**Assignment 1, Cloud Computing**

**Exercise 1.**

1. Differences between services and examples of usage are in Table 1.

|  | Infrastructure as a Service (IaaS) | Platform as a Service (PaaS) | Software as a Service (SaaS) |
| --- | --- | --- | --- |
| Control | High | Medium | Low |
| Flexibility | High | Medium | Low |
| Use cases | * Provides virtualized computing resources (e.g., servers, storage, networks) * Users have full control over the infrastructure and can configure it to meet their specific needs * Users are responsible for managing and maintaining the infrastructure * Examples: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP) | * Provides a complete development and deployment environment for applications * Users have control over the application and data, but not the underlying infrastructure * Users do not need to manage or maintain the infrastructure * Examples: Heroku, Google App Engine, Microsoft Azure App Service | * Provides software applications over the internet * Users access the software applications through a web browser or mobile app * Users do not have control over the underlying infrastructure or application code * Examples: Salesforce, Microsoft Office 365, Google Workspace |
| GCP services | Compute Engine, Google Kubernetes Engine (GKE), Cloud, Storage Cloud Load Balancing | App Engine, Cloud Functions, Cloud Run, Cloud Dataflow | Google Workspace (Gmail, Google Drive, and Google Docs), Google Cloud Search, Google Cloud Identity and Access Management (IAM), Google Cloud Security Command Center |

Table 1: Differences between IaaS, PaaS and SaaS

Examples of using each service:

* **IaaS:** Example: A large e-commerce company, for example, needs to scale its website to handle a sudden surge in traffic during a holiday sale. They need to provision additional servers to handle the increased traffic quickly. With IaaS, they can spin up new virtual machines (VMs) in minutes, configure them as needed, and scale back down when the traffic subsides. This allows them to quickly adapt to changing demands without worrying about the underlying infrastructure.
* **PaaS:** Example: A startup that wants to build a mobile app that allows users to order food from local restaurants. They need a platform to develop, test, and deploy their app, but they don't want to worry about managing the underlying infrastructure. With PaaS, they can use a managed platform like Google App Engine to build and deploy their app, without worrying about server provisioning, scaling, or maintenance.
* **SaaS:** Example: A small business must manage its social media presence across multiple platforms. They want to use a social media management tool to schedule posts, track engagement, and analyze performance. With SaaS, they can use a cloud-based social media management tool like Hootsuite or Sprout Social, which provides a user-friendly interface and scalable infrastructure to manage their social media presence. They don't need to worry about installing, configuring, or maintaining the software on their own devices.

**Exercise 2. Exploring Google Cloud Platform’s Core Services**

1. **What is the primary use case of Compute Engine?** The primary use case of Google Compute Engine (GCE) is to provide a scalable and flexible infrastructure for running virtual machines (VMs) and containers. GCE is designed to provide a managed, cloud-based infrastructure for running a wide range of workloads.
2. **How does Google Kubernetes Engine (GKE) simplify the management of containerized applications?** GKE automates the deployment of containerized applications, allowing you to manage the lifecycle of containers, including scaling, rolling updates, and self-healing. It’s easier to control access. enabling security with regulatory requirements. scale the containers up or down to match changing workload demands, and integrate with other Google Services. as it’s integrated with them(such as Google Cloud, and Cloud SQL and Cloud Pub/Sub).
3. **What advantages does Cloud Storage offer for data management?** Cloud storage offers several advantages for data management:
   * **Scalability and flexibility**: it allows users to store and access large amounts of data from anywhere, at any time.
   * **Backups and disaster recovery**: ensuring that data is safe and secure in the event of hardware failure or natural disasters.
   * **Collaboration and sharing** of files and data across teams and organizations, improving productivity and reducing the risk of data loss or corruption.
   * **Security:** encryption and access controls, to protect sensitive data.
4. **Why would a business choose BigQuery for their data analysis needs?** Scalability, flexibility, and cost-effectiveness. It allows for fast and efficient querying of large datasets, and its integration with other Google Cloud services enables seamless data processing and analysis. Additionally, BigQuery's serverless architecture eliminates the need for infrastructure management, reducing administrative burdens and allowing data scientists to focus on analysis rather than infrastructure.
5. **Google App Engine** is a fully managed platform for building scalable web applications and mobile backends. It provides a reliable and secure environment for applications to run, automatically scaling to handle changes in traffic and providing high uptime and reliability. App Engine supports a wide range of programming languages, including Python, Java, PHP, and Go, and integrates with other Google services.

**Exercise 3: Creating and Managing Virtual Machines with Compute Engine**

1. Creating the VM in Figure 1

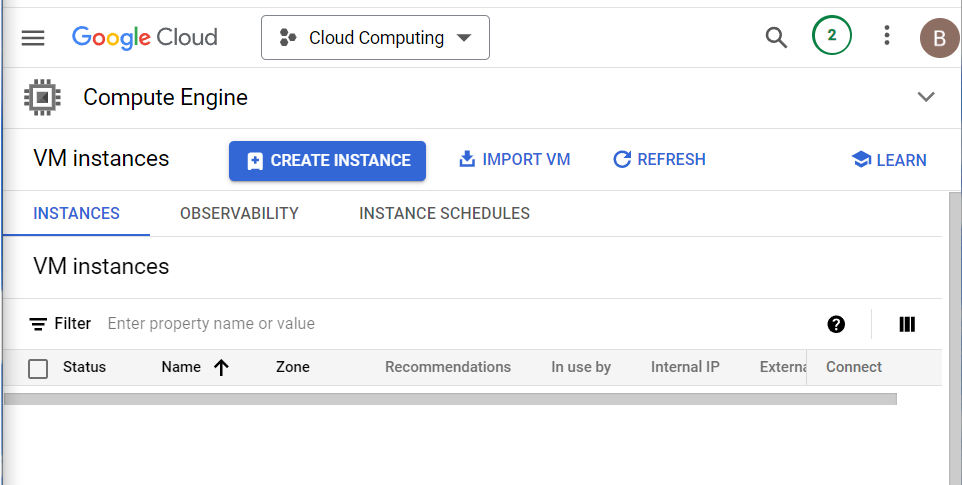
****

Figure 1: Creating VM instance

1. When pressing the “Create instance” button, we configured properties of the VM instance in Figure 2. For example, here I set the VM provisioning model to “Spot”, because I do not need all time availability. As well as set a time limit for the VM. Region was us-east1(South Carolina) but has changed since this region wasn’t available at the moment.

