Assignment 1, Cloud Computing

Put all deliverables into github repository in your profile. Share link to google form 24 hours before defense. Defend by explaining deliverables and answering questions.
Deliverables: report in pdf
Google form:
https://docs.google.com/forms/d/e/1FAIpQLSe0GyNdOYlvM1tX_I_CtlPod5jBf-ACLGdHYZq1gVZbUeBzIg/viewform?usp=sf_link

Exercise 1: Understanding Cloud Computing Models

1. **Objective**: Explore different cloud computing models and understand their key differences.
2. **Steps**:
    - Research the three primary cloud service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).
    - Create a table comparing these models in terms of control, flexibility, and use cases.
    - Identify examples of services offered by Google Cloud Platform (GCP) under each model.
3. **Questions**:
    - What are the main differences between IaaS, PaaS, and SaaS?
    - Which GCP services fall under each of these models?
    - Provide a real-world example where each cloud service model might be the most appropriate choice.
      IaaS: hosting a big company's entire IT infrastructure
      PaaS: custom web app
      SaaS: email services

Cloud service model	Control	Flexibility	Use Cases	GCP Examples
IaaS	High control over infrastructure, OS, networking	Very flexible, users manage the infrastructure	Hosting websites, virtual machines, storage	Compute Engine, Cloud Storage
PaaS	Moderate control over applications and data	Moderate flexibility, developer-focused	Developing, testing, and deploying applications	App Engine, Cloud Functions
SaaS	Minimal control, mostly just using the software	Low flexibility, ready-to-use software	Using email, CRM, or collaboration tools	Google Workspace, Gmail

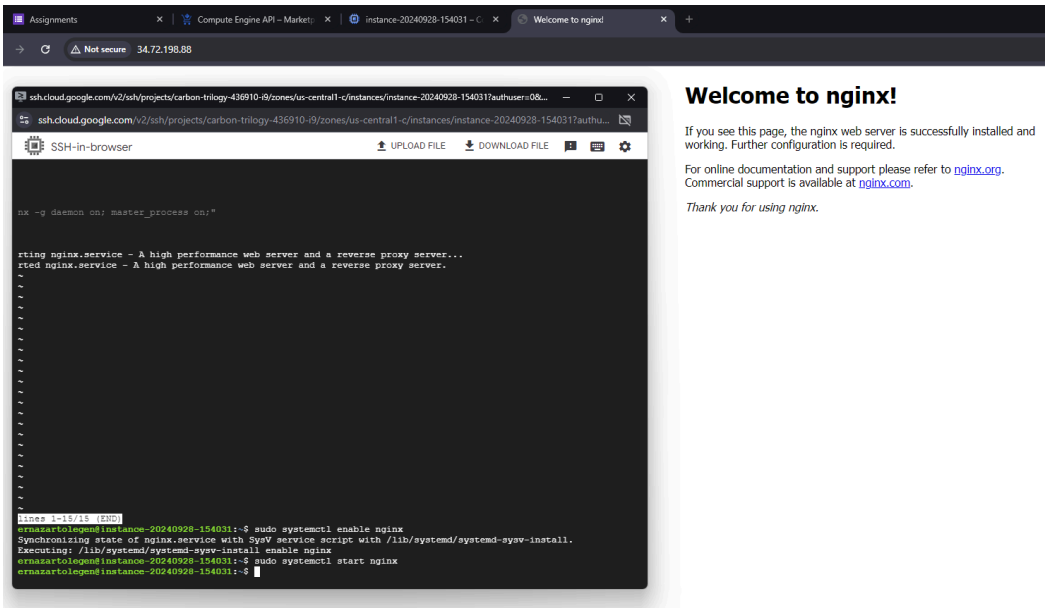
Exercise 2: Exploring Google Cloud Platform's Core Services

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.
    - Explore and describe the purpose of the following core services:
        - Compute Engine
          Offers scalable virtual machines (VMs) to run applications.
        - Google Kubernetes Engine (GKE)
          Simplifies deployment and management of containerized applications using Kubernetes.
        - App Engine
          Fully managed platform for building scalable web apps.
        - Cloud Storage
          Object storage for unstructured data like files, backups, or media.
        - BigQuery
          A fully-managed, serverless data warehouse for analyzing large datasets.
    - For each service, identify a potential use case in a business scenario.
3. **Questions**:
    - What is the primary use case of Compute Engine?
      Hosting web applications or running custom workloads like databases.
    - How does Google Kubernetes Engine (GKE) simplify the management of containerized applications?
      GKE automates tasks like scaling, updates, and load balancing for containers, making it easier to manage complex, container-based applications without manual intervention.
    - What advantages does Cloud Storage offer for data management?
      Cloud Storage provides scalable, highly available, and cost-effective storage. It's great for storing large amounts of data, offering durability, access from anywhere, and seamless integration with other GCP services.
    - Why would a business choose BigQuery for their data analysis needs?
      BigQuery handles big data efficiently with fast SQL queries, scalability, and the ability to analyze petabytes of data without managing infrastructure. It's perfect for businesses doing deep data analysis or reporting.

