

Assignment 1, Cloud Computing

Exercise 1: Understanding Cloud Computing Models

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

	IaaS	PaaS	SaaS
Control	High control over infrastructure and OS.	Moderate Control.	Minimal control.
Flexibility	High flexibility in configuration.	Moderate flexibility with tools and platforms.	Low flexibility which is limited by the capabilities of the provider.
Use cases	Hosting websites, virtual data centers, disaster recovery.	Developing web applications, APIs, and mobile backends.	Email, CRM, collaboration tools.

Which GCP services fall under each of these models?

IaaS: Google Compute Engine, Google Cloud Storage

PaaS: Google App Engine, Google Cloud Functions

SaaS: Google Apps, Google Workspace

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

IaaS: An online based startup is facing the problem of low web traffic. Hence, it is forced to quickly grow its infrastructure. It utilizes Google Compute Engine to create virtual servers so as to create machines and skip the cost of buying physical units.

PaaS: A development team has set a goal to develop the web app within the shortest time frame possible. That is why they chose Google App Engine to be able to concentrate on coding and deploying applications instead of servers.

SaaS: A startup looks for a simple tool to manage all organizational emails and documents. Which they use Google Workspace for, to have all the tools and apps they need in one space, which is easy to access.

Exercise 2: Exploring Google Cloud Platform's Core Services

What is the primary use case of Compute Engine?

The main application of Google Compute Engine is to enable business/organisations to have scalable virtual machines for hosting websites,

running applications among others uses such as data analysis and highly computational tasks.

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

GKE manages containerized applications by doing a large amount of work that would otherwise have to be manually done in establishing Kubernetes clusters. It does things such as updates, scalability and load balancing without having developers to necessarily worry about such things since they can easily concentrate on their application development.

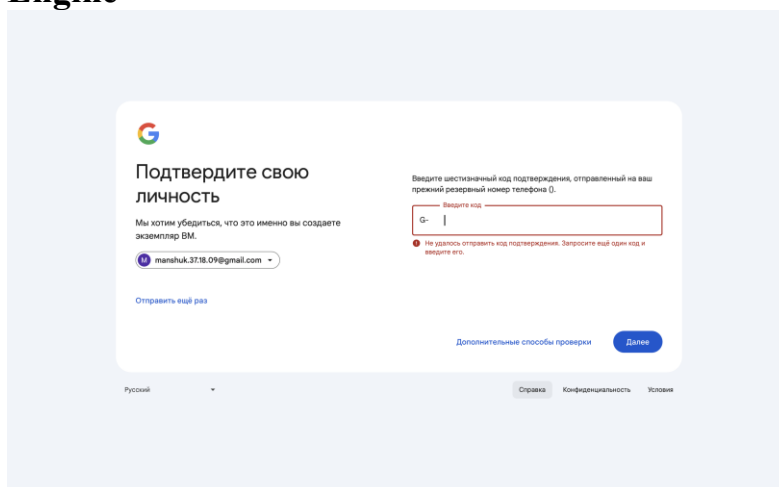
What advantages does Cloud Storage offer for data management?

First, it affords the capacity to add more information as you like or require without a physical barrier to the extension of this capability. One more advantage is high availability and durability, so your data is secure and can be viewed at any time. In addition, it improves the ease of backing up and disaster recovery operations because you can simply create and restore a backup. Moreover, it can be used hand in hand with other Google Cloud services, which make it handy in managing data in your applications.

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

It provides them serverless architecture, this is they do not need to deal with infrastructure. BigQuery uses SQL, which enables analysts to query this database without necessarily having to program IT. Also, it is very compatible with other Google Cloud solutions and can perform machine learning right from the application, providing businesses with insights in a short span of time.

Exercise 3: Creating and Managing Virtual Machines with Compute Engine



I couldn't create a VM, because the code didn't come, and after several attempts it was blocked for 4 hours several times.

Steps:

- I went to the Google Cloud Console and in the left sidebar clicked on Compute Engine. Clicked on “Create instance” button.
- Gave the name for the VM. Selected the region and zone to deploy the VM. Selected the VM provisioning model. Chose the machine type, then OS and Storage. Chose networking configurations. Then clicked the “Create” button to launch the VM.
- Connected to VM by clicking on the “SSH” button next to the VM in the Console. In the opened terminal wrote commands:
“sudo apt update
sudo apt install apache2”
- In the console selected the VM and clicked on the “Delete” button. Then clicked on the “Start” button to restart the VM. To remove the VM, selected and clicked on the “Delete” button.

What steps did you follow to create the VM?

I went to the Google Cloud Console and in the left sidebar clicked on Compute Engine. Clicked on “Create instance” button.