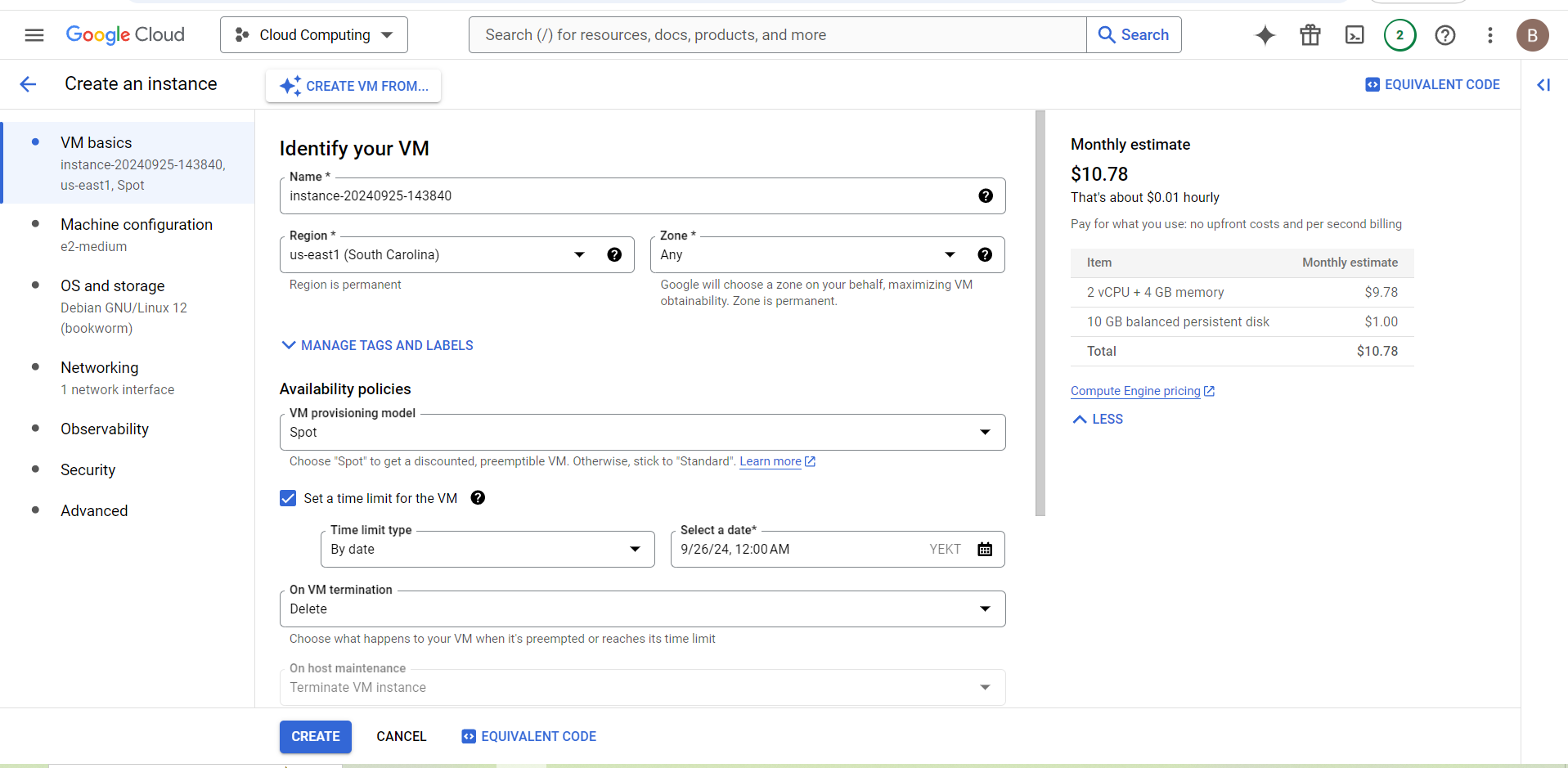
****

Figure 2: Configuring properties of the VM

1. After changing the region to us-central1 successfully have created new VM instance in Figure 3

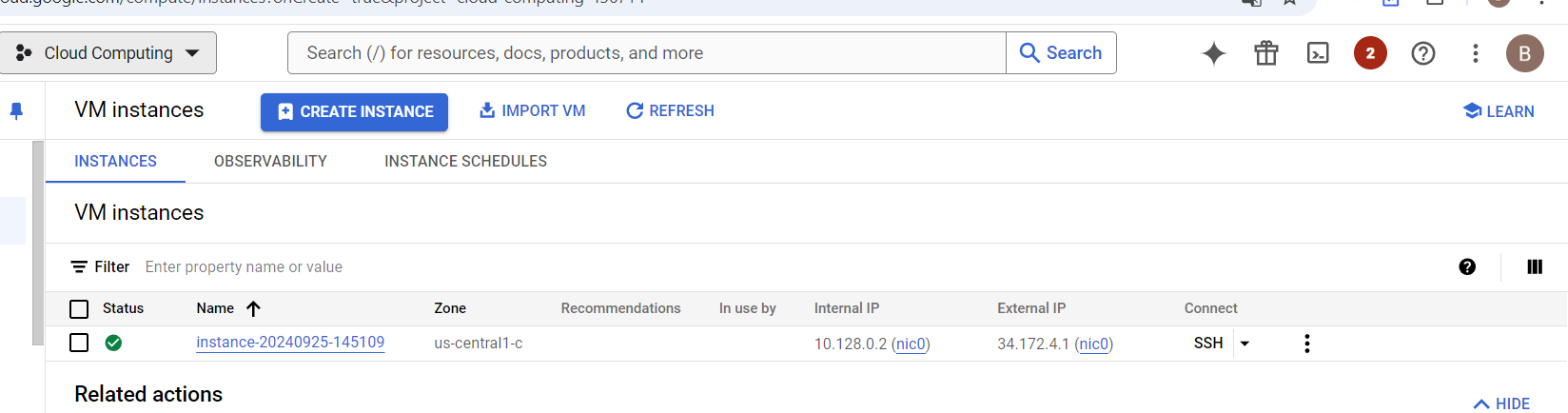
****

Figure 3: Successful creation of the VM

1. Accessing VM using SSH. I also generated ssh keys for my account in Figure 4

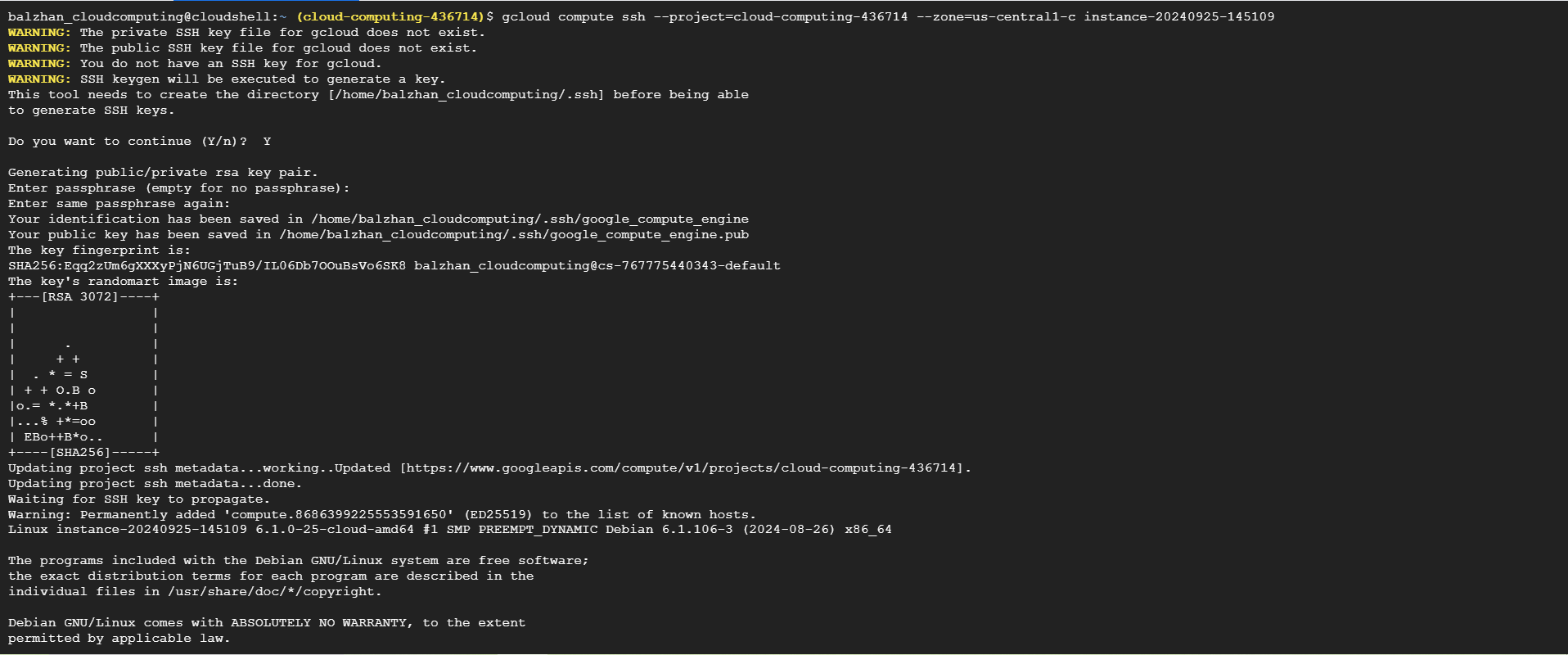
****

Figure 4: Generating ssh keys for the google account

1. First we should update the package list in Figure 5

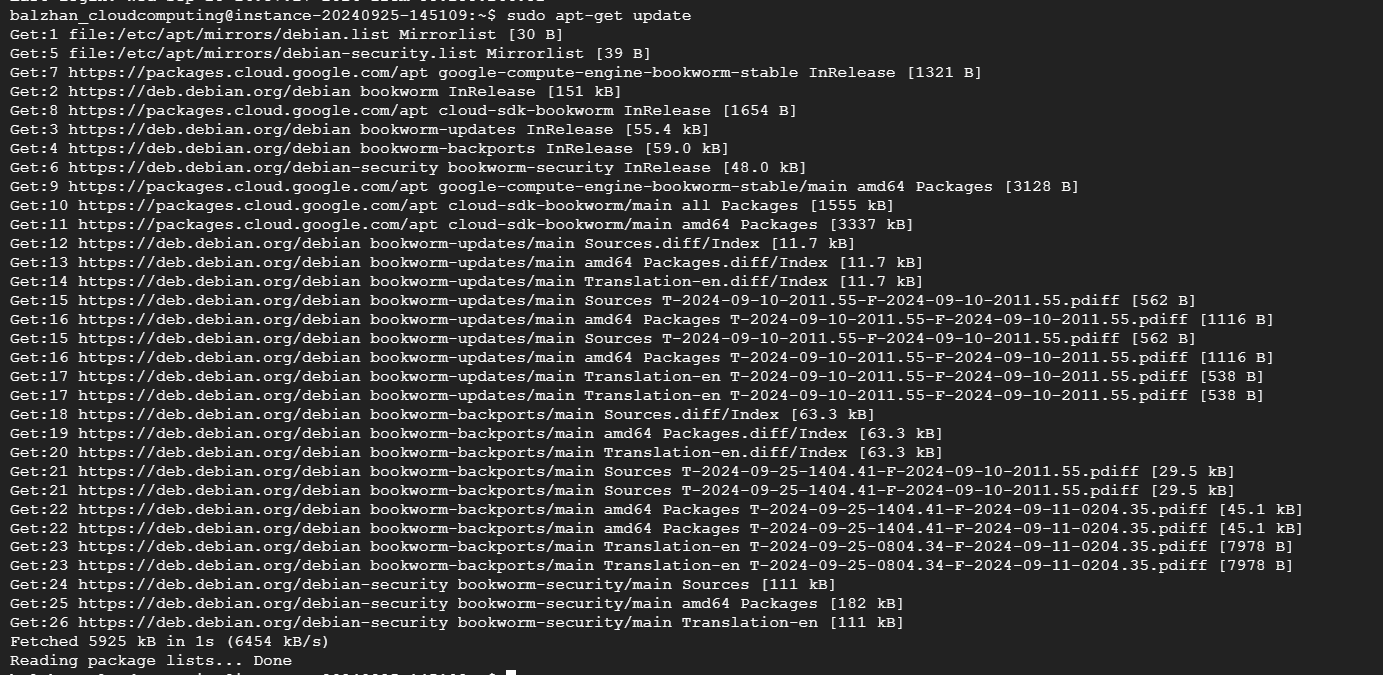
****

Figure 5: Updating the package list

1. Installing Apache web server in Figure 6

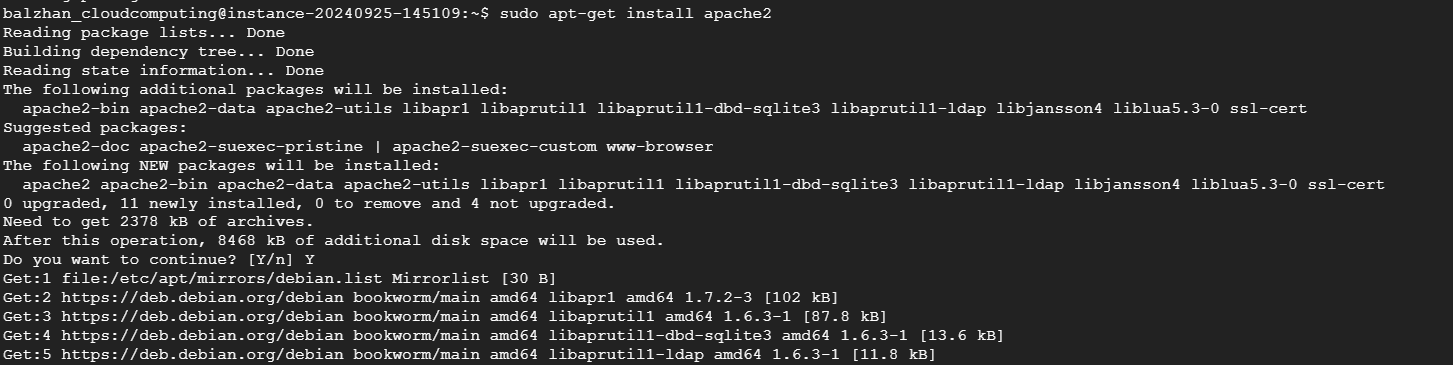


Figure 6: Installing Apache web server

1. Starting Apache server in Figure 7



Figure 7: Starting Apache server

1. Enable the Apache service to start automatically on boot in Figure 8

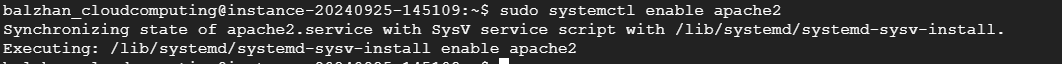


Figure 8: Enabling Apache server

1. Setting firewall policy to allow access only for my ip for this service in Figure 9 and Figure 10

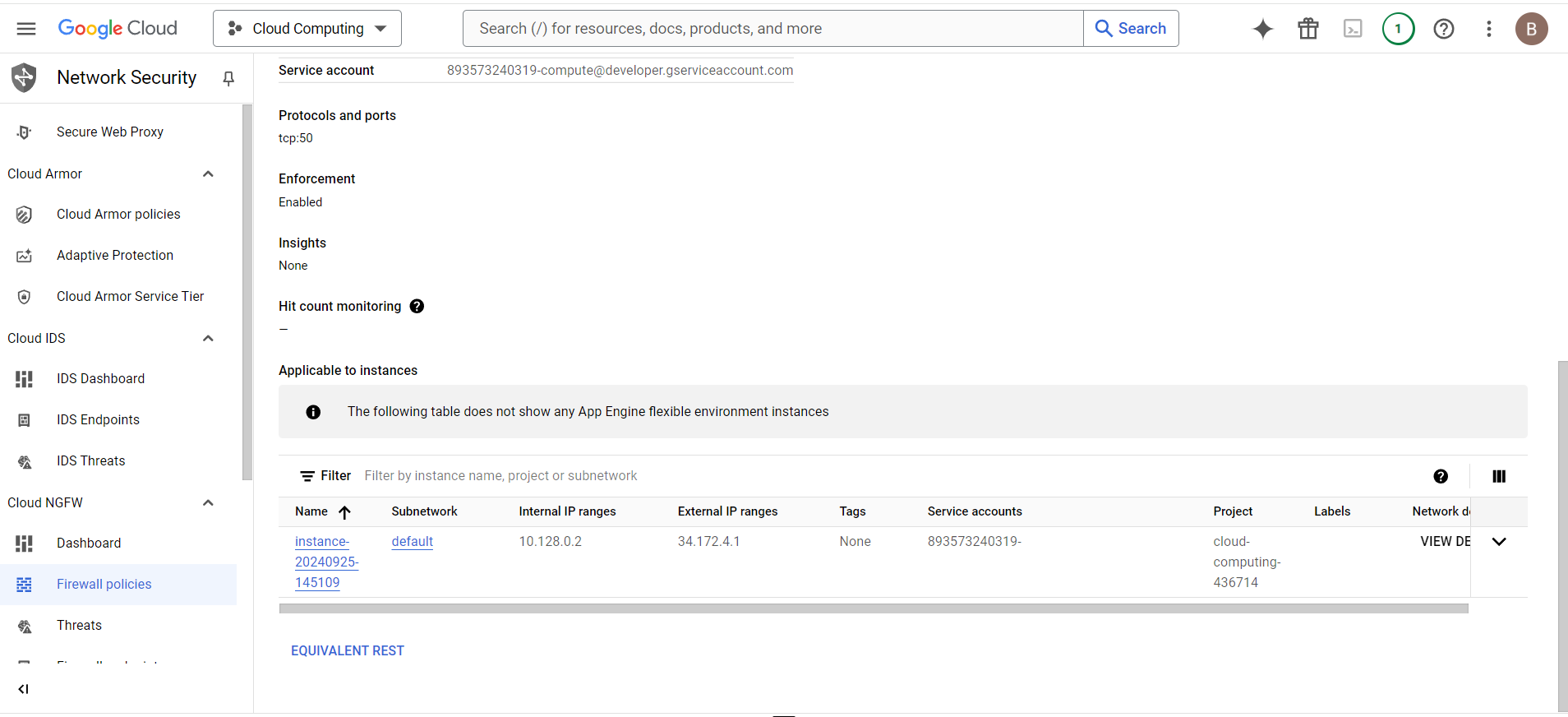


Figure 9: Setting firewall policy

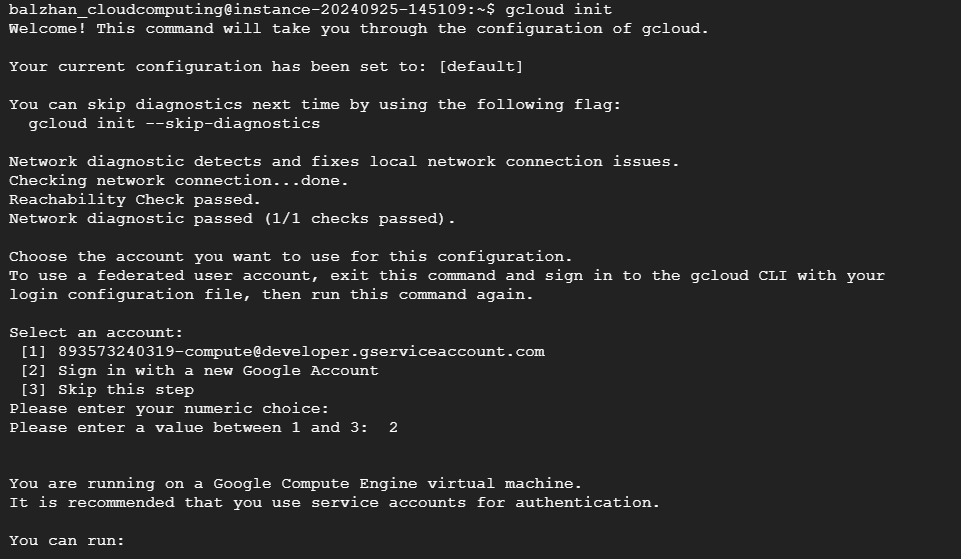


