Zhanat Seitkuzhin

Assignment 1, Cloud Computing

Exercise 1: Understanding Cloud Computing Models

Question 1: What are the main differences between IaaS, PaaS, and SaaS?

The main differences between IaaS, PaaS, and SaaS lie in how much control the user has over the infrastructure and how much responsibility the cloud provider has.

1. **Infrastructure as a Service (IaaS)** is the most flexible option, where you manage the infrastructure, but the cloud provider takes care of the physical hardware. You're in charge of everything from the operating system upwards, including security, storage, and networking. It's perfect if you need full control over your environment, but it also means you're responsible for keeping everything running smoothly.
2. **Platform as a Service (PaaS)** provides a platform to build and run your applications without managing the underlying infrastructure. You only focus on coding and deploying applications, while the cloud provider manages the servers, storage, and networking. This makes it faster and simpler for developers to build apps, though you lose some control over customization at the infrastructure level.
3. **Software as a Service (SaaS)** provides access to fully developed software without worrying about the infrastructure at all. You simply use the software, with no need to handle installation, maintenance, or updates. It's the least flexible model, but the trade-off is that it requires almost no technical management from the user.

Question 2: Which GCP services fall under each of these models?

1. **IaaS GCP Services:**
    a. **Compute Engine:** create and manage VMs with complete control over your computing resources.
    b. **Cloud Storage:** flexible solution for storing and retrieving data anytime.
    c. **Cloud Load Balancing:** helps distribute incoming traffic across multiple VMs, ensuring reliability and performance.

2. **PaaS GCP Services:**

   a. **App Engine:** fully managed platform lets you build and deploy applications without worrying about server maintenance.
   b. **Cloud Functions:** perfect for running code in response to specific events, eliminating the need for server management.
   c. **Cloud Run:** allows to deploy and manage containerized applications in a serverless environment, making it super easy to scale.

3. **SaaS GCP Services:**
   a. **Google Workspace:** includes tools like Gmail, Google Drive, and Google Docs.
   b. **Cloud Identity:** helps to manage user access and security for Google applications.
   c. **Google BigQuery:** fully managed data warehouse that makes it easy to analyze large datasets without needing to worry about the infrastructure behind it.

Question 3: Provide a real-world example where each cloud service model might be the most appropriate choice.

1. **IaaS Example:**
   a. Qazsport needs to deliver high-quality live streaming of the World Nomad Games and anticipates a spike in viewership. By leveraging IaaS, they can quickly scale their virtual machines to handle increased traffic, ensuring a seamless streaming experience. This model allows them to maintain control over their infrastructure while optimizing performance and storing large volumes of video content without the costs of physical servers.
2. **PaaS Example:**
   a. Cerebra (AI-powered software designed for early stroke detection) aims to develop a new application, and by utilizing PaaS, the team can focus on coding and refining their AI algorithms without worrying about server management or infrastructure complexities. This would enable them to quickly iterate on features, updates, and respond to feedback, while ensuring the software is scalable and efficient for healthcare providers.
3. **SaaS Example:**
   a. My mom is opening a small kindergarten and needs reliable tools for email, document collaboration, and file storage. By adopting SaaS solutions like

Google Workspace, she can access all the necessary applications without worrying about installation, updates, or infrastructure management.

Cloud Service Models Comparison Table:

Model	Control	Flexibility	Use Cases
Infrastructure as a Service (IaaS)	Full control over infrastructure (VMs, storage networks)	Highly flexible (users manage the OS, storage, and apps)	Hosting websites, managing large datasets, custom enterprise apps
Platform as a Service (PaaS)	Focused more on app development, less on infrastructure	Moderately flexible (developers control the apps, but not the underlying infrastructure)	Developing, testing, and deploying apps without handling servers
Software as a Service (SaaS)	Least control (only interact with the software)	Low flexibility (users can configure settings but don't manage infrastructure)	Ready-to-use software for end users like email, collaboration tools, CRM systems

Zhanat Seitkuzhin

Exercise 2: Exploring Google Cloud Platform's Core Services

App Engine:

- ***Purpose:*** **App Engine** is a platform for building and deploying applications without worrying about the underlying infrastructure. It automatically handles scaling, load balancing, and application health monitoring.
- ***Potential Use Case:*** A mobile app development team can use **App Engine** to host their application's backend services, allowing them to focus on development while GCP manages the scaling and availability.