Gave the name for the VM. Selected the region and zone to deploy the VM.

Selected the VM provisioning model. Chose the machine type, then OS and Storage. Chose networking configurations. Then clicked the “Create” button to launch the VM.

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

Connected to VM by clicking on the “SSH” button next to the VM in the Console. In the opened terminal wrote commands:

```
“sudo apt update  
sudo apt install apache2”
```

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

When VM is stopped we can start again later without losing anything. Because the VM is turned off, but all data and settings are preserved.

When VM is deleted, we can’t recover any data stored in VM. Since deleting permanently erases the VM and all its contents.

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

Steps:

First, I needed a basic web application. I decided to use the application that I had made before.

Then, I wrote a Dockerfile to containerize my application.

Built the Docker image using the command: `docker build -t`

After building the image, I needed to push it to GCR. First, I authenticated my Docker client with Google Cloud using the command: `gcloud auth configure-docker`

Then, I pushed the image to GCR using the command: `docker push`

Went to the Google Cloud Console and navigated to the Kubernetes Engine section. Clicked “Create Cluster” and chose my settings, such as the number of nodes and the region. After creating the cluster, I authenticated my local `kubectl` command to use it with: `gcloud container clusters`

I created a deployment file to define the deployment. And applied this deployment using: `kubectl apply -f`

To make my application accessible from the internet, I created a service using the command: `kubectl expose deployment --type=LoadBalancer --port 80 --target-port 8080`

Checked if my application was running correctly using: `kubectl get services`

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

First, I needed a basic web application. I decided to use the application that I had made before.

Then, I wrote a Dockerfile to containerize my application.

Built the Docker image using the command: `docker build -t`

After building the image, I needed to push it to GCR. First, I authenticated my Docker client with Google Cloud using the command: `gcloud auth configure-docker`

Then, I pushed the image to GCR using the command: `docker push`

What steps were involved in setting up the GKE cluster?

Went to the Google Cloud Console and navigated to the Kubernetes Engine section. Clicked “Create Cluster” and chose my settings, such as the number of nodes and the region. After creating the cluster, I authenticated my local `kubectl` command to use it with: `gcloud container clusters`

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

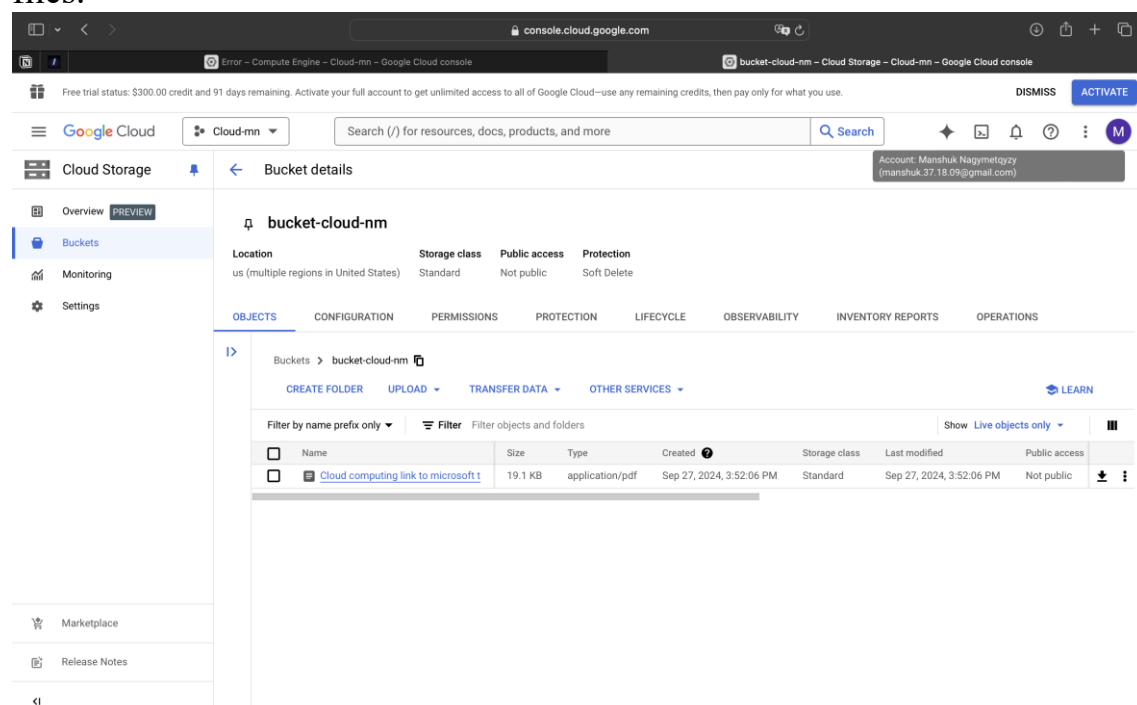
To make my application accessible from the internet, I created a service using the command: `kubectl expose deployment --type=LoadBalancer --port 80 --target-port 8080`

Checked if my application was running correctly using: `kubectl get services`

Exercise 5: Storing and Accessing Data in Google Cloud Storage

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

Navigated to the Google Cloud Console. Selected “Cloud Storage” and clicked “Create Bucket.” Chose a name, location and configured settings. Uploaded files.