Exercise 3: Creating and Managing Virtual Machines with Compute Engine

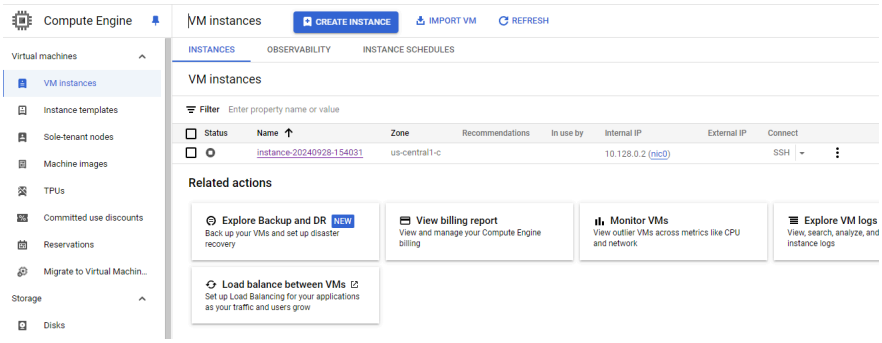
1. **Objective**: Learn how to create, manage, and interact with virtual machines (VMs) using Compute Engine.
2. **Steps**:

- In the Google Cloud Console, navigate to Compute Engine and create a new VM instance.
- Configure the VM with specific parameters, such as the machine type, region, and operating system.
- Connect to the VM using SSH and install a basic web server (e.g., Apache or Nginx).
  Changed firewalls HTTP traffic to on, and now i can visit my server



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

3. **Questions**:
   - What steps did you follow to create the VM?
     Entered to Compute Engine API and clicked Enable in the Google Cloud. Then "Create Instance". Selected preferable characteristics of VM such as, OS, machine type, region and clicked on the Create button.

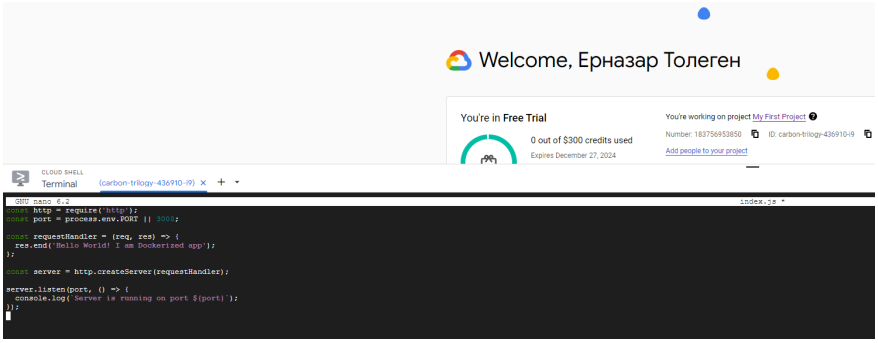


   - How did you connect to the VM, and what commands did you use to install the web server?
     sudo apt update
     sudo apt install nginx
   - What happens to the VM and its data when it is stopped versus when it is deleted?

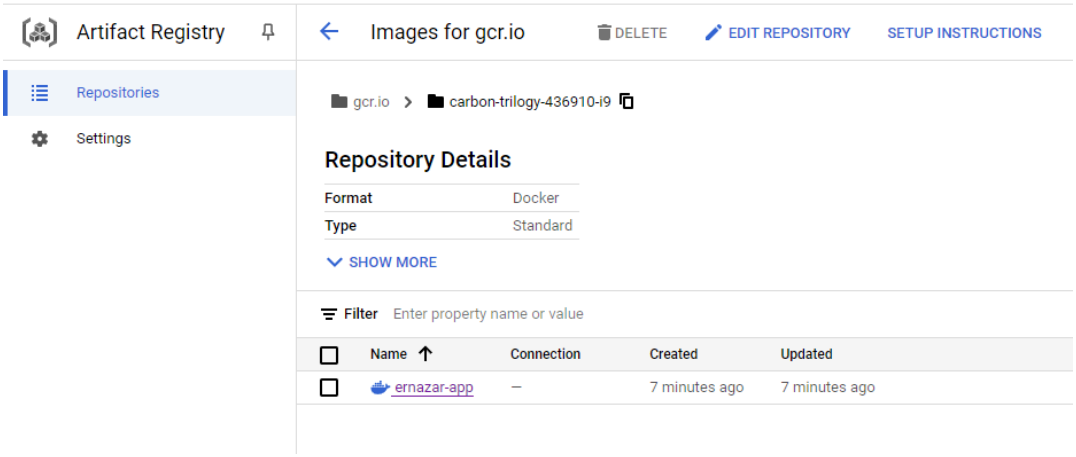
If it is stopped, data is still approachable, we can enter to VM . But when we delete our VM, data will be deleted fully and VM is not approachable.

