Cloud Fundamentals Final Report

R Mohan Poornachandra
Sasanka Panda
26 December 2024

Problem Statement

Design and implement a cloud solution for a small business to host a website, store customer data, and ensure robust data security.

Objectives

1. Host a functional, scalable, and secure website.
2. Store customer data in a secure, easily accessible manner.
3. Implement comprehensive security measures to protect data integrity and confidentiality.

Cloud Service Provider Selection - GCP

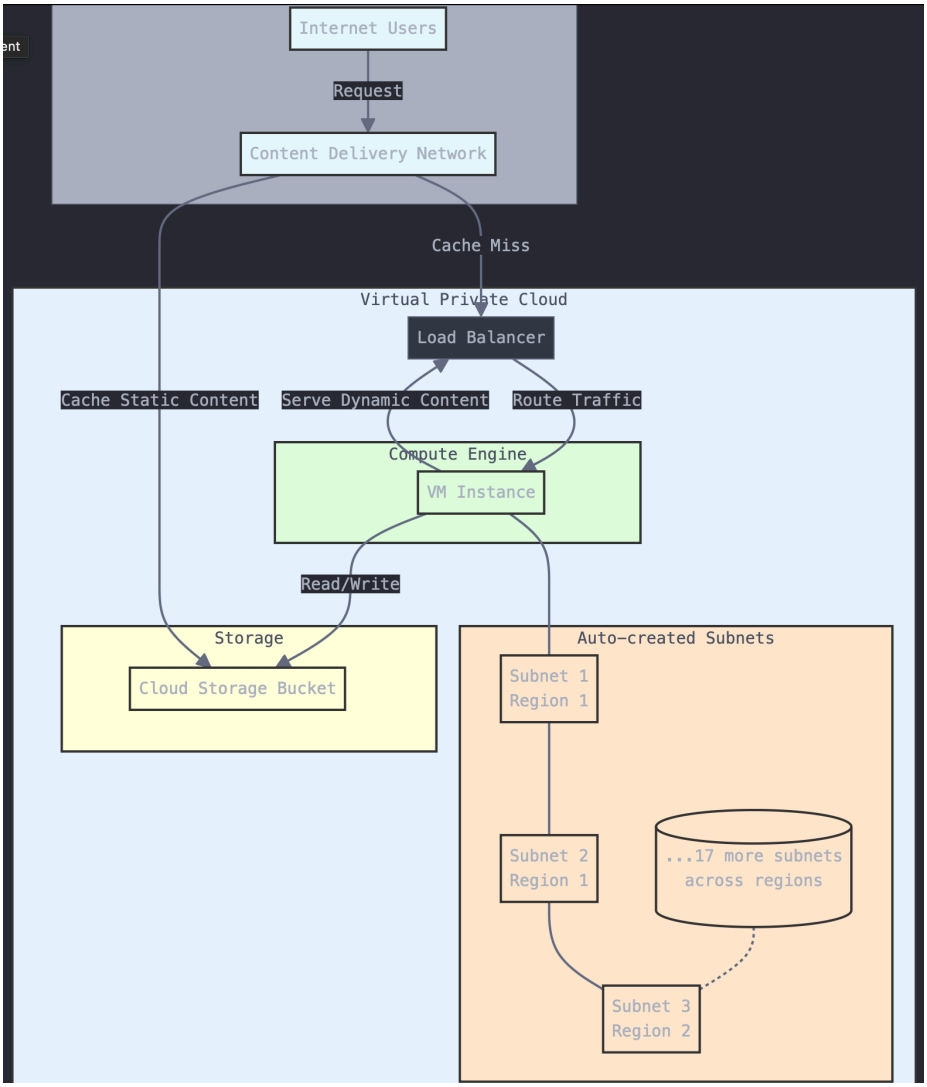
1. Signing up in the GCP platform was way easier and less time consuming. They had various payment options out of which I could choose the most appropriate one.
2. GCP is an excellent choice if your focus is on AI/ML workloads, big data analytics, networking performance, or if you prefer a more transparent and developer-friendly platform.
3. GCP is cost efficient and reliable with its cutting edge technology and security.

Architecture Diagram and List of Required Services

Service Model : PaaS (Platform as a Service)

Deployment Model : Public network

Architecture Diagram:



Components

1. Compute Engine: Needed to perform all the computation of the business.
2. Bucket: A bucket needs to be created from where the website of the business will hosted and all the customer data will be stored.
3. Load Balancer: To enable Content Delivery Network(CDN) in GCP, a load balancer needs to be set up that distributes the traffic based on the load in a server.
4. VPC: A Virtual Private Network is needed to have a secure private connection.

Implementation

1. First, created a GCP account with my personal email and with its easy payment methods I was able to create my account easily.
2. Explored compute services and selected Compute Engine - the basic and easy IaaS compute service. Selected the required memory, CPU and storage. The instance was created and I could easily connect with its terminal.
3. Created a storage bucket by the name clod101 and uploaded all the files needed to host a website. This bucket will also have all the essentials needed to store customer information.
4. To host the website, created a load balancer and enable Content Delivery Network(CDN) in it. Connected the storage bucket "clod101" so that the website in it can be hosted. After creating the load balancer, the frontend IP can be copied and pasted in a new tab with index.html as suffix. This will run the website.
5. Created a Virtual Private Network(VPC) with 20 auto subnets created by google at various regions around the world. This will help in secure connections for the company.

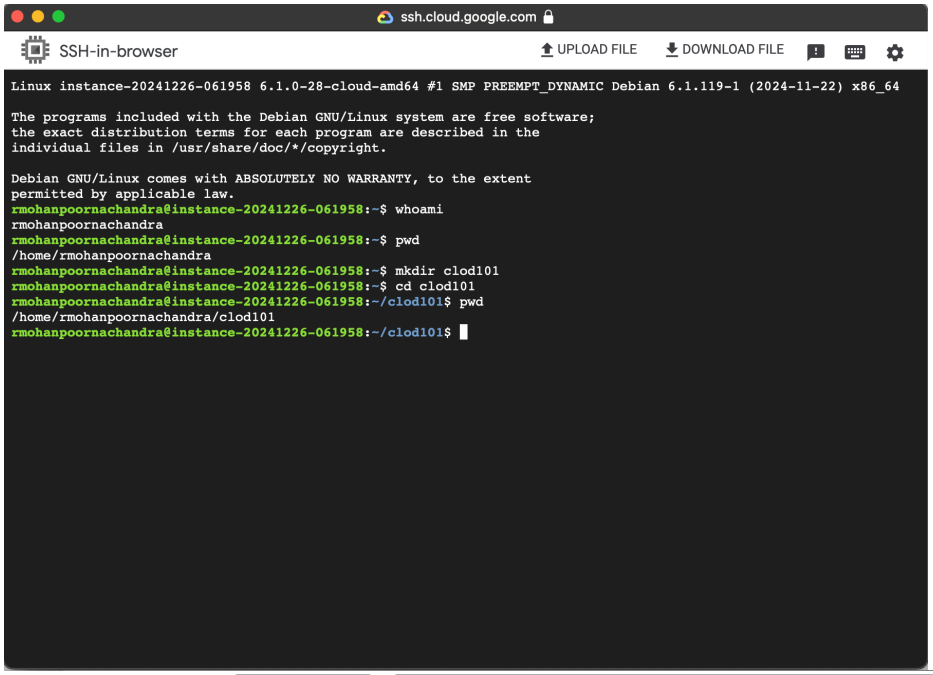
Testing and Observations