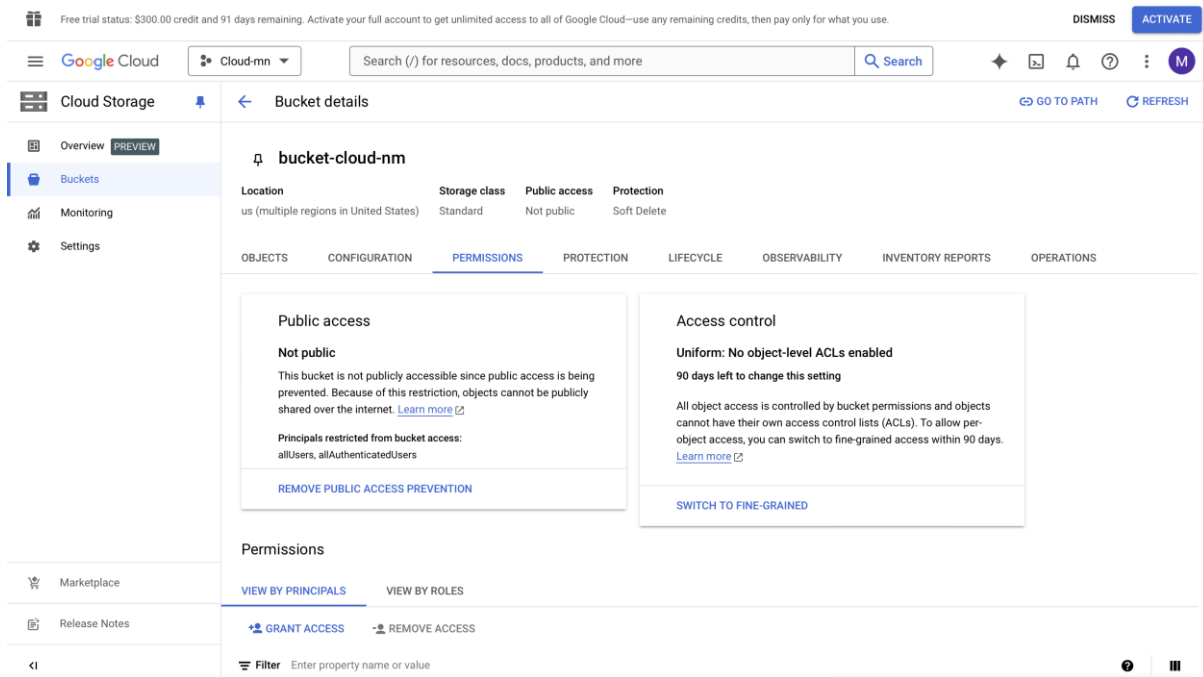
During the setup, there was options like the bucket name, location (like multi-region or single region), storage class (standard, nearline, coldline, archive), and access control settings (public or private). Also features like versioning and lifecycle management.

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

Public buckets can be accessed from the internet by any user, while private buckets may be accessed by their owners only or by users belonging to specific roles.

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

By selecting the file, clicking the Permissions, and configuring settings:



Exercise 6: Analyzing Data with BigQuery

Steps:

Access BigQuery in the Google Cloud Console.