Figure 10: Setting configuration

1. Stopping the VM instance in Figure 11

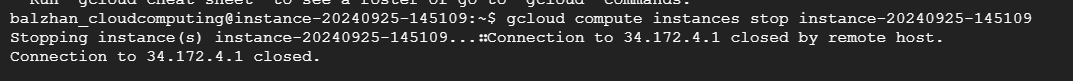


Figure 11: Stopping VM instance

The status is shown to be stopped in Figure 12

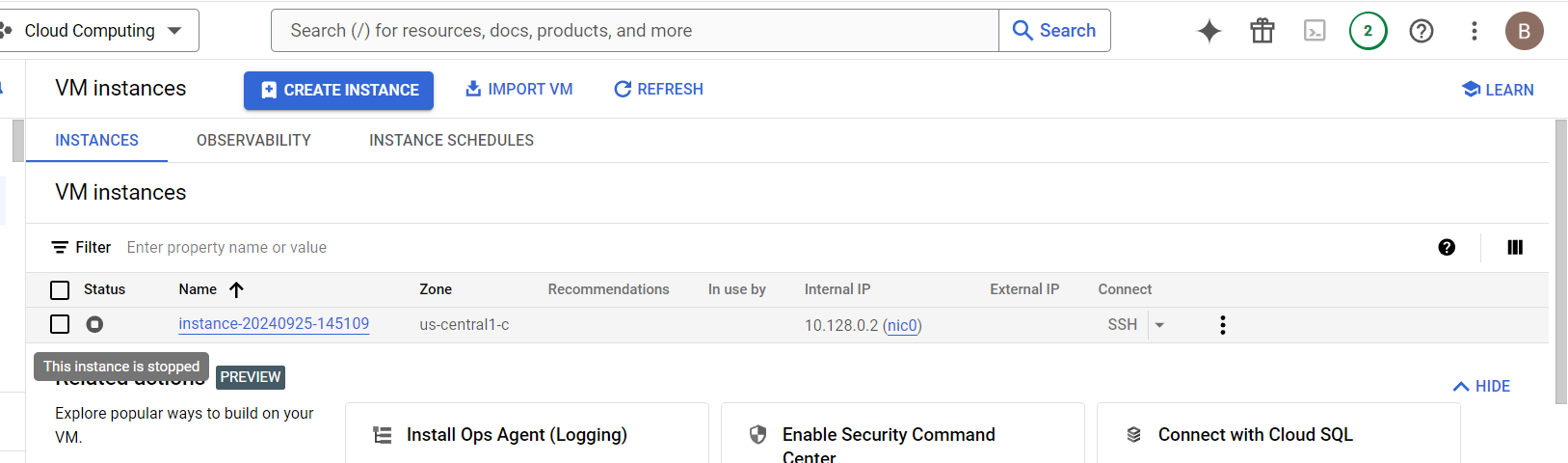


Figure 12: Status of the VM instance (stopped)

1. Restarting VM instance in Figure 13

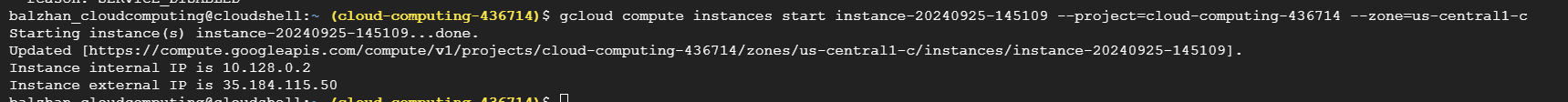


Figure 13: Restarting VM instance

Status is running now as shown in Figure 14.

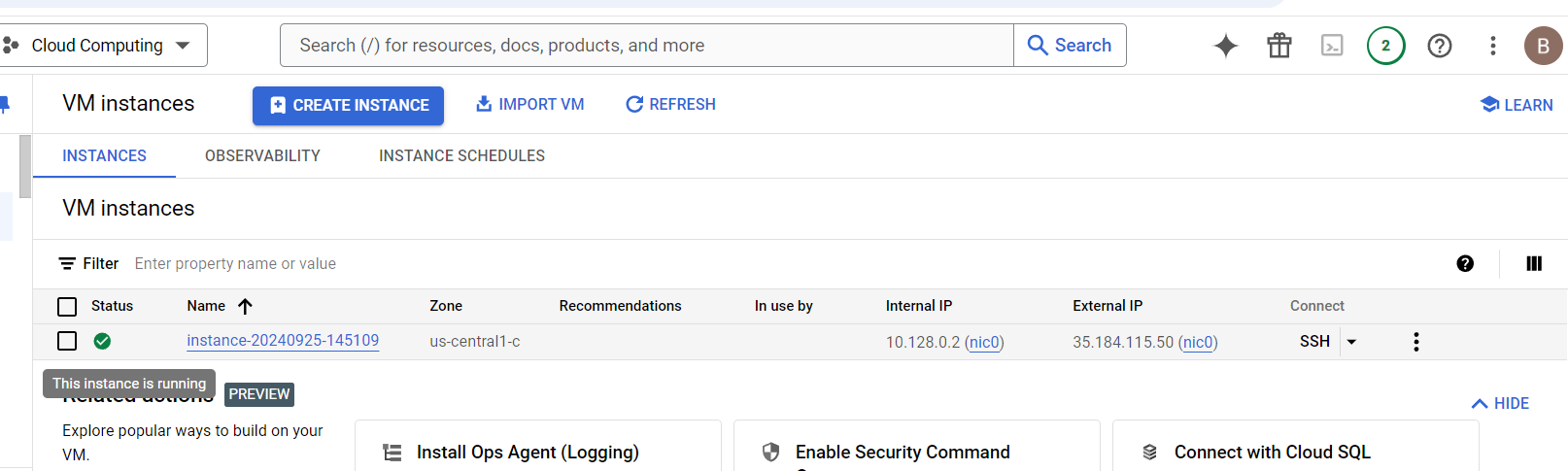


Figure 14: Status of the VM instance (running)

1. Clean up and deletion in Figure 15

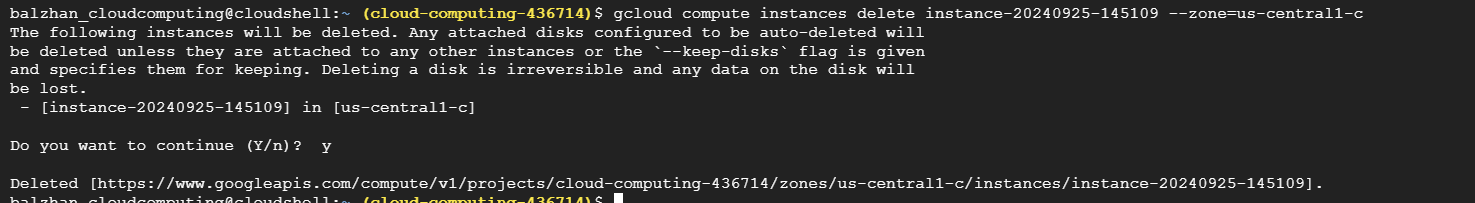


Figure 15: Deleting VM instance

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

In Google Cloud, when you stop a VM (Virtual Machine), it is not deleted, but rather it is shut down and its resources are deallocated. Here's what happens:

* The VM is shut down, and its processes are terminated.
* The VM's disk is not deleted, and its data remains intact.
* The VM's IP address is released, and it is no longer accessible.
* The VM's resources, such as CPU, memory, and storage, are deallocated, and you are no longer charged for them.