Cloud Storage:

- ***Purpose:*** **Cloud Storage** offers a highly scalable and secure way to store and retrieve data. It provides different classes of storage to optimize costs based on access frequency.
- ***Potential Use Case:*** A media company can use Cloud Storage to store large video files for on-demand streaming, benefiting from its scalability and durability while ensuring fast access for viewers.

Question 1: What is the primary use case of Compute Engine?

Compute Engine:

- ***Purpose****:* the primary use case of **Compute Engine** is to provide scalable virtual machines for running applications, allowing businesses to customize their computing resources based on their specific workload requirements.
- ***Use Case:*** a startup developing a web application can use **Compute Engine** to host its backend services, enabling them to easily scale up or down based on user demand, especially during peak traffic times.

Question 2: How does Google Kubernetes Engine (GKE) simplify the management of containerized applications?

GKE simplifies the management of containerized applications by automating tasks like deployment, scaling, and load balancing. It provides an easy-to-use interface for managing Kubernetes clusters, allowing teams to focus on developing applications rather than handling infrastructure concerns.

Purpose: **GKE** is a managed service for deploying, managing, and scaling containerized applications using Kubernetes. It simplifies the orchestration of containers, making it easier to manage complex applications.

Use Cases: A software company can use **GKE** to manage its microservices architecture, ensuring that each service is efficiently deployed, updated, and scaled automatically based on usage patterns.

Question 3: Why would a business choose BigQuery for their data analysis needs?

A business would choose **BigQuery** for its data analysis needs due to its ability to handle massive datasets quickly and efficiently, its serverless architecture that eliminates the need for infrastructure management. Additionally, its powerful SQL capabilities enable complex queries for deep insights into data.

Purpose: **BigQuery** is a fully managed, serverless data warehouse that allows businesses to run super-fast SQL queries on large datasets. It's designed for high-speed analytics and data processing.

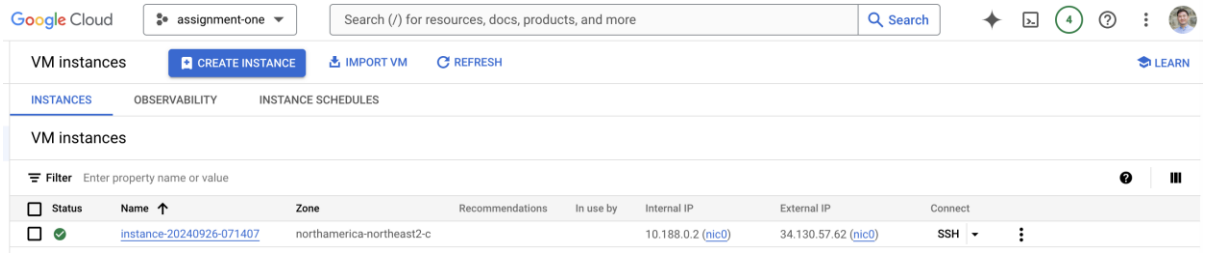
Use Cases**:** A retail chain can use **BigQuery** to analyze sales data and customer behavior across its stores, gaining insights that drive marketing strategies and inventory management.

Zhanat Seitkuzhin

Exercise 3: Creating and Managing VMs with Compute Engine

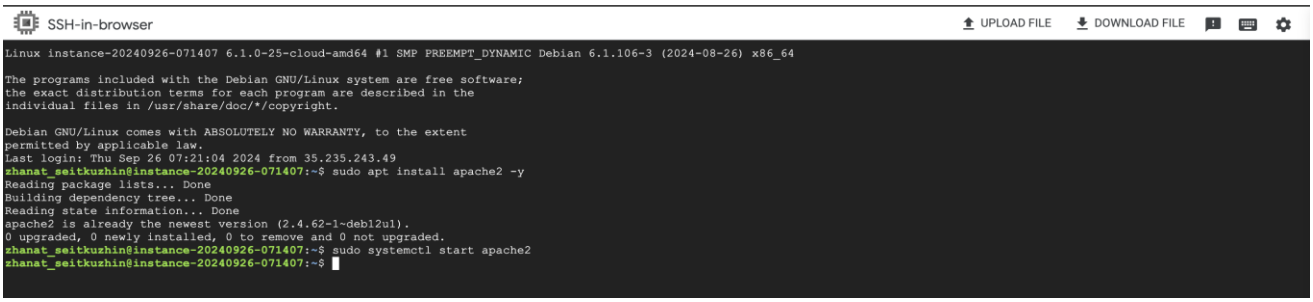
Question 1: What steps did you follow to create the VM?