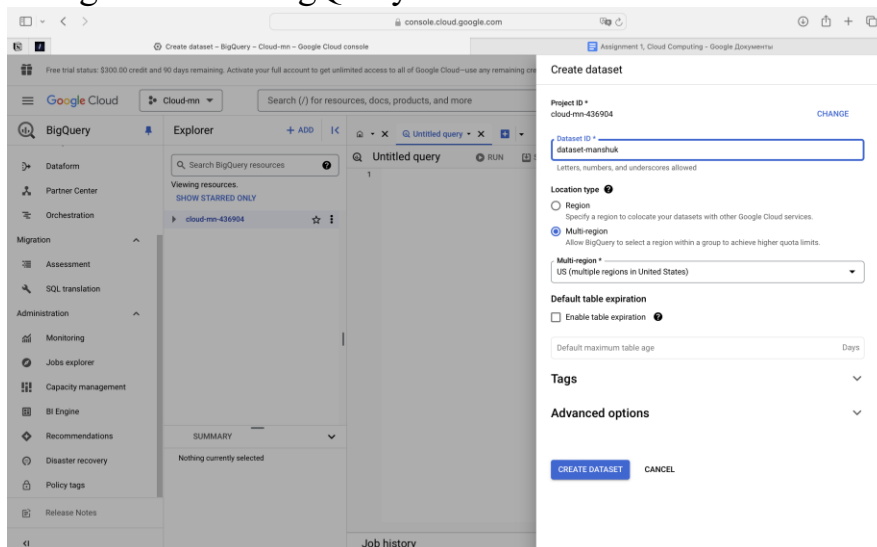
Create a dataset and table by importing a sample dataset provided by Google.

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

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

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

Navigated to the BigQuery Section.



Clicked “Create Dataset” and filled in the dataset ID and other relevant options, then clicked “Create dataset”.

The screenshot shows the Google Cloud console interface. On the left is a navigation menu with categories like Dataform, Partner Center, and Migration. The main area displays the 'dataset_manshuk' dataset page. The 'Dataset info' section on the right lists metadata: Dataset ID (cloud-mn-436904.dataset_manshuk), Created (Sep 28, 2024, 12:57:32 PM UTC+5), Default table expiration (Never), Last modified (Sep 28, 2024, 12:57:32 PM UTC+5), Data location (US), Description, Default collation, Default rounding mode (ROUNDING_MODE_UNSPECIFIED), Case insensitive (false), Labels, and Tags. Below this is the 'Dataset replica info' section showing the Primary location as US. At the top right of the dataset page, there is a 'CREATE TABLE' button. A notification at the bottom center states: "'dataset_manshuk' created. GO TO DATASET".

Within the dataset, clicked “Create table”. For "Source," selected upload csv file. Defined table name, schema, and other options, then clicked "Create Table."

The screenshot shows the 'Create table' dialog box. It is divided into three main sections: Source, Destination, and Schema. In the 'Source' section, 'Create table from' is set to 'Upload', 'Select file' is 'customers-100.csv', and 'File format' is 'CSV'. In the 'Destination' section, 'Project' is 'cloud-mn-436904', 'Dataset' is 'dataset_manshuk', and 'Table' is 'table_manshuk'. A note below the table name states: 'Maximum name size is 1,024 UTF-8 bytes. Unicode letters, marks, numbers, connectors, dashes, and spaces are allowed.' In the 'Schema' section, 'Auto detect' is checked, and a message says: 'Schema will be automatically generated.' At the bottom, there is a 'CANCEL' button and a notification: 'Load job created. GO TO JOB'.

The screenshot shows the Google Cloud BigQuery console interface. On the left is a navigation menu with categories like Dataform, Partner Center, Migration, Assessment, SQL translation, Administration, Monitoring, Jobs explorer, Capacity management, BI Engine, Recommendations, Disaster recovery, Policy tags, and Release Notes. The main area is titled 'table_manshuk' and shows the 'SCHEMA' tab. A table lists the fields of the dataset:

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
Index	INTEGER	NULLABLE	-	-	-	-	-
Customer Id	STRING	NULLABLE	-	-	-	-	-
First Name	STRING	NULLABLE	-	-	-	-	-
Last Name	STRING	NULLABLE	-	-	-	-	-
Company	STRING	NULLABLE	-	-	-	-	-
City	STRING	NULLABLE	-	-	-	-	-
Country	STRING	NULLABLE	-	-	-	-	-
Phone 1	STRING	NULLABLE	-	-	-	-	-
Phone 2	STRING	NULLABLE	-	-	-	-	-
Email	STRING	NULLABLE	-	-	-	-	-
Subscription Date	DATE	NULLABLE	-	-	-	-	-
Website	STRING	NULLABLE	-	-	-	-	-

A notification at the bottom states: 'table_manshuk' created. Below the schema table, there is a 'SUMMARY' section showing details for 'table_manshuk' in the 'cloud-mn-436904.dataset_manshuk' dataset, including the last modified date (Sep 28, 2024, 1:51:28 PM UTC+5), data location (US), and a description.

How did you write and execute SQL queries in BigQuery?