On the other hand, when you delete a VM, it is permanently removed from your Google Cloud account, and its data is lost. Here's what happens:

* The VM is deleted, and its disk is also deleted.
* The VM's data is lost, and it cannot be recovered.
* The VM's IP address is released, and it is no longer accessible.
* The VM's resources, such as CPU, memory, and storage, are deallocated, and you are no longer charged for them.

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

1. Setting up properties. The command is successful as it returned Updated property in Figure 16

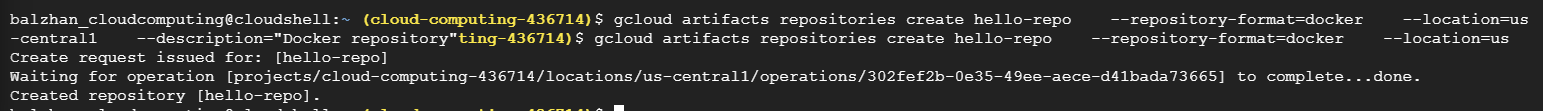


Figure 16: Updating property

1. Creating container in Figure 17

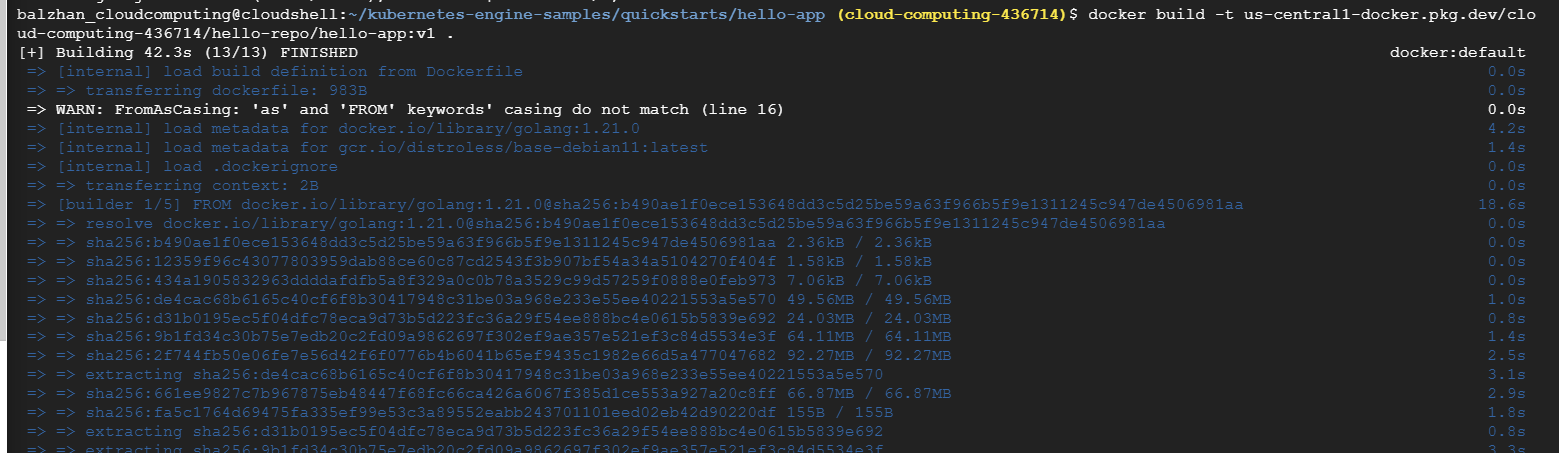


Figure 17: Creating container

1. Created images from the previous program in Fugure 18

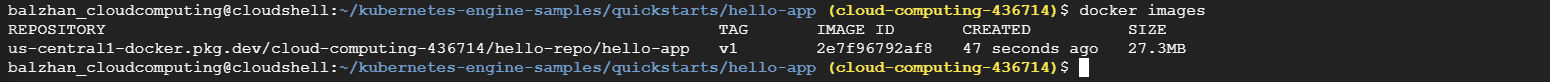


Figure 18: Creating image

1. The docker run command running a Docker container from the hello-app:v1 image in the hello-repo repository in the cloud-computing-436714 project on Google Cloud in Figure 19