1. Access the Google Cloud Console
2. Navigate to Compute Engine
3. Create a New VM Instance
4. Configure the VM Settings
5. Create the VM
6. Access the VM (via SSH)



Question 2: How did you connect to the VM, and what commands did you use to install the web server?

1. I found the VM and clicked on SSH next to its name. This opened an SSH terminal in my browser.
2. To install Apache, use the following command: sudo apt install apache2 –y



Question 3: What happens to the VM and its data when it is stopped versus when it is deleted?

- **Stopped VM**: Temporarily powered down (transitions to a terminated state), data and configurations are retained. We can restart the VM, and all files, configurations, or applications stored on the disk will remain intact. Billing for CPU and memory resources stops, but charges will remain for persistent disk storage.
- **Deleted VM:** The VM is permanently removed from Google Cloud. Once deleted, it cannot be restarted or recovered, and associated data is lost (unless persistent

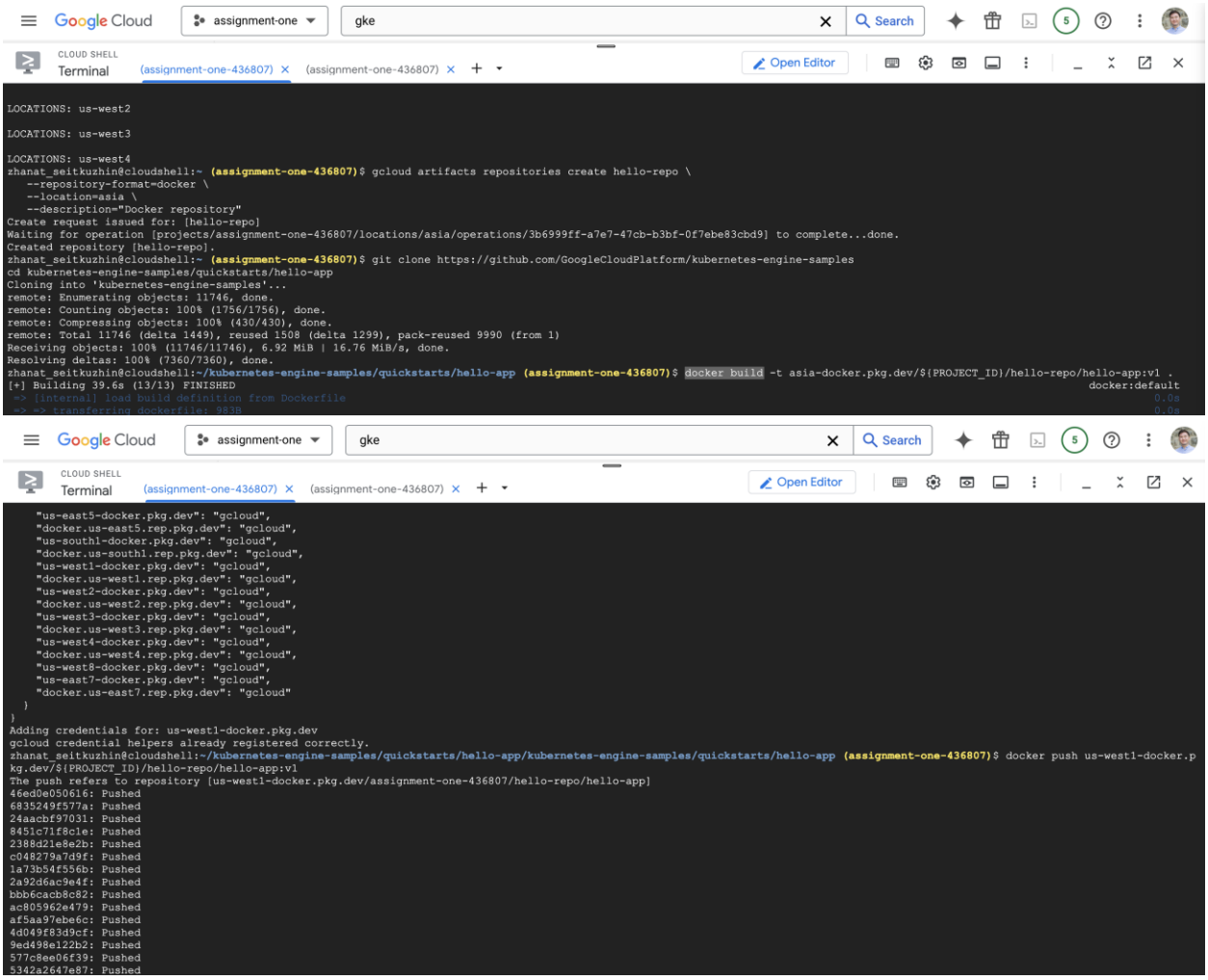
disks are retained). Billing stops for compute and storage resources associated with the VM.