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**:
    - Create a simple Docker container for a web application.

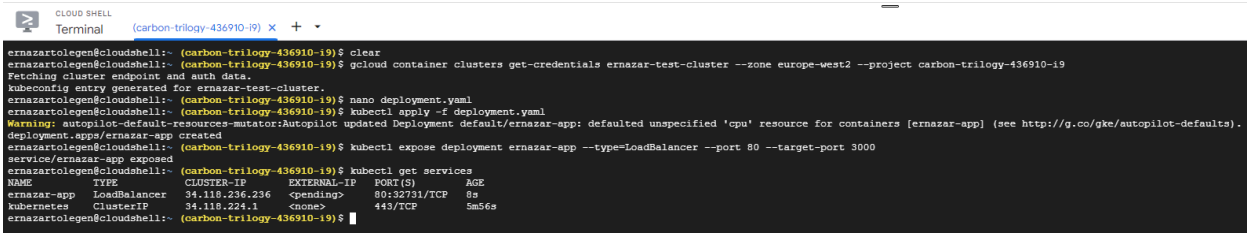


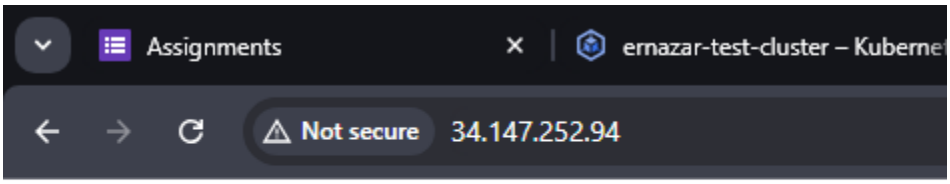
    - Push the container image to Google Container Registry (GCR).

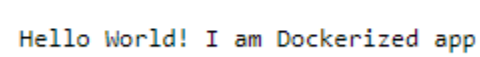


    - Create a GKE cluster in Google Cloud Console.
    - Deploy the containerized application to the GKE cluster.

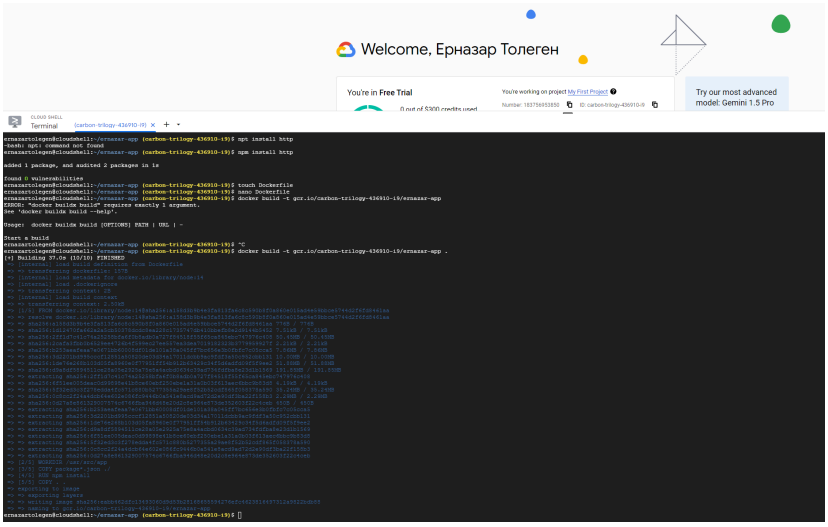
- Expose the application to the internet and verify its accessibility.