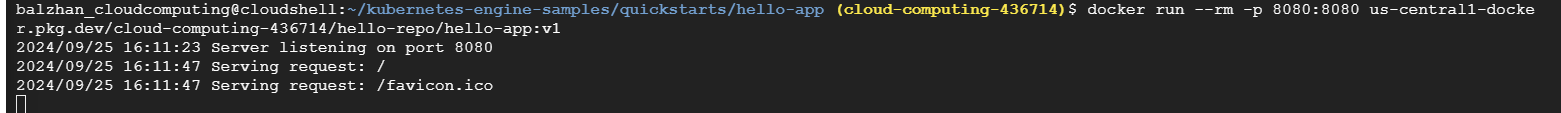


Figure 19: Docker run command

1. Viewing deployed app in the web in Figure 20

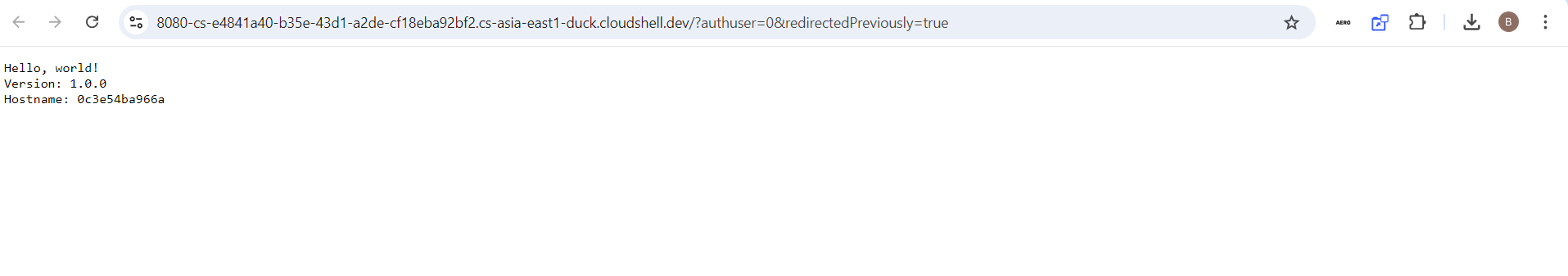


Figure 20: Deployed app

1. Creating cluster in Figure 21

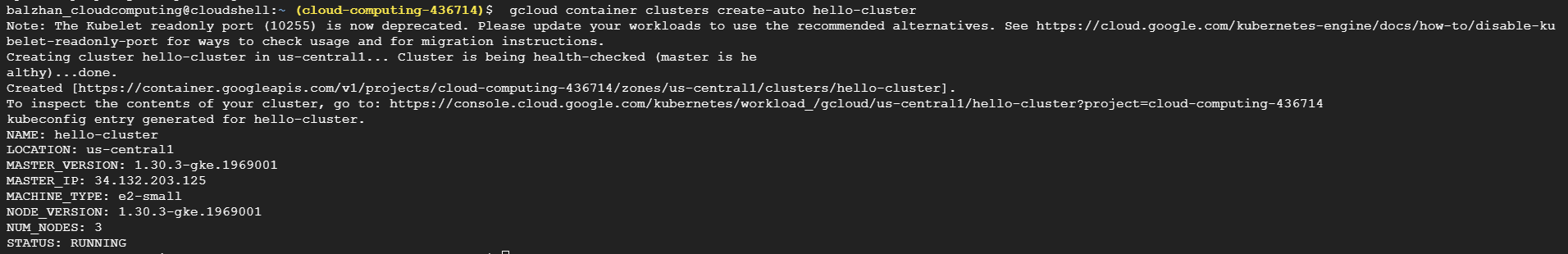


Figure 21: Creating cluster

1. Authorizing command to create the cluster in Figure 22 and Figure 23

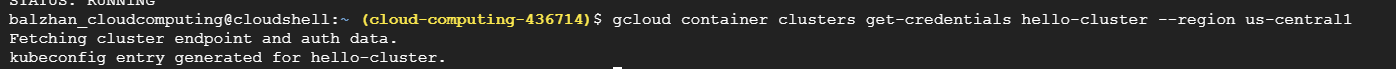


Figure 22: Container credentials

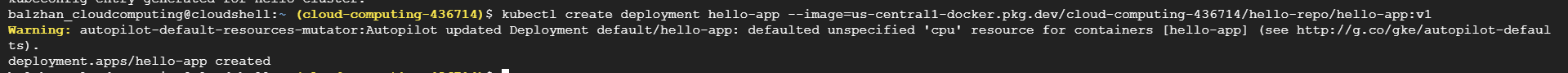


Figure 23: Deployment

8. Scaling the deployment ton 3 replicas of the pods running in Figure 24 and Figure 25



Figure 24: Scaling the pods

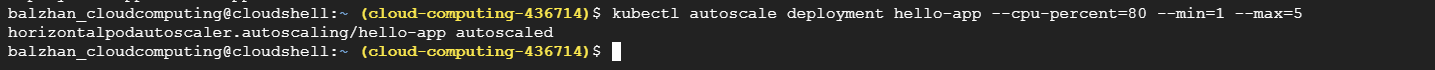


Figure 25: Autoscaling

9. Checking the pods running on cluster in Figure 26

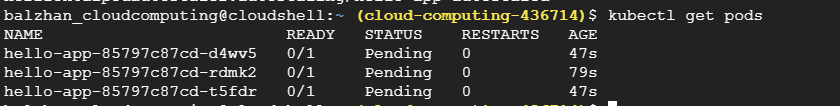


Figure 26: Running pods on cluster

10. Clean up in Figure 27

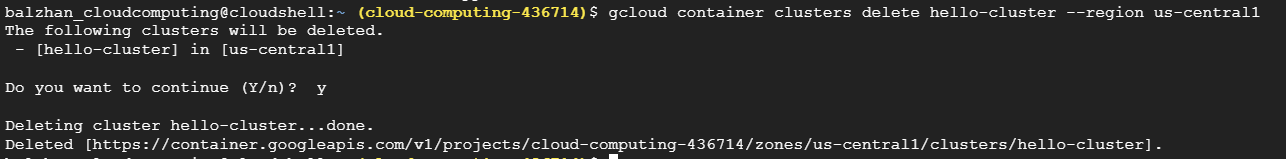


Figure 27: Clean up of containers

**Exercise 5: Storing and Accessing Data in Google Cloud Storage**

1. Creating bucket with name balzhan in Figure 28

****

Figure 28: Creating bucket

1. Verifying the successful creation of the bucket in Figure 29

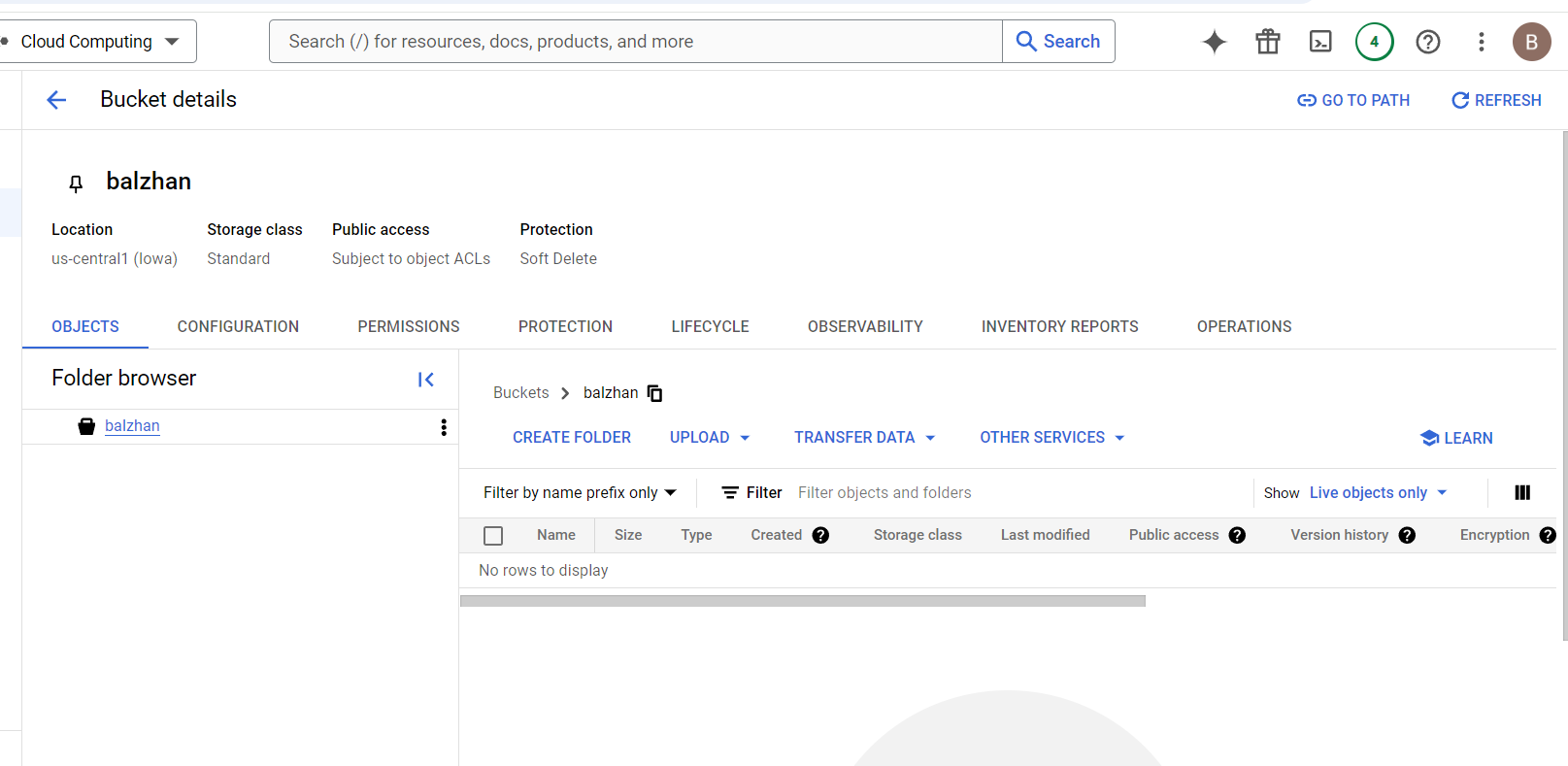
****

Figure 29: Successful creation of the bucket

1. Uploading files in Figure 30

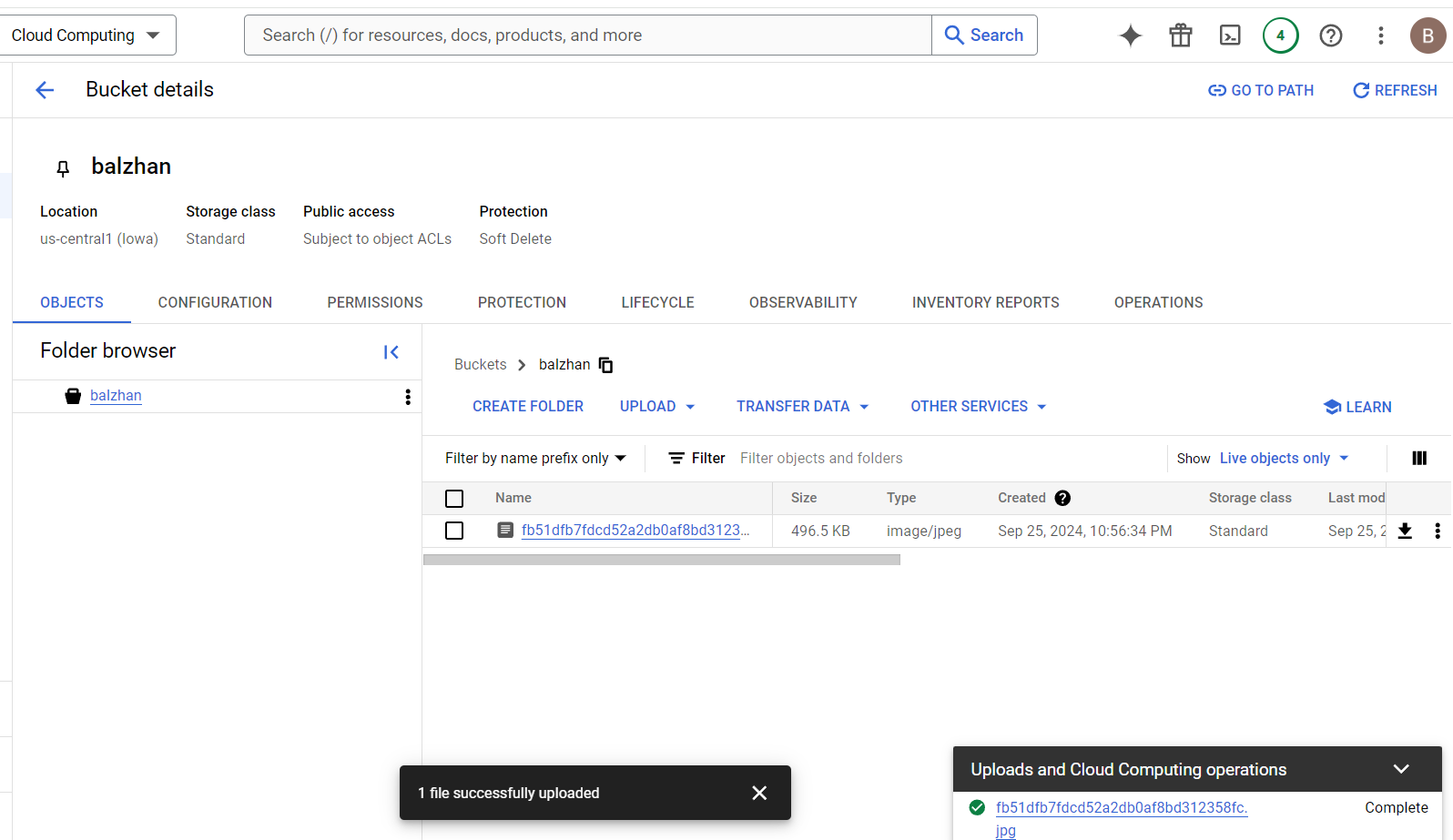
****

Figure 30: Uploading file to the bucket

1. Setting up access permissions for the file uploaded in Figure 31

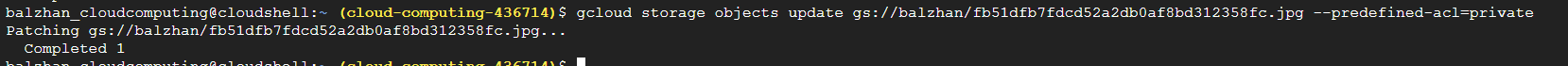
****

Figure 31: Setting up access permissions to the file

1. Setting up access permissions to whole bucket in Figure 32

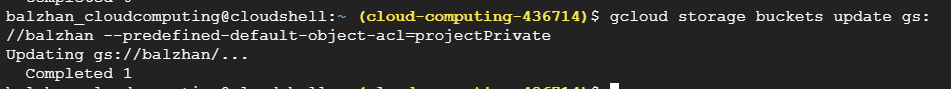
****

Figure 32: Setting up access permissions to the bucket

1. Verifying access permissions of the bucket in Figure 33 and Figure 34

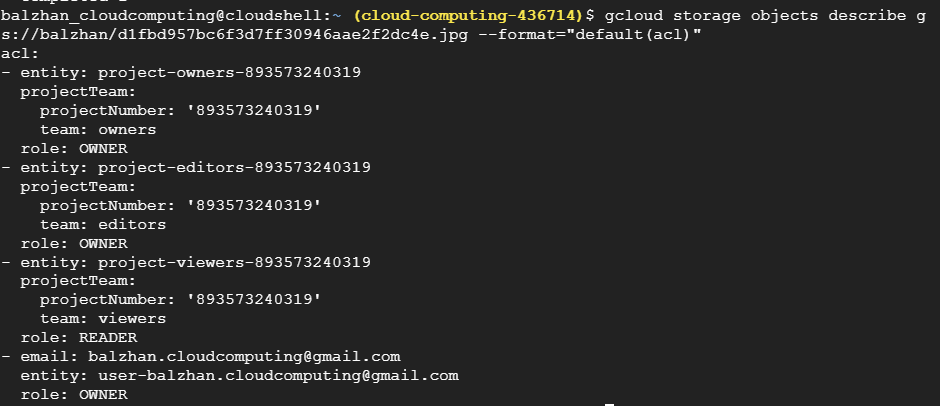
****

Figure 33: Permissions of the bucket

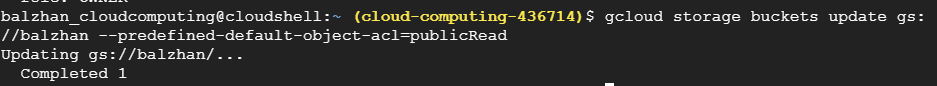
****

Figure 34: Setting public access to read

1. Renaming and copying uploaded file in Figure 35

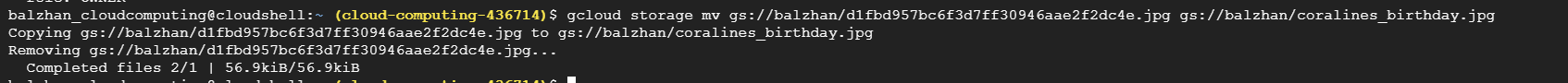
****

Figure 35: Renaming and copying uploaded file

1. Deleting the file in Figure 36

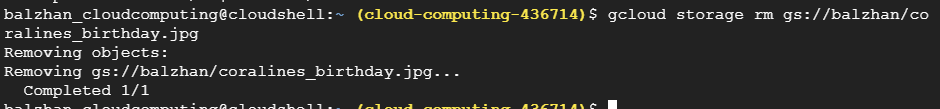
****

Figure 36: Delete file

1. Clean up in Figure 37

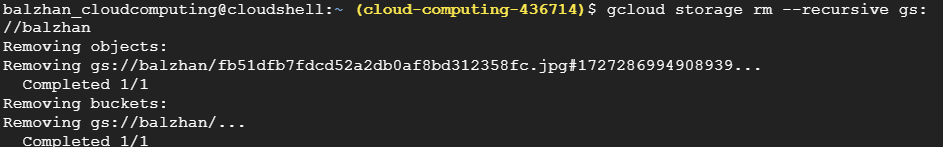
****

Figure 37: Cleaning up the bucket

1. What are the differences between setting a bucket to public versus private?

Public Bucket:

* Anyone can read your files and folders
* No login required
* Good for sharing public data or hosting a website
* But, it's like leaving your front door open - anyone can walk in!

