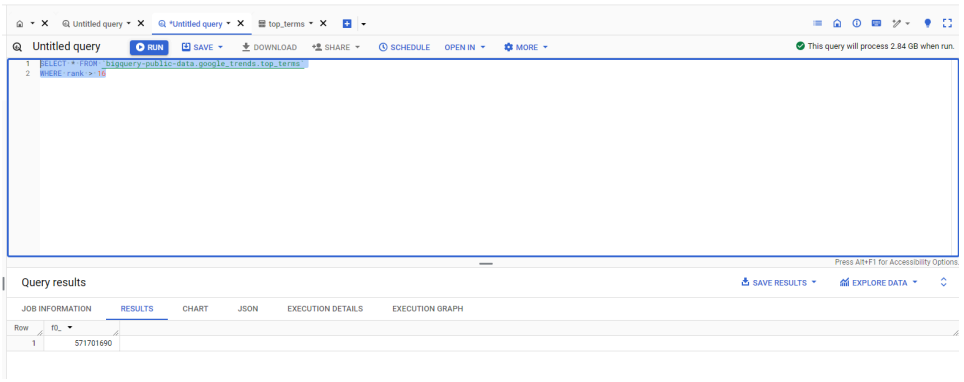


In picture 13 displayed the aggregation of data by getting sum of rank

sql code:

```
SELECT SUM(rank) FROM `bigquery-public-data.google_trends.top_terms`
```

Picture 13 - SQL aggregation

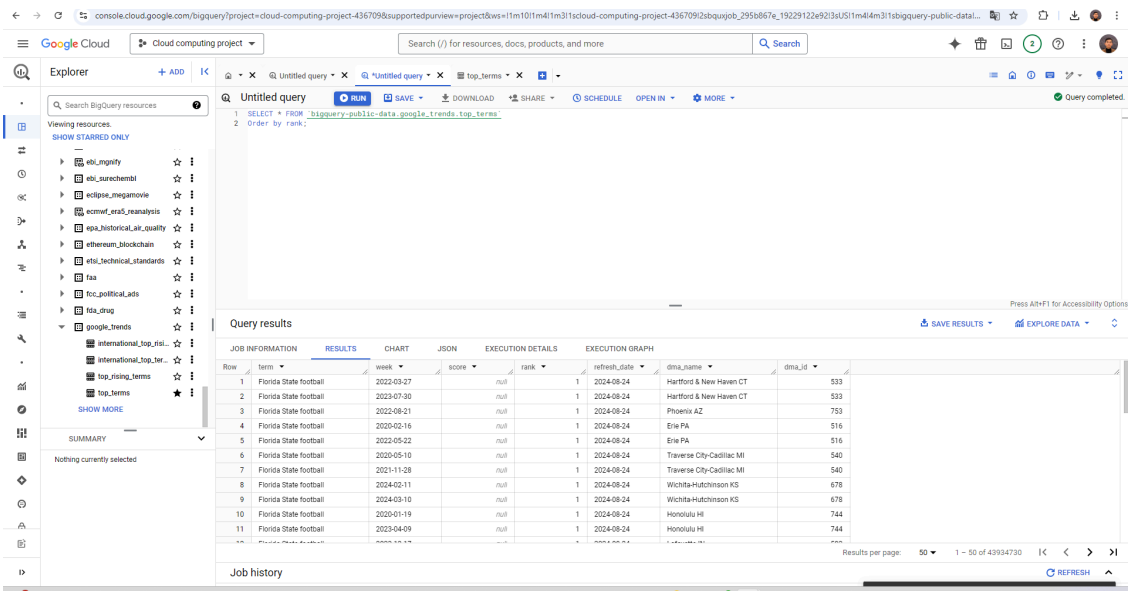


In picture 13 displayed the sorting of data by rank

sql code: `SELECT * FROM `bigquery-public-data.google_trends.top_terms`

`Order by rank;`

Picture 14 - SQL sorting



- In picture 15 displayed how it looks like visualization of data by a data chart provided by Google. There is a visualization of SQL where each row was sorted by rank.

Picture 14 - SQL sorting

3. **Questions**:
   - What steps did you take to create a dataset and table in BigQuery?
   - **Answer:** I clicked to BigQuery studio where Google offers a ready dataset and tables for self study.
   - How did you write and execute SQL queries in BigQuery?
   - **Answer:** There in BigQuery studio exists an untitled Query where you could execute SQL that you want.
   - What insights were you able to derive from the data analysis?
   - **Answer:** It's very convenient for usage queries, to work with big data. BigQuery gives such an opportunity to make analysis from data such as visualization, execution details, etc.