GCP Assignment 3

1. What distinguishes the Google Cloud Machine Learning Engine from others?

The Google Cloud ML Engine is a hosted platform to run machine learning training jobs and predictions at scale. The service treats these two processes (training and predictions) independently. It is possible to use Google Cloud ML Engine just to train a complex model by leveraging the GPU and TPU infrastructure. The outcome from this step — a fully-trained machine learning model — can be hosted in other environments including on-prem infrastructure and public cloud. The service can also be used to deploy a model that is trained in external environments. Cloud ML Engine automates all resource provisioning and monitoring for running the jobs. It can also manage the lifecycle of deployed models and their versions.

Apart from training and hosting, Cloud ML Engine can also perform hyperparameter tuning that influences the accuracy of predictions. Without automated hyperparameter tuning, data scientists will have to experiment with multiple values while evaluating the accuracy of the results.
Whether Google Cloud Platform is less expensive than its competitors depends a lot on which type of workloads you want to run and how you run them. It offers discounts based on what it calls sustained use, meaning situations where customers keep a workload running for extended periods.

This distinguishes the Google Cloud Machine Learning Engine from others.

2. What are GCP's cloud storage libraries and tools?

At the center level, **XML API** and **JSON API** are there for the cloud storage on Google Cloud Platform. Yet, alongside these, there are following choices furnished by Google to collaborate with the cloud storage.

Google Cloud Platform Console, which performs fundamental tasks on objects and buckets

Cloud Storage Client Libraries, which give programming backing to different languages including Java, Ruby, and Python

GustilCommand-line Tool, which gives a command-line interface to the cloud storage.

There are numerous external libraries and tools, for example, terraform

3. Define the Google Cloud pricing model.

You can choose one of the following pricing models for your product:

- **Free**: Customers only pay for the Google Cloud resources that they use. If you are offering your product free of charge, skip to Submitting your pricing for review.

- **Subscription-based pricing**: Customers pay a flat monthly fee for using your software. For partial months, the cost is prorated.
- **Usage-based pricing**: Customers pay for your software based on measurements that you choose, called *metrics*, such as data or storage. For example, you can set your price by gibibyte-hours of storage. If you choose a usage-based pricing model, your app must measure and report usage information to Google.
- **Combined pricing**: Customers pay a base subscription fee for using your software, and additional charges based on their usage.

4. The Google Compute Engine API has a variety of authentication mechanisms.

Google Cloud APIs use the OAuth 2.0 protocol for authenticating both user accounts and service accounts. The OAuth 2.0 authentication process determines both the principal and the application.

Most Google Cloud APIs also support anonymous access to public data using API keys. However, API keys only identify the application, not the principal. When using API keys, the principal must be authenticated by other means.

Google Cloud APIs support multiple authentication flows for different runtime environments. For the best developer experience, we recommend using Google Cloud Client Libraries with Google Cloud APIs. They use Google-provided authentication libraries that support a variety of authentication flows and runtime environments.

To build an application using Google Cloud APIs, follow these general steps:

- Choose and use the provided Google Cloud Client Libraries
- Determine the correct authentication flow for your application
- Find or create the application credentials needed for your application
- Pass the application credentials to the client libraries at application startup time, ideally through Application Default Credentials (ADC)

You should choose application credentials based on what your application needs and where it runs. The following table provides some general recommendations for common requirements:

Requirement	Recommendation	Comment
Accessing public data anonymously	API key	An API key only identifies the application and doesn't require user authentication. It is sufficient for accessing public data.
Accessing private data on behalf of an end user	OAuth 2.0 client	An OAuth 2.0 client identifies the application and lets end users authenticate your application with Google. It allows your application to access Google Cloud APIs on behalf of the end user.
Accessing private data on behalf of a service account inside Google Cloud environments	Environment-provided service account	If your application runs inside a Google Cloud environment, such as Compute Engine, App Engine, GKE, Cloud Run, or Cloud Functions, your application should use the service account provided by the environment.

Requirement	Recommendation	Comment
		Google Cloud Client Libraries will automatically find and use the service account credentials.
Accessing private data on behalf of a service account outside Google Cloud environments	Service account key	You need to create a service account, and download its private key as a JSON file. You need to pass the file to Google Cloud Client Libraries, so they can generate the service account credentials at runtime. Google Cloud Client Libraries will automatically find and use the service account credentials by using the GOOGLE_APPLICATION_CREDENTIALS environment variable.