Private Bucket:

* Only authorized users can access your files and folders
* You need to log in to get in
* Perfect for storing sensitive data or controlling who can upload files
* It's like locking your front door - only those with the right key can get in!

**Exercise 6: Analyzing Data with BigQuery**

1. Creating dataset in Figure 38

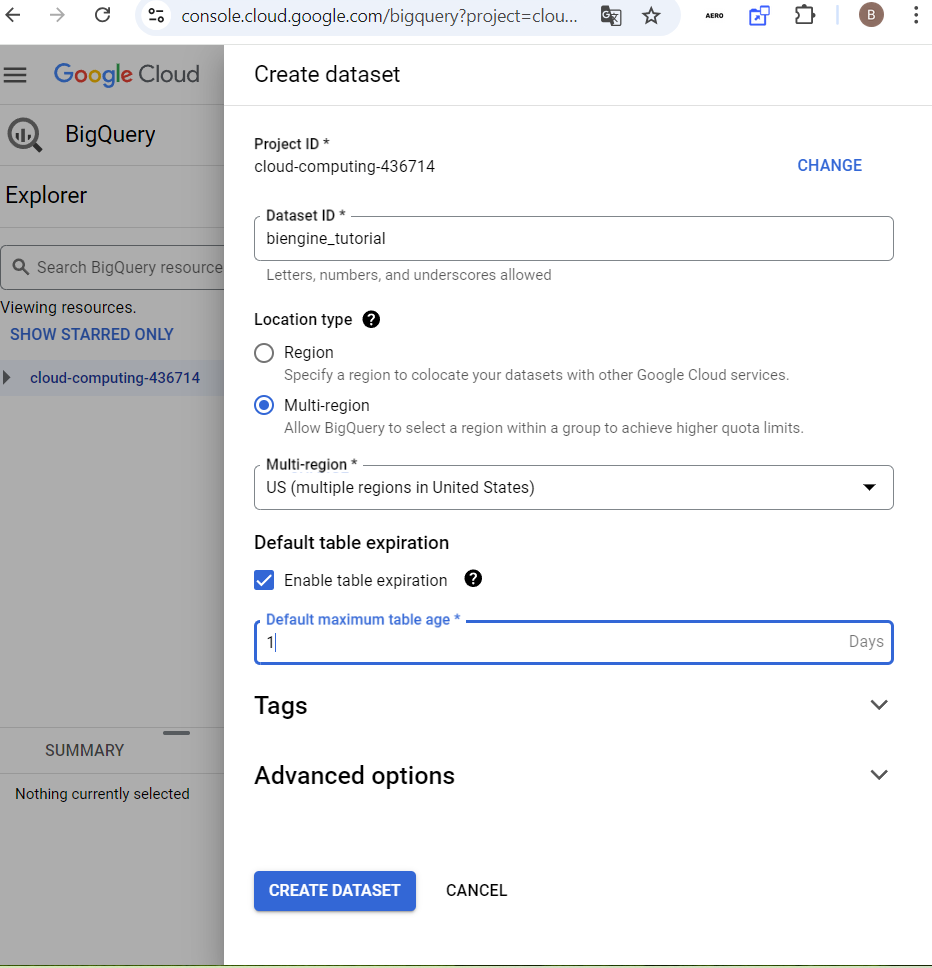


Figure 38: Creating dataset

1. Copying dataset from the google public data with table san\_francisco\_311 in Figure 39

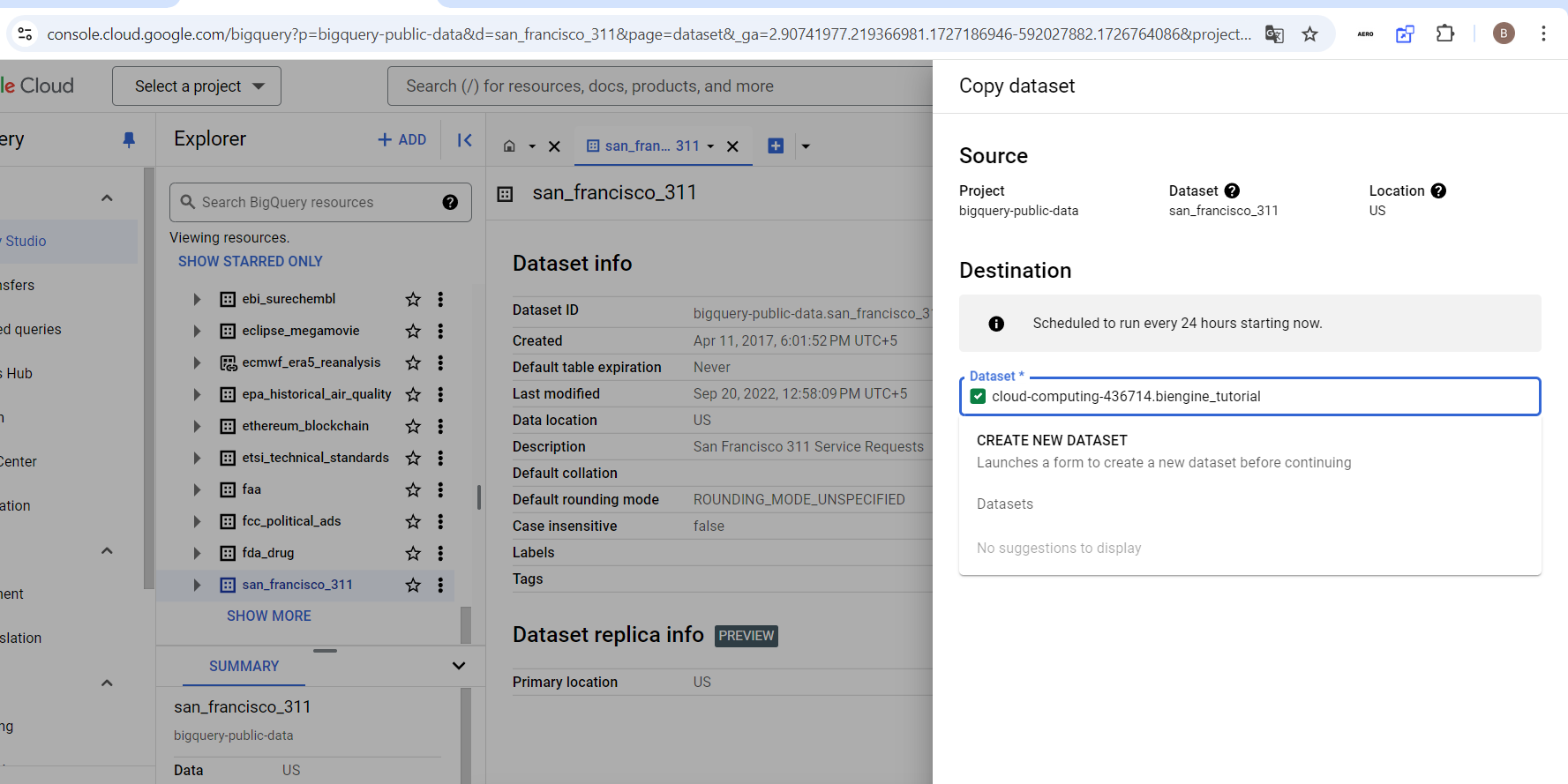


Figure 39: Copying data from the table to the dataset