Zhanat Seitkuzhin

Exercise 4: Deploying a Containerized Application on GKE

Question 1: How did you create and push the Docker container to GCR?

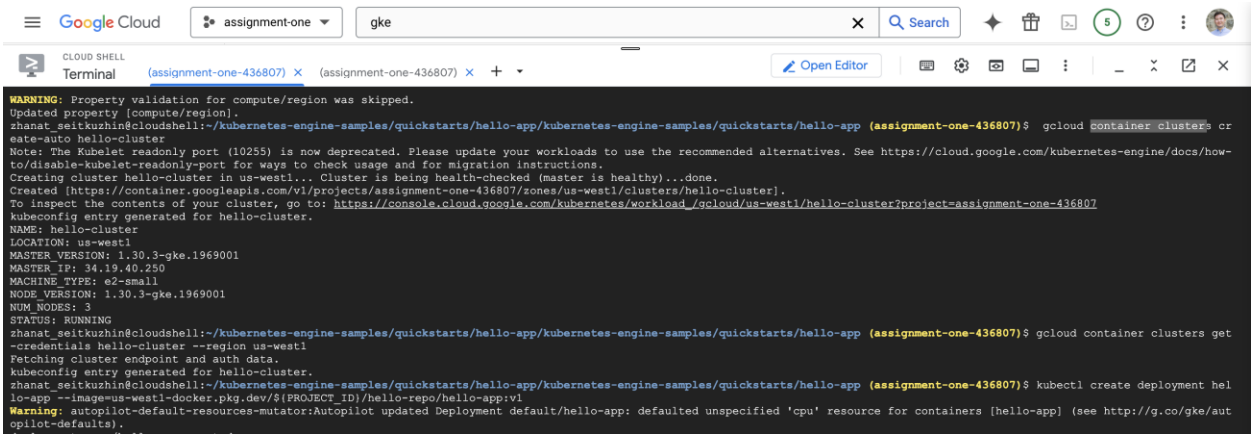
I created the Docker container using a Dockerfile. After building the Docker image, I pushed it to Google Container Registry (GCR) with the following commands:



Question 2: What steps were involved in setting up the GKE cluster?

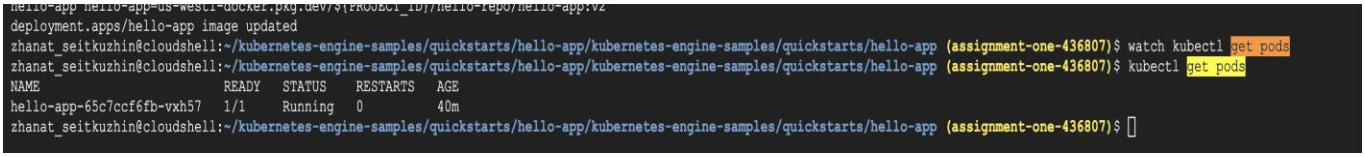
I stored docker image in Artifact Registry, create a GKE cluster to run my app. I set up the GKE cluster by executing the following command in the Google Cloud Console:

Zhanat Seitkuzhin

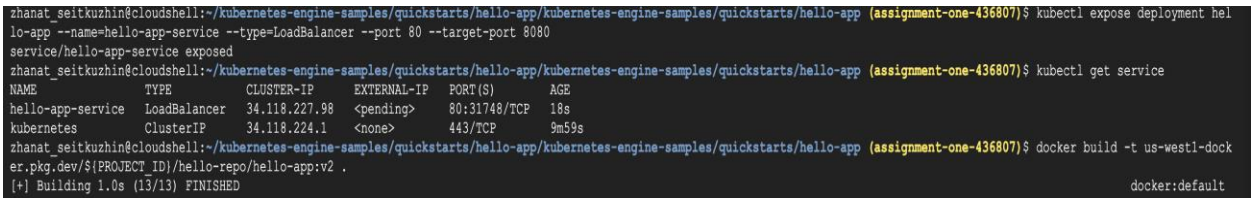


Question 3: How did you verify that your application was successfully deployed and accessible?