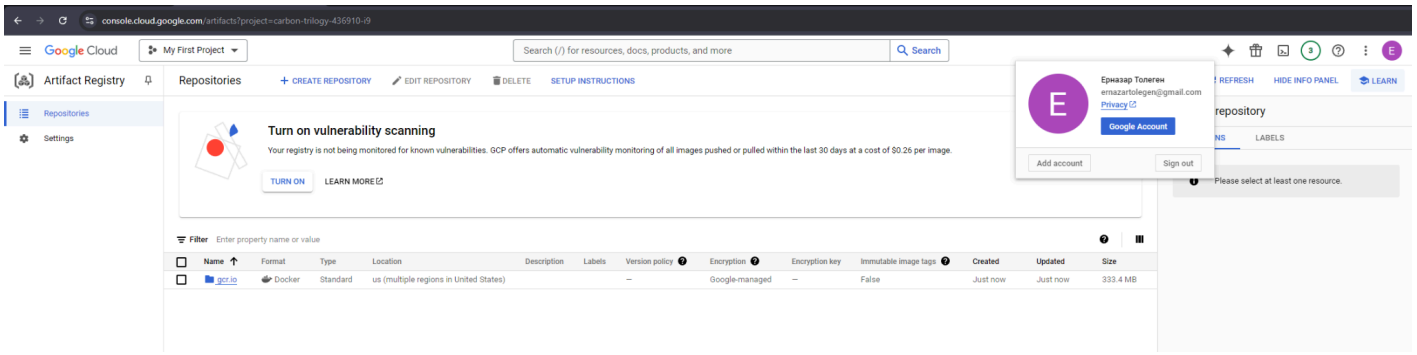




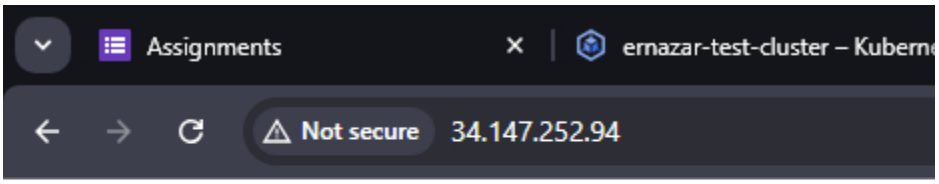


3. **Questions**:
   - How did you create and push the Docker container to GCR?
     First, I created Web application using “Cloud Shell”, then created Dockerfile. In Cloud Shell built docker using **docker build -t** command and pushed Docker Image to GCR





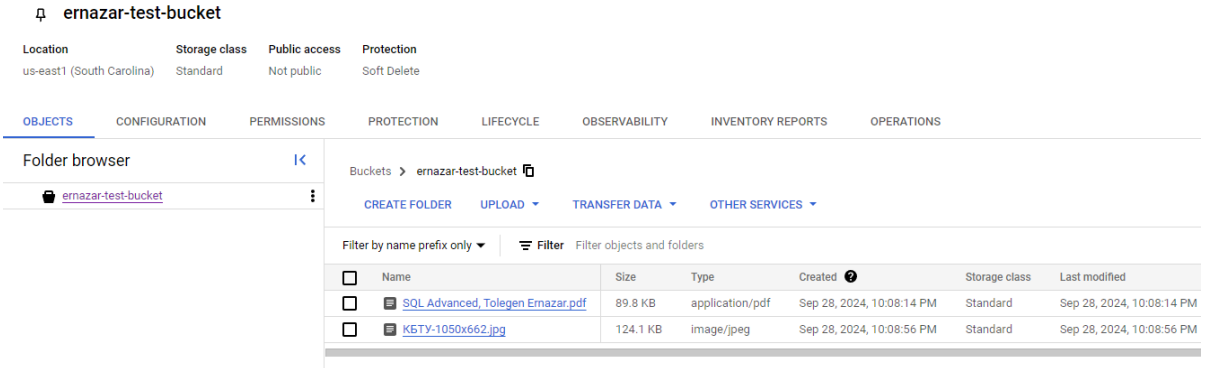
- What steps were involved in setting up the GKE cluster?
- How did you verify that your application was successfully deployed and accessible?
  After exposing my app, waited a few minutes for the external IP address to be provisioned. Then visited this IP address and it showed me my Web application




```
Hello World! I am Dockerized app
```

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**:
   - Create a new Cloud Storage bucket in the Google Cloud Console.
   - Upload various types of files (e.g., text, images, videos) to the bucket.



   - Set access permissions for the bucket and test public and private access to the files.

- Use the Cloud Console to download, move, and delete files in the bucket.

3. **Questions**:
   - How do you create a Cloud Storage bucket, and what options are available during setup?

     Location type: Choose a **Region**, **Dual-region**, or **Multi-region** based on your geographic storage needs.

     Storage Class: Options include **Standard**, **Nearline**, **Coldline**, or **Archive**, depending on the access frequency and cost.