1. In the dataset the copied table appeared in Figure 40

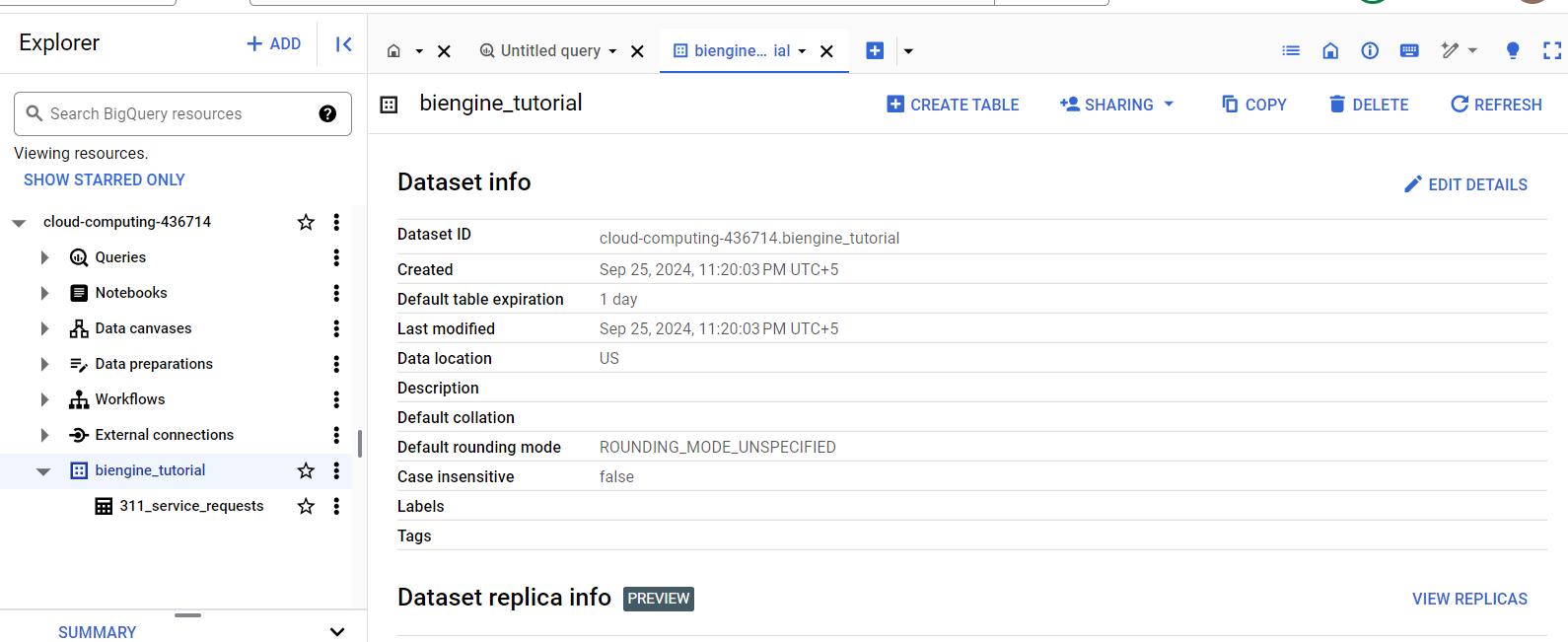


Figure 40: Verification of the copied data

1. Running SQL commands in Figure 41 and Figure 42

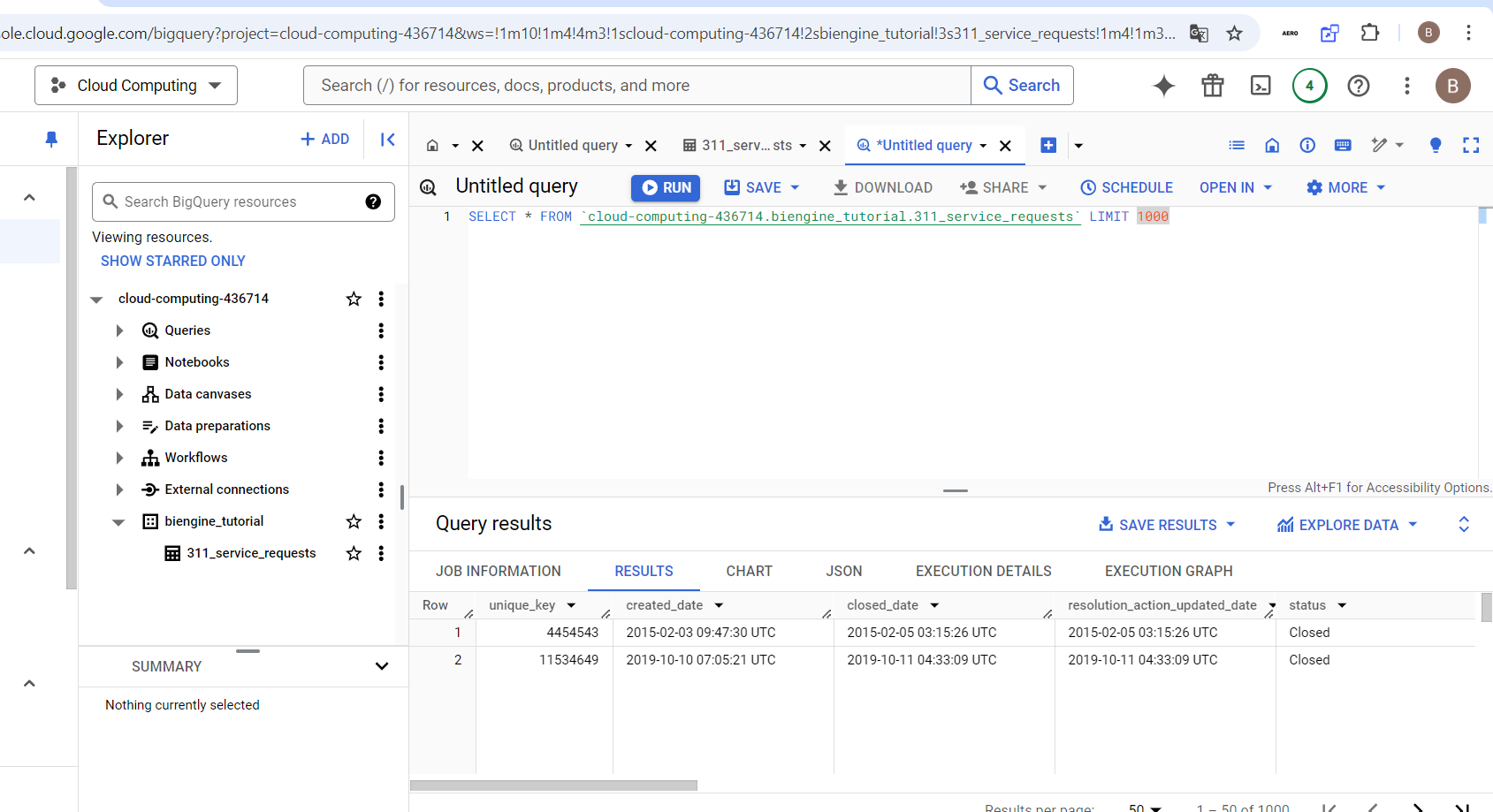


Figure 42: Query

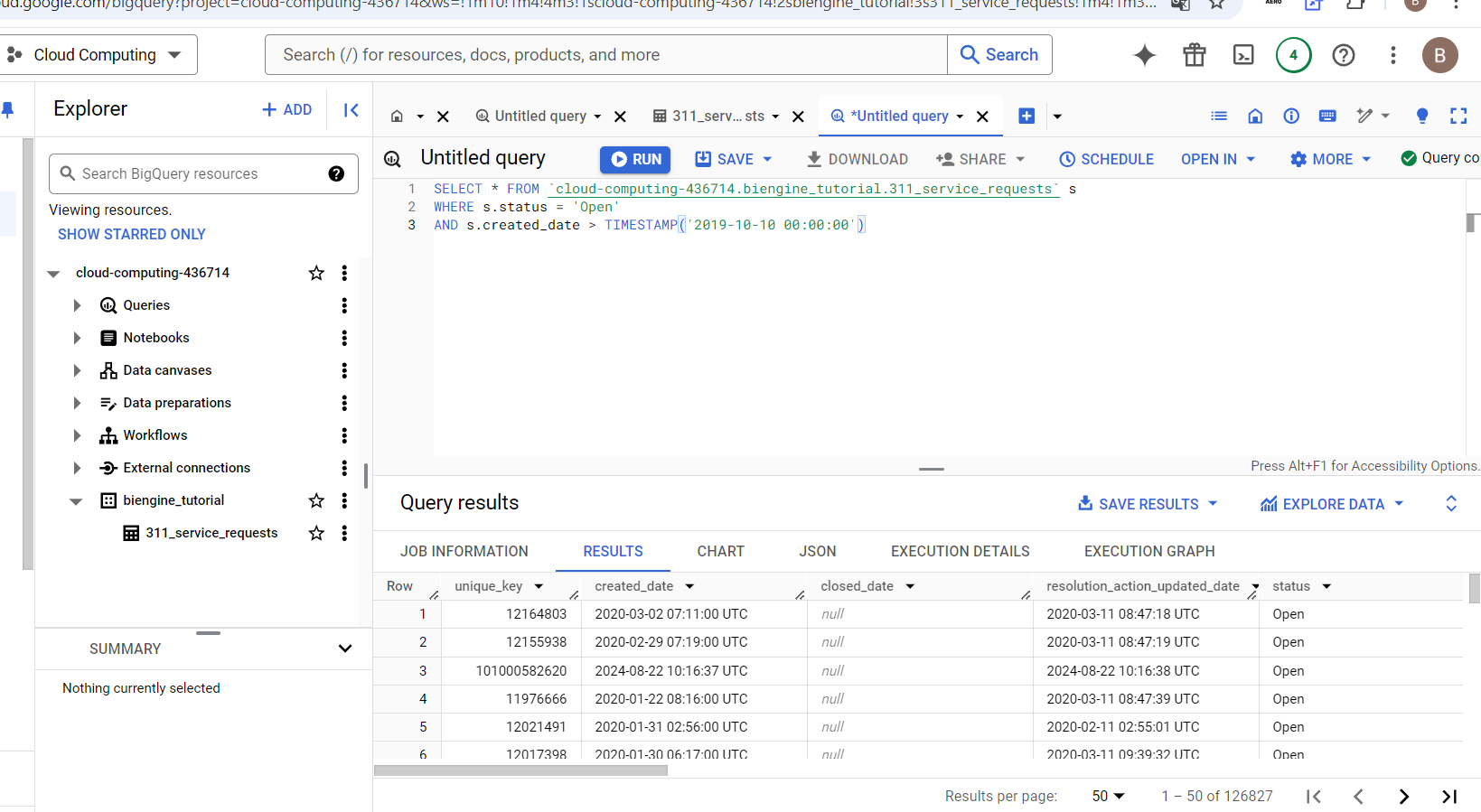


Figure 43: Query

1. Sql in Figure 44 and Figure 46 and corresponding visualisation in Looker Studio in Figure 45 and Figure 47

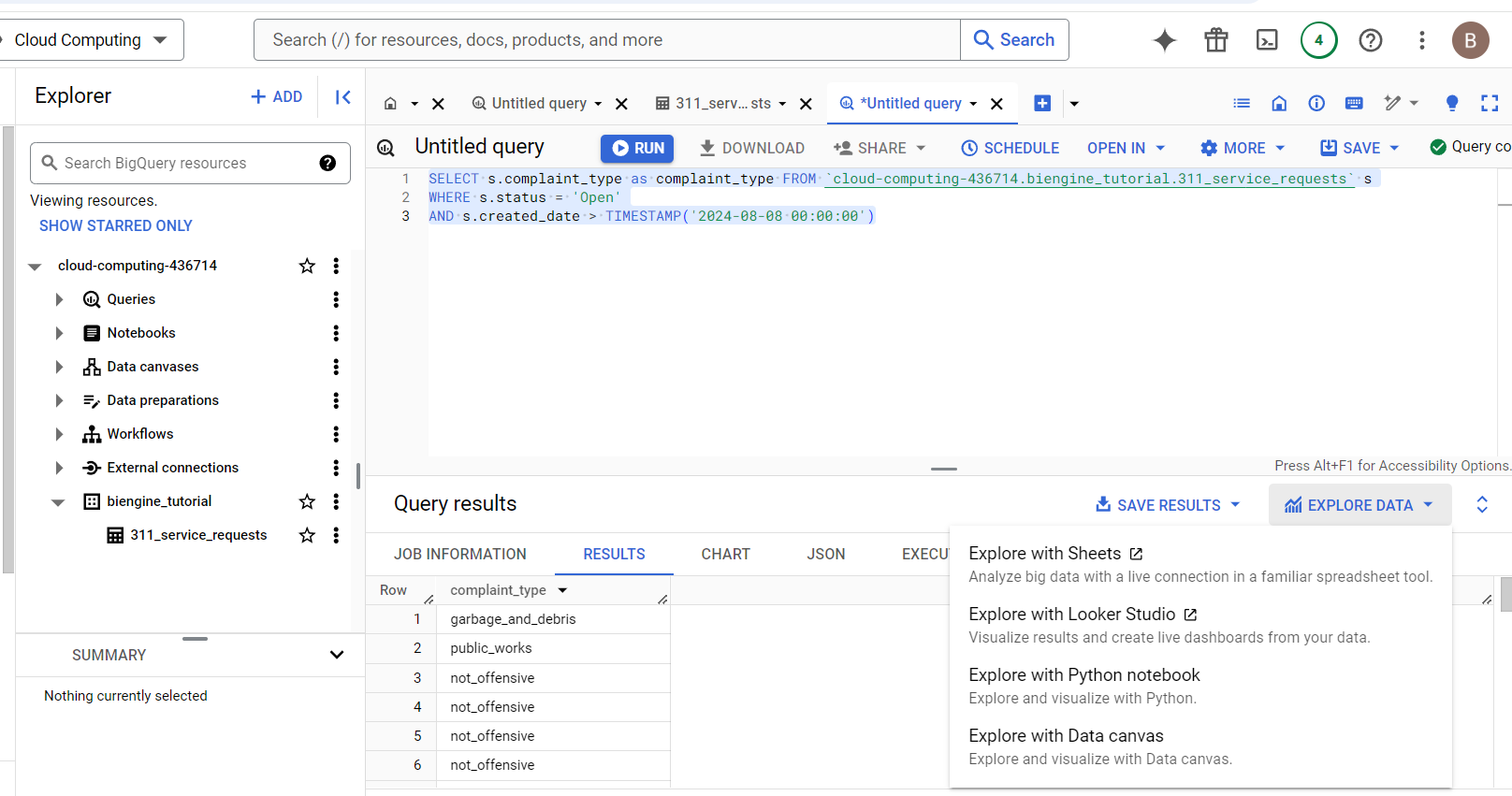


Figure 44: Query

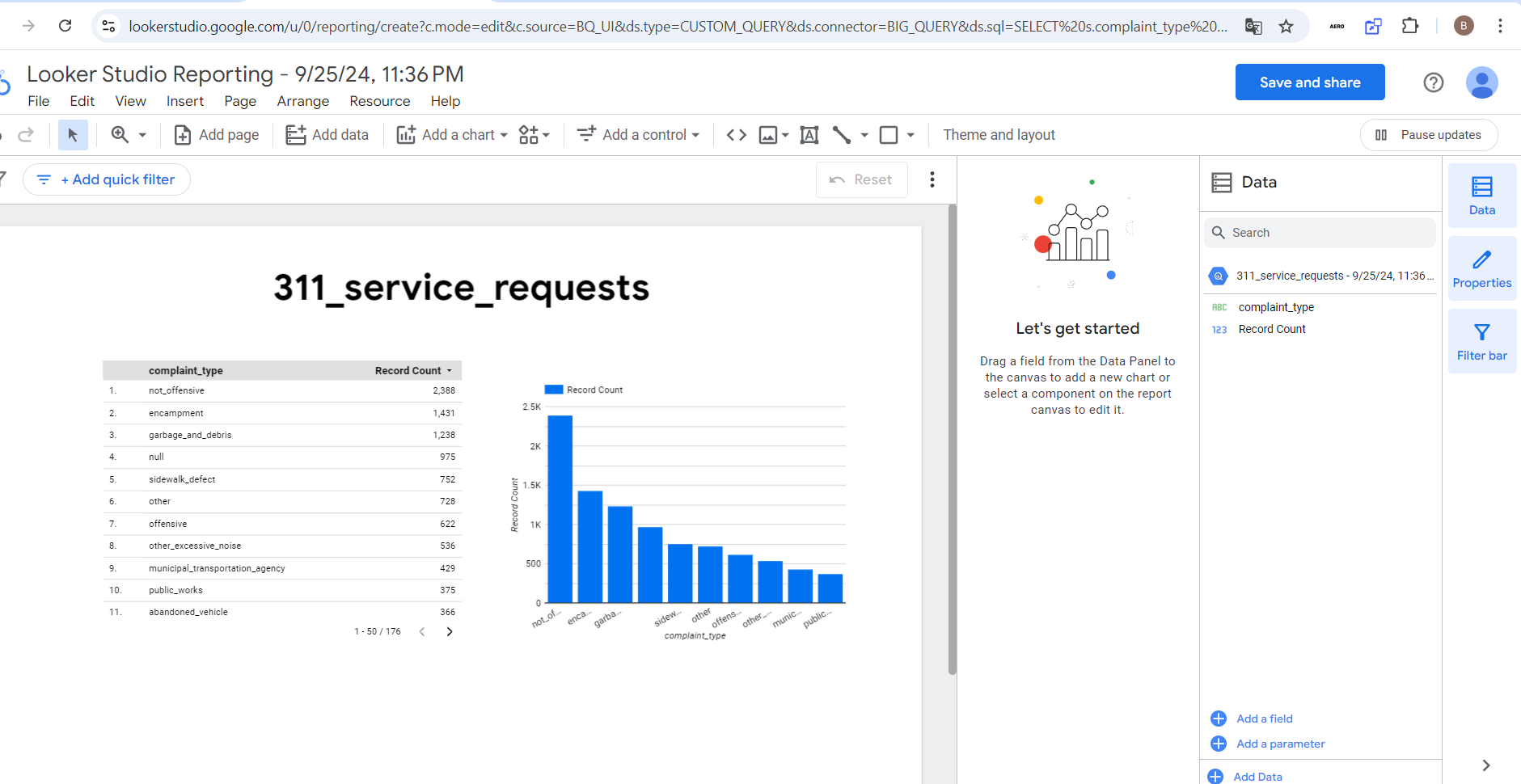


Figure 45: Visualisation of the query

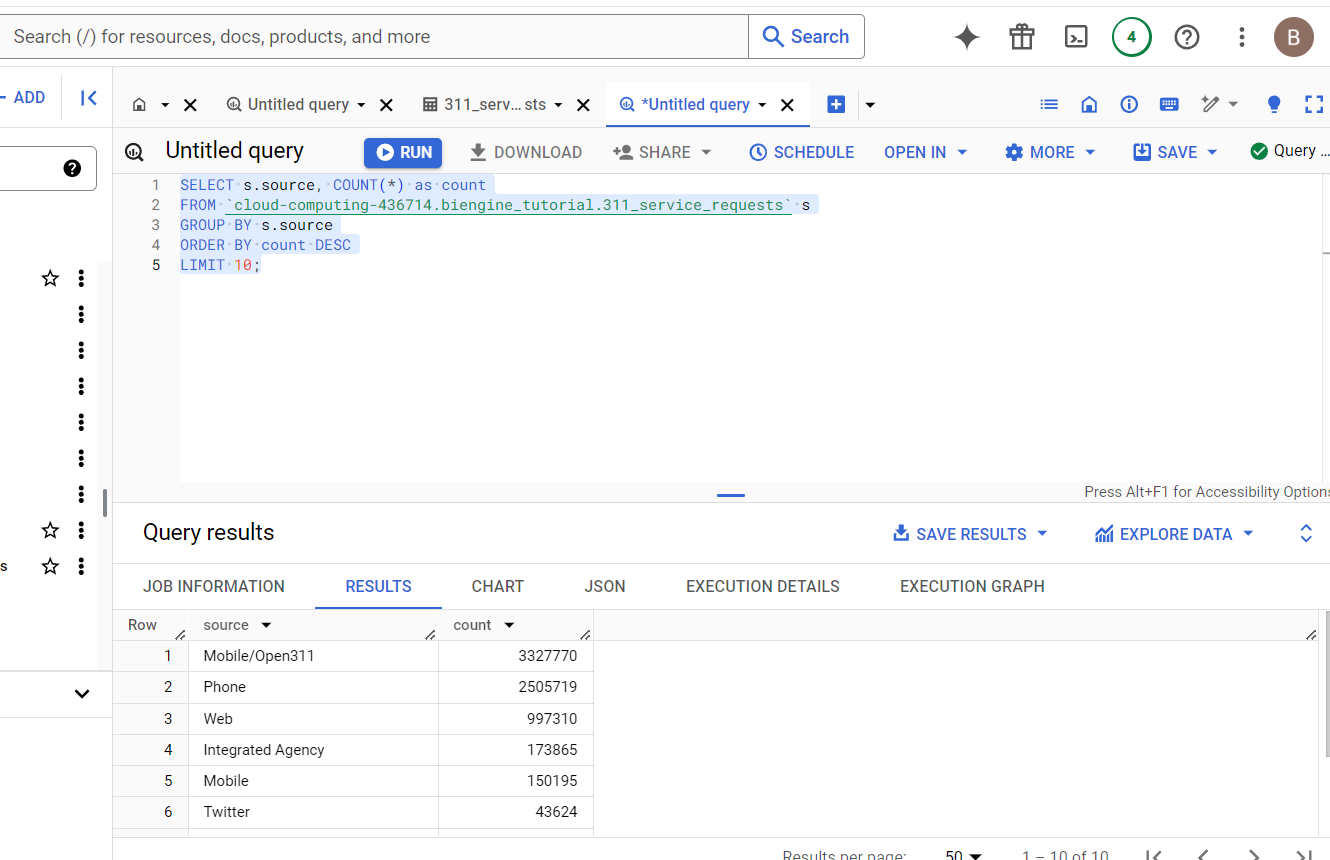