In the BigQuery console, opened the Query Editor for dataset. For example filtering:

The screenshot shows the BigQuery console with a query editor open. The query being executed is:

```
SELECT * FROM `cloud-mn-436904.dataset_manshuk.table_manshuk` WHERE Country="Canada"
```

The query results are displayed in a table with columns: Row, Index, Customer Id, First Name, Last Name, and Company. The results show two rows of data for customers from Canada:

Row	Index	Customer Id	First Name	Last Name	Company
1	51	Aa20BDe68eAb0e9	Gerald	Hawkins	Phelps, Forbes and Koch
2	70	CC68FD103Bbf22	Riley	Good	Wade PLC

The interface also includes a 'Query results' section with options to 'SAVE RESULTS', 'EXPLORE DATA', and 'CHART'. At the bottom, it shows 'Results per page: 50' and '1 - 2 of 2'.

For example sorting:

The screenshot shows the Google Cloud BigQuery console interface. On the left is a navigation menu with options like Dataform, Partner Center, Orchestration, Migration, Assessment, SQL translation, Administration, Monitoring, Jobs explorer, Capacity management, BI Engine, Recommendations, Disaster recovery, Policy tags, and Release Notes. The main area displays a query titled 'Untitled query' with the following SQL: `SELECT * FROM `cloud-mn-436984.dataset_manshuk.table_manshuk` ORDER BY `Subscription Date` DESC`. Below the query editor, the 'Query results' section shows a table with 9 rows of data. The columns are Row, Index, Customer Id, First Name, Last Name, and Company. The data includes names like Frederick Harper, Marcus Moody, Luis Greer, Chloe Hutchinson, Emma Cunningham, Miranda Beasley, Greg Mata, Eddie Jimenez, and Gloria McCall, along with their respective companies.

Row	Index	Customer Id	First Name	Last Name	Company
1	28	b92EBf8a3f0E6	Frederick	Harper	Hinton, Chaney and Stoi
2	26	09D7D7C8Fe9aea	Marcus	Moody	Giles Ltd
3	44	D3fC11A9C235Dc6	Luis	Greer	Cross PLC
4	64	FCBdICEAe20A8Dc	Chloe	Hutchinson	Simon LLC
5	86	C6763c99d0bd16D	Emma	Cunningham	Stephens Inc
6	16	8cad0b4C8caaec	Miranda	Beasley	Singleton and Sons
7	18	F8Aa9d6DfcBeeF8	Greg	Mata	Valentine LLC
8	63	aEcbE5365BbC67D	Eddie	Jimenez	Caldwell Group
9	48	283DFCD0Db40aF	Gloria	McCall	Brennan, Acosta and Ra

For example aggregation:

The screenshot shows the Google Cloud BigQuery console interface. The left navigation menu is the same. The main area displays the schema for the table 'table_manshuk'. The schema table has columns: Field name, Type, Mode, Key, and Collation. The fields include Index (INTEGER, NULLABLE), Customer Id (STRING, NULLABLE), First Name (STRING, NULLABLE), Last Name (STRING, NULLABLE), Company (STRING, NULLABLE), City (STRING, NULLABLE), Country (STRING, NULLABLE), Phone 1 (STRING, NULLABLE), Phone 2 (STRING, NULLABLE), Email (STRING, NULLABLE), Subscription Date (DATE, NULLABLE), and Website (STRING, NULLABLE). Below the schema, there are buttons for 'EDIT SCHEMA' and 'VIEW ROW ACCESS POLICIES'. To the right, a query titled 'Untitled query' is shown with the SQL: `SELECT 'First Name', COUNT(*) AS count FROM `cloud-mn-436984.dataset_manshuk.table_manshuk` GROUP BY 'First Name'``. The 'Query results' section shows a table with 7 rows of data. The columns are Row, First Name, and count. The data includes names like Corey, Alexandria, Darrell, Miranda, Chad, Tracey, and Shelley, each with a count of 1.

Field name	Type	Mode	Key	Collation
Index	INTEGER	NULLABLE	-	-
Customer Id	STRING	NULLABLE	-	-
First Name	STRING	NULLABLE	-	-
Last Name	STRING	NULLABLE	-	-
Company	STRING	NULLABLE	-	-
City	STRING	NULLABLE	-	-
Country	STRING	NULLABLE	-	-
Phone 1	STRING	NULLABLE	-	-
Phone 2	STRING	NULLABLE	-	-
Email	STRING	NULLABLE	-	-
Subscription Date	DATE	NULLABLE	-	-
Website	STRING	NULLABLE	-	-

Row	First Name	count
1	Corey	1
2	Alexandria	1
3	Darrell	1
4	Miranda	1
5	Chad	1
6	Tracey	1
7	Shelley	1

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

I gain insights into data distribution, such as the percentage of clients from various places, by grouping and summarizing data. This aids in locating prospective segmentation regions.