5. Create Google Cloud service accounts and explain how to use them.

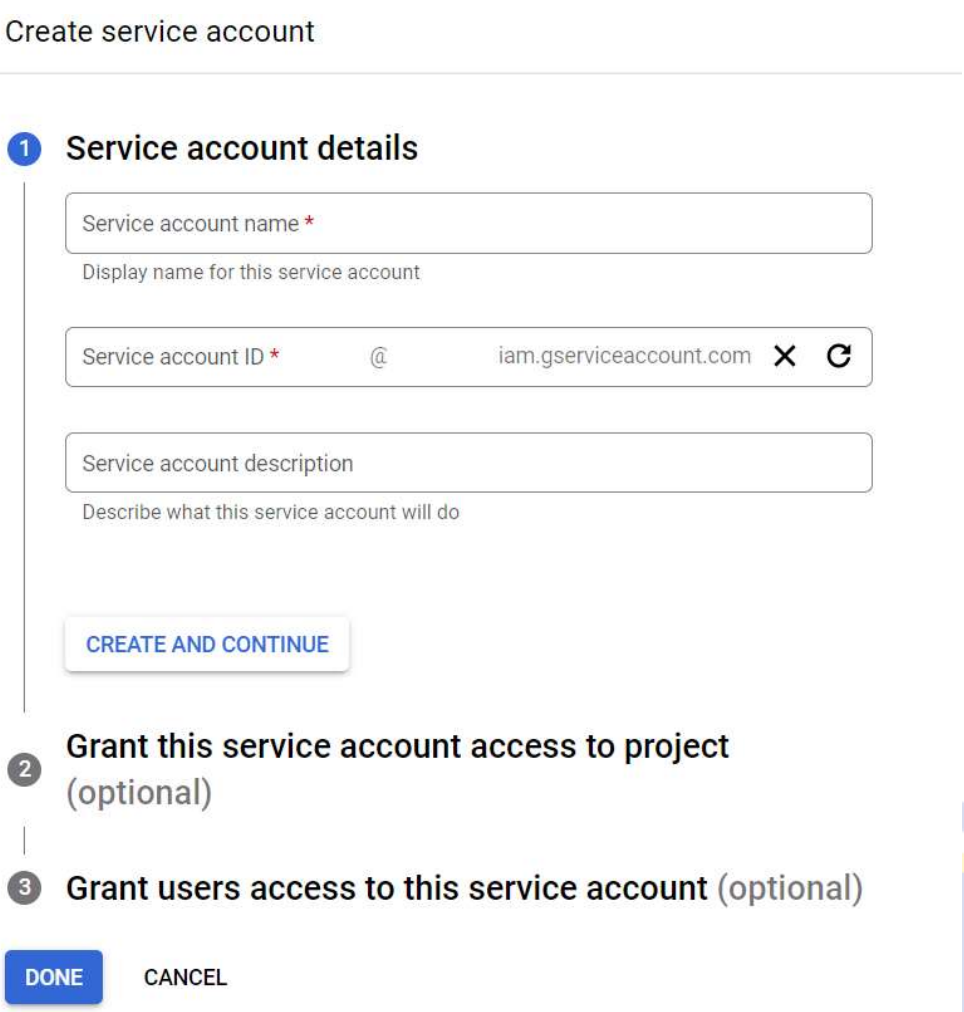
    1. In the Cloud Console, go to the **Create service account** page.

        Go to Create service account
    2. Select a project.
    3. Enter a service account name to display in the Cloud Console.

        The Cloud Console generates a service account ID based on this name. Edit the ID if necessary. You cannot change the ID later.
    4. Optional: Enter a description of the service account.
    5. If you do not want to set access controls now, click **Done** to finish creating the service account.

        To set access controls now, click **Create and continue** and continue to the next step.
    6. Optional: Choose one or more IAM roles to grant to the service account on the project.
    7. When you are done adding roles, click **Continue**.
    8. Optional: In the **Service account users role** field, add members that can impersonate the service account.
    9. Optional: In the **Service account admins role** field, add members that can manage the service account.
    10. Click **Done** to finish creating the service account.

After creating service account eirther create key and use them while accessing the gcp service.



6. How to make a Google Cloud Storage project?

1. In the Google Cloud Console, go to the Cloud Storage **Browser** page.

   Go to Browser

2. Click **Create bucket**.
3. On the **Create a bucket** page, enter your bucket information. To go to the next step, click **Continue**.
   - For **Name your bucket**, enter a name that meets the bucket name requirements.
   - For **Choose where to store your data**, select a **Location type** and **Location** where the bucket data will be permanently stored.
   - For **Choose a default storage class for your data**, select a storage class for the bucket. The default storage class is assigned by default to all objects uploaded to the bucket.

Note: The **Monthly cost estimate** panel in the right pane estimates the bucket's monthly costs based on your selected storage class and location, as well as your expected data size and operations.

- For **Choose how to control access to objects**, select whether or not your bucket enforces public access prevention, and select an **Access control** model for your bucket's objects.

  Note: If public access prevention is already enforced by your project's organization policy, the **Prevent public access** toggle is locked.

- For **Choose how to protect object data**, configure **Protection tools** if desired, and select a **Data encryption** method.

4. Click **Create**.