     Access Control: Choose between **Uniform access** (bucket-wide permissions) or **Fine-grained access** (object-level permissions).

     Encryption: Use Google-managed keys or customer-managed keys for encryption.
   - What are the differences between setting a bucket to public versus private?

     If it is public, **anyone** can access objects inside bucket(read-only), while if it is private, only users or services **with explicit permissions** can access the objects
   - How can you manage access permissions for individual files in a bucket?

     Navigate to the **Select the file**, then **Permissions tab**. Add or remove permissions for specific users or service accounts. Use roles such as **Storage Object Viewer** (read access) or **Storage Object Admin** (full access).

Exercise 6: Analyzing Data with BigQuery

1. **Objective**: Perform data analysis tasks using BigQuery.
2. **Steps**:
   - Access BigQuery in the Google Cloud Console.
   - Create a dataset and table by importing a sample dataset provided by Google.

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

Untitled query

RUN SAVE DOWNLOAD SHARE SCHEDULE OPEN IN MORE

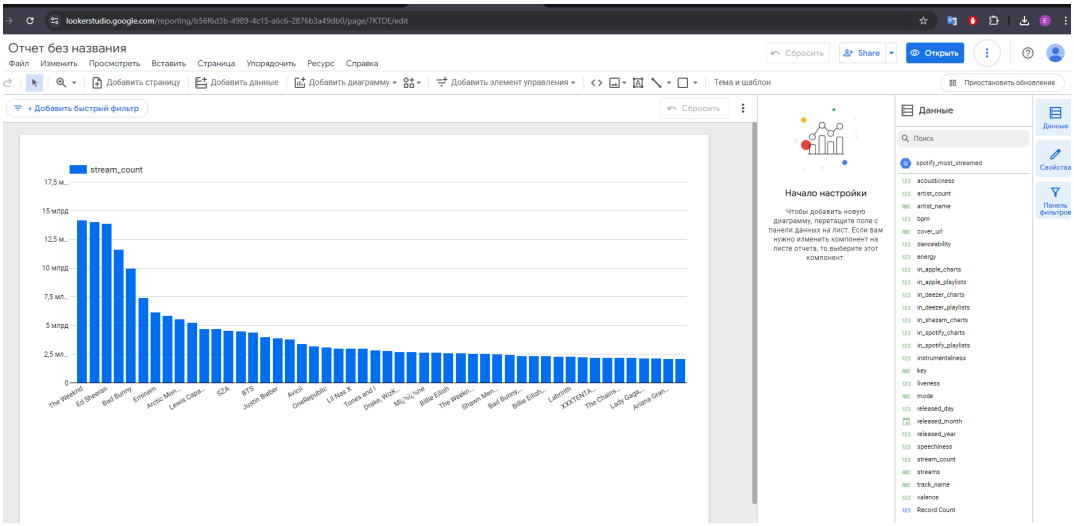
```
SELECT artist_name, COUNT(track_name) as tracks, SUM(CAST(streams as integer)) as stream
FROM `carbon-trilogy-436910-i9.ernazar_test_dataset.spotify_most_streamed`
where released_year >= 2010
group by artist_name
order by stream desc
LIMIT 1000
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	artist_name	tracks	stream
1	The Weeknd	22	14185552870
2	Taylor Swift	34	14053658300
3	Ed Sheeran	9	13908947204
4	Harry Styles	17	11608645649
5	Bad Bunny	19	9997799607
6	Olivia Rodrigo	7	7442148916
7	Bruno Mars	4	5846920599
8	Imagine Dragons	3	5272484650
9	Lewis Capaldi	3	4734698360
10	Doja Cat	6	4702294655
11	SZA	19	4557811204
12	Adele	4	4508746590
13	BTS	8	4389891591
14	Arctic Monkeys	3	4352686021
15	The Neighbourhood	3	4010009939
16	Justin Bieber	3	3919813522
17	Avicii	2	3426754746
18	Dua Lipa	3	3227639000
19	OneRepublic	2	3097149603
20	Kendrick Lamar	12	3033135947
21	Lil Nas X	3	2988745319
22	Tones and I	1	2864791672
23	Post Malone, Swae Lee	1	2808096550
24	Drake, WizKid, Kyla	1	2713922350
25	James Arthur	2	2686344050

- Visualize the results using Google Data Studio or another visualization tool.



3. **Questions**:
    - What steps did you take to create a dataset and table in BigQuery?
    - How did you write and execute SQL queries in BigQuery?
    - What insights were you able to derive from the data analysis?