Figure 46: Query

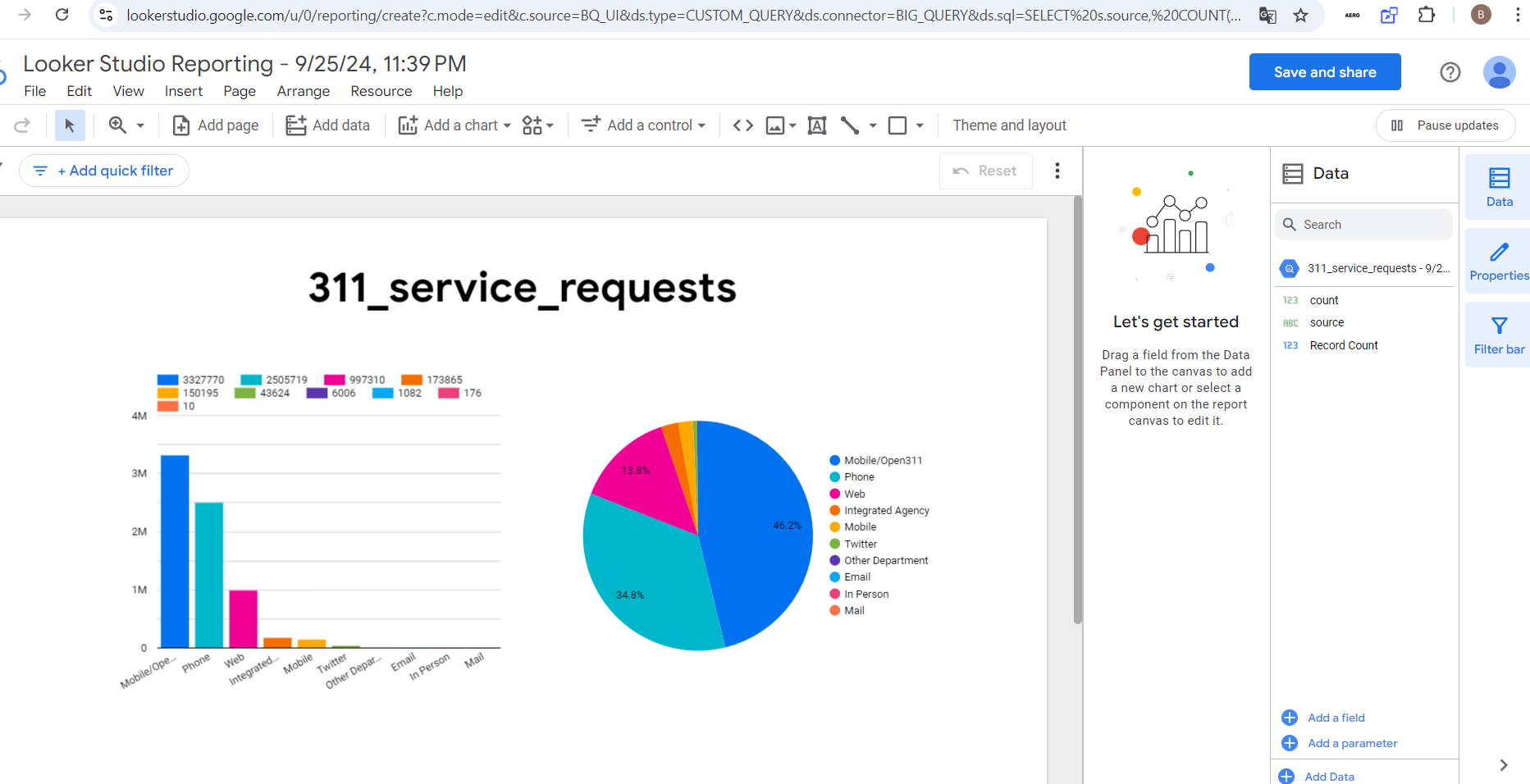


Figure 47: Visualisation

Conclusion on visualisation. This data includes all San Francisco 311 service requests from July 2008 to the present and is updated daily. 311 is a non-emergency number that provides access to non-emergency municipal services. Most of the service requests were tracked directly from the mobile call on 311, over a third of the requests were through the phone and approximately 14% via the web. Most requests were non-offensive. but 622 were offensive. Mostly they were about garbage and debris or sidewalk defects.