1. I checked the pods status: after deploying your application using the kubectl command, you can verify the status of the Pods with the command (ensuring that the status is “Running”):



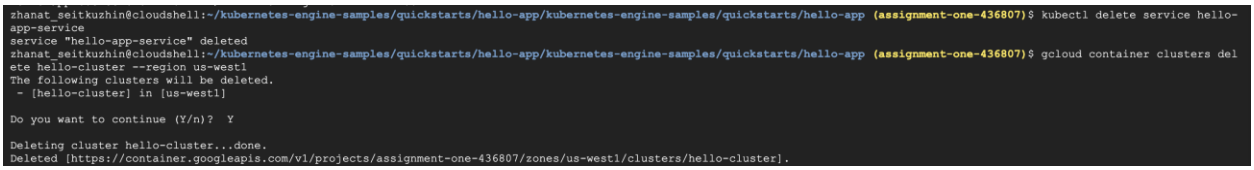
2. I exposed the application for external access by creating a LoadBalancer Service, grouping my Pods under a single IP and ensuring that other users can access my application over the internet.
3. I retrieved external IP address so other users can use it to reach my application, ensuring that LoadBalancer is ready to route traffic to my Pods.



4. I tested application accessibility using external IP in a new browser tab, confirming that my Pods are running correctly, the Service is properly routing traffic, and the application is functioning as expected.:

Zhanat Seitkuzhin

5. Clean Up:

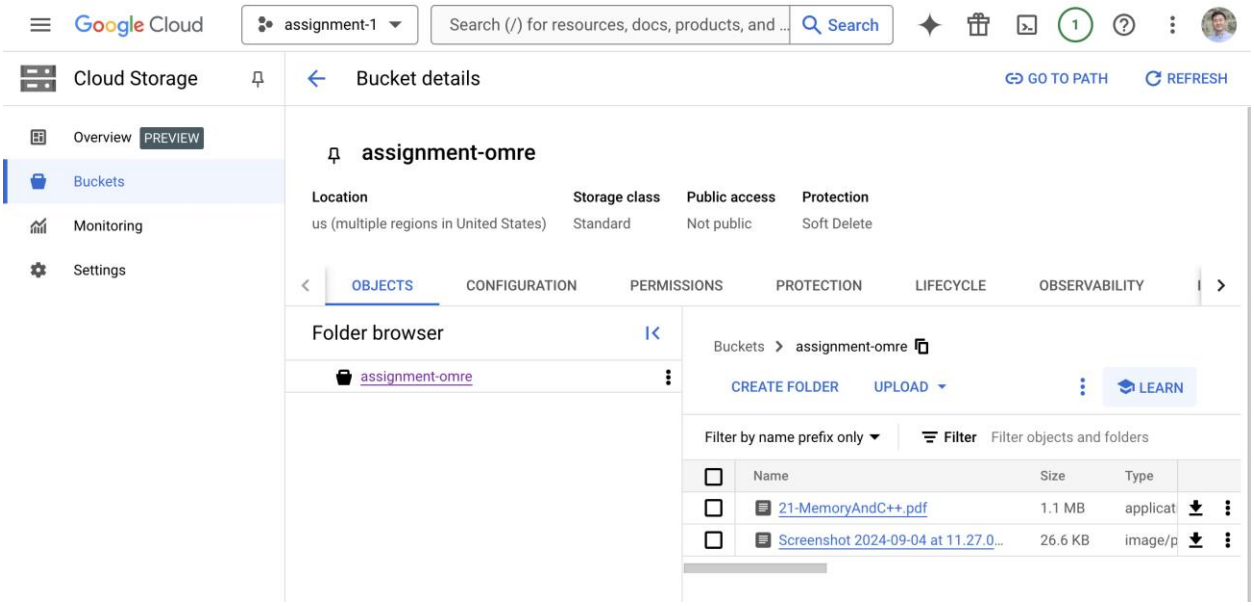


Zhanat Seitkuzhin

Exercise 5: Storing and Accessing Data in Google Cloud Storage

Question 1: How do you create a Cloud Storage bucket, and what options are available during setup?

