**GCPAssignment1**

**1. What is the GCP project quota? If necessary, how can GCP quota be increased?**

**Definition**: GCP project quota limits the number of resources (like API calls, storage, and compute instances) you can use in your GCP project. Quotas protect the community of Google Cloud users by preventing unforeseen spikes in usage.

**Increasing Quota**: To increase your quota:

1. **Go to the Quotas page**: In the Google Cloud Console, navigate to the IAM & admin section and click on "Quotas".
2. **Select the quota**: Find and select the quota you want to increase.
3. **Request Increase**: Click on the "Edit Quotas" button, fill out the necessary fields, and submit your request.

**Real-time Use Case**: If a business is deploying a new application expected to experience rapid growth, they might initially hit their API call limit. They can request an increase in the API quota to ensure their service remains uninterrupted as usage scales up.

**2. What is the definition of a virtual machine? Virtual machine types are offered in GCP. How to create a virtual computer in Google Cloud Platform.**

**Definition**: A virtual machine (VM) is an emulation of a physical computer. VMs run an operating system and applications just like a physical machine.

**VM Types in GCP**:

1. **General-purpose**: Balanced CPU and memory. Suitable for a variety of workloads.
2. **Compute-optimized**: High performance per core. Ideal for CPU-intensive workloads.
3. **Memory-optimized**: High memory configurations. Suitable for in-memory databases.
4. **Storage-optimized**: High performance for data-intensive applications.

**Creating a VM in GCP**:

1. **Navigate to Compute Engine**: Go to the Google Cloud Console and select "Compute Engine".
2. **Create Instance**: Click "Create" to start a new VM instance.
3. **Configure Instance**: Choose the machine type, boot disk, and other configurations.
4. **Create**: Click "Create" to launch the VM.

**Real-time Use Case**: An e-commerce company might use general-purpose VMs to handle their web traffic, compute-optimized VMs for real-time analytics, and memory-optimized VMs for their database servers.

**3. What is Google BigQuery, and how does it work? Replicate certain instances to demonstrate a use case.**

**Definition**: Google BigQuery is a fully managed, serverless data warehouse that allows for super-fast SQL queries using the processing power of Google's infrastructure.

**How it Works**:

1. **Data Ingestion**: Load your data into BigQuery using batch or streaming methods.
2. **SQL Queries**: Use SQL to query your data, taking advantage of BigQuery’s distributed processing capabilities.
3. **Results**: Get quick results even for large datasets.

**Real-time Use Case**: A marketing analytics firm can use BigQuery to analyze large datasets from various campaigns to derive insights about customer behavior. By using SQL queries, they can quickly generate reports and dashboards for their clients.

SELECT campaign, SUM(revenue) as total\_revenue

FROM marketing\_data

WHERE date BETWEEN '2024-01-01' AND '2024-01-31'

GROUP BY campaign

ORDER BY total\_revenue DESC;

### 4. What exactly is the Google Cloud SDK? List the numerous Google Cloud SDK installation options.

**Definition**: The Google Cloud SDK is a set of tools that you can use to manage resources and applications hosted on Google Cloud. It includes the gcloud, gsutil, and bq command-line tools.

**Installation Options**:

1. **Install Script**: Download and run the install script from the Google Cloud website.
2. **Package Managers**: Use package managers like apt for Debian/Ubuntu or brew for macOS.
3. **Docker**: Use a Docker container with the SDK pre-installed.
4. **Manual Installation**: Download the archive and manually extract and configure it.

**5. List the many cloud computing deployment models.**

1. **Public Cloud**: Services offered over the public internet and shared across multiple organizations. E.g., Google Cloud Platform.
2. **Private Cloud**: Cloud infrastructure dedicated to a single organization.
3. **Hybrid Cloud**: Combination of public and private clouds, allowing data and applications to be shared between them.
4. **Multi-Cloud**: Use of multiple cloud services from different providers to reduce reliance on a single provider and enhance redundancy.

**6. Describe the Google Cloud Platform's security features.**

**Security Features**:

1. **IAM**: Identity and Access Management to control who can do what on your cloud resources.
2. **Encryption**: Data is encrypted in transit and at rest.
3. **VPC**: Virtual Private Cloud for network isolation and control.
4. **DLP**: Data Loss Prevention to protect sensitive information.
5. **Security Command Center**: Centralized security management and threat detection.
6. **Cloud Armor**: DDoS protection and WAF capabilities.

**Real-time Use Case**: A financial institution can use GCP's IAM to ensure that only authorized personnel can access sensitive financial data, while encryption protects data at rest and in transit. Cloud Armor can help protect their services from DDoS attacks.

**7. What exactly is Vertex AI? With some usage scenarios, implement Vertex AI.**

**Definition**: Vertex AI is Google Cloud's unified AI platform that enables you to build, deploy, and scale machine learning (ML) models. It integrates with Google Cloud’s data and analytics services.

**Use Case 1: Predictive Maintenance**: A manufacturing company uses Vertex AI to predict equipment failures. By training ML models on historical data, they can forecast when a machine is likely to fail and schedule maintenance proactively.

from google.cloud import aiplatform

# Initialize the AI platform

aiplatform.init(project='my-project', location='us-central1')

# Define the dataset

dataset = aiplatform.TabularDataset.create(

display\_name='maintenance\_dataset',

gcs\_source=['gs://my-bucket/maintenance\_data.csv']

)

# Create and train the model

model = aiplatform.AutoMLTabularTrainingJob(

display\_name='maintenance\_model',

optimization\_prediction\_type='classification',

optimization\_objective='maximize-log-likelihood'

)

model.run(

dataset=dataset,

target\_column='failure',

training\_fraction\_split=0.8,

validation\_fraction\_split=0.1,

test\_fraction\_split=0.1

)

**Use Case 2: Retail Demand Forecasting**: A retail chain uses Vertex AI to forecast product demand across different stores, optimizing inventory and reducing waste.

**Real-time Example**:

1. **Data Ingestion**: Load historical sales data into BigQuery.
2. **Model Training**: Use Vertex AI to train a demand forecasting model.
3. **Deployment**: Deploy the model to predict future demand.

# Deploy the trained model

endpoint = model.deploy(machine\_type='n1-standard-4')

# Make a prediction

prediction = endpoint.predict(instances=[

{'store\_id': '001', 'product\_id': 'A1', 'date': '2024-07-05'}

])

print(prediction)

These examples illustrate how Vertex AI can be applied to various industries to leverage machine learning for predictive analytics and demand forecasting