I created a Cloud Storage bucket via the GCC by navigating to the Storage section and clicking "Create bucket." There are various options to choose from: setting a unique bucket name, choosing a default storage class (Standard, Nearline, Coldline, Archive) based on how frequently I would use this bucket, and selecting a location type (multi-region, dual-region, or single-region). Then I clicked “Create.”



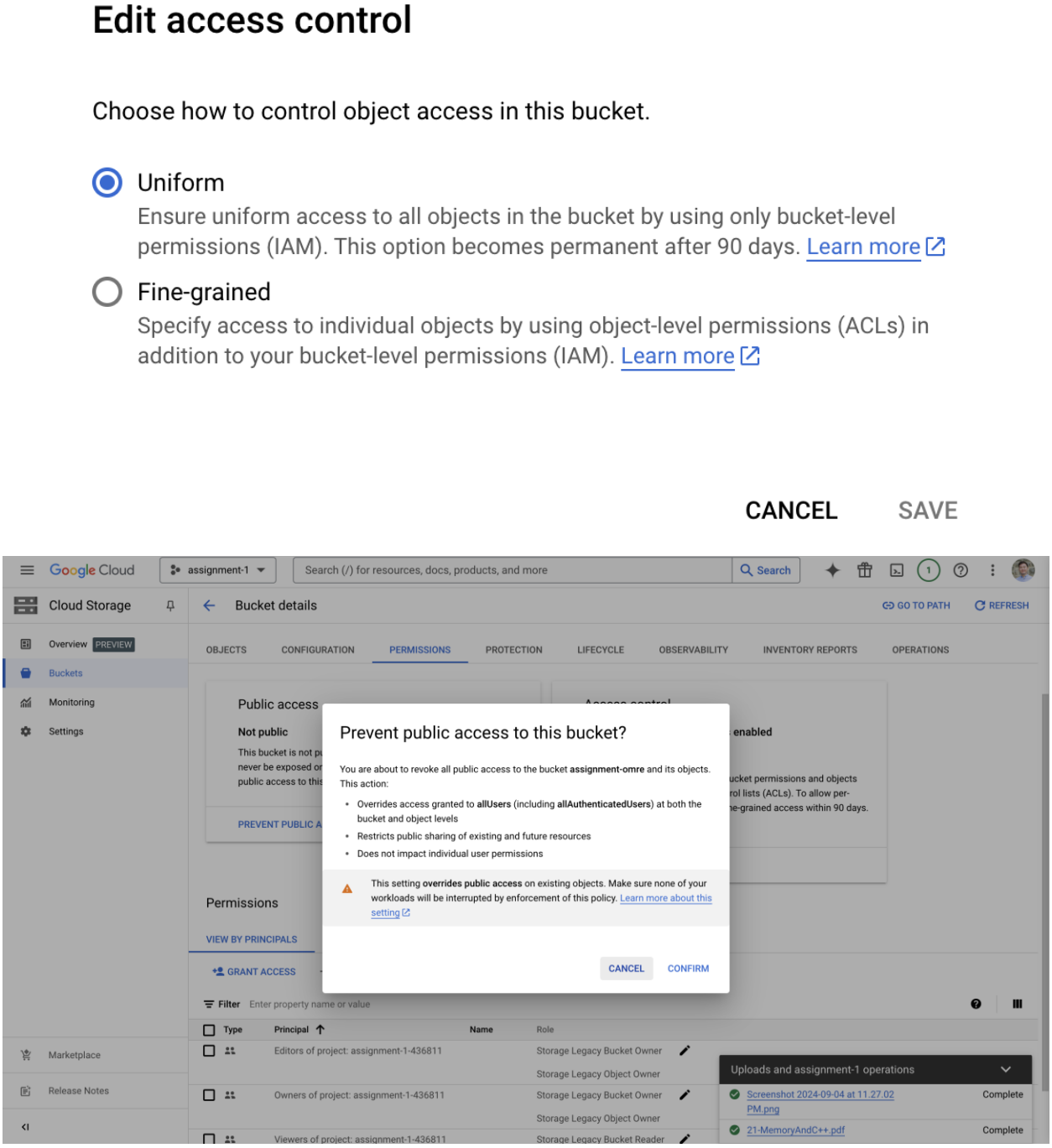
Question 2: What are the differences between setting a bucket to public versus private?

Public bucket allows anyone on the internet to access its files without authentication, and is usually for shared resources.

Private bucket restricts access to only those who have been explicitly granted permissions to access sensitive data.

Question 3: How can you manage access permissions for individual files in a bucket?

I can manage access permissions for individual files by selecting the file in the Cloud Storage Console, navigating to the “Permissions tab”, and adding specific users or groups and selecting their roles. Alternatively, I could also set ACLs (Access Control Lists) for finer control over access. For example:



Exercise 6: Analyzing Data with BigQuery

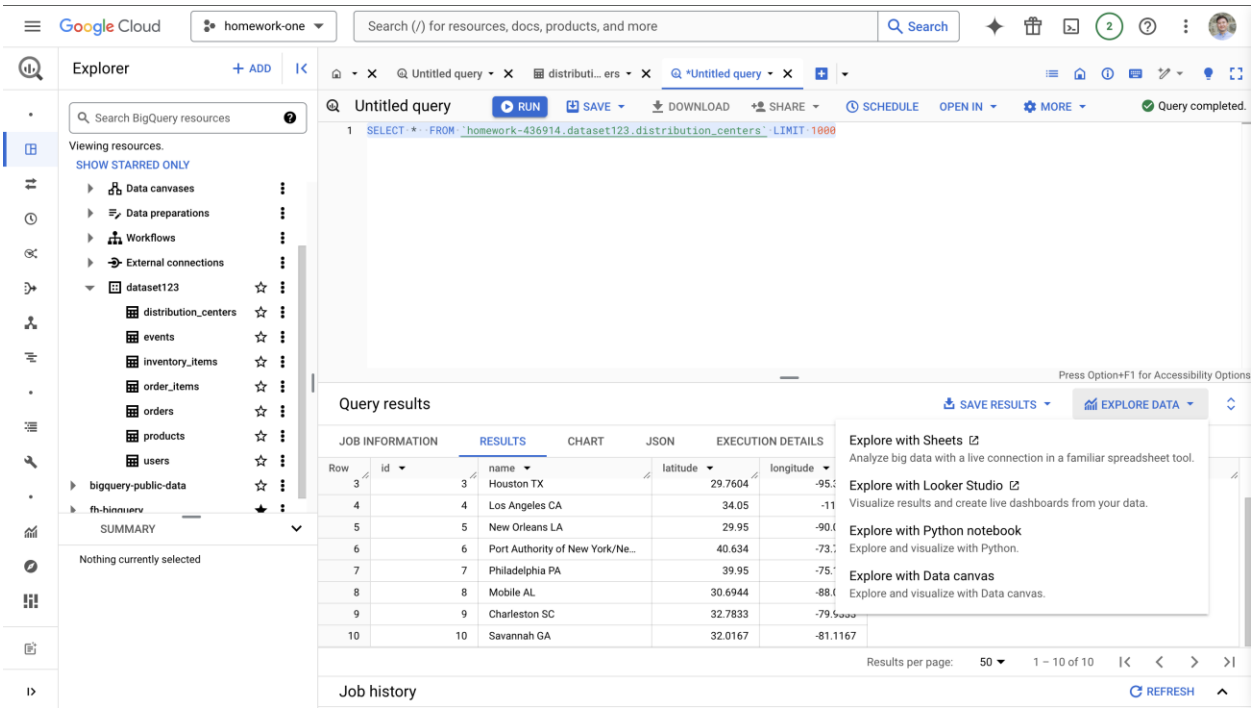
Zhanat Seitkuzhin

Question 1: What steps did you take to create a dataset and table in BigQuery?

1. I accessed BigQuery from Google Cloud Console.
2. Clicked “Create Dataset,” providing a name and using default configurations.
3. I found a public dataset at Google Marketplace. I chose “theLook eCommerce.”
4. I copied it to my existing dataset.

Question 2: How did you write and execute SQL queries in BigQuery?

I used the SQL editor in the BigQuery interface and run a simple SQL query to test it.



Zhanat Seitkuzhin

Question 3: What insights were you able to derive from the data analysis?

I visualized the data using Locker Studio, and besides a nice-looking table and a blue colored column chart I have no insights since the data only has information about data center's locations with longitude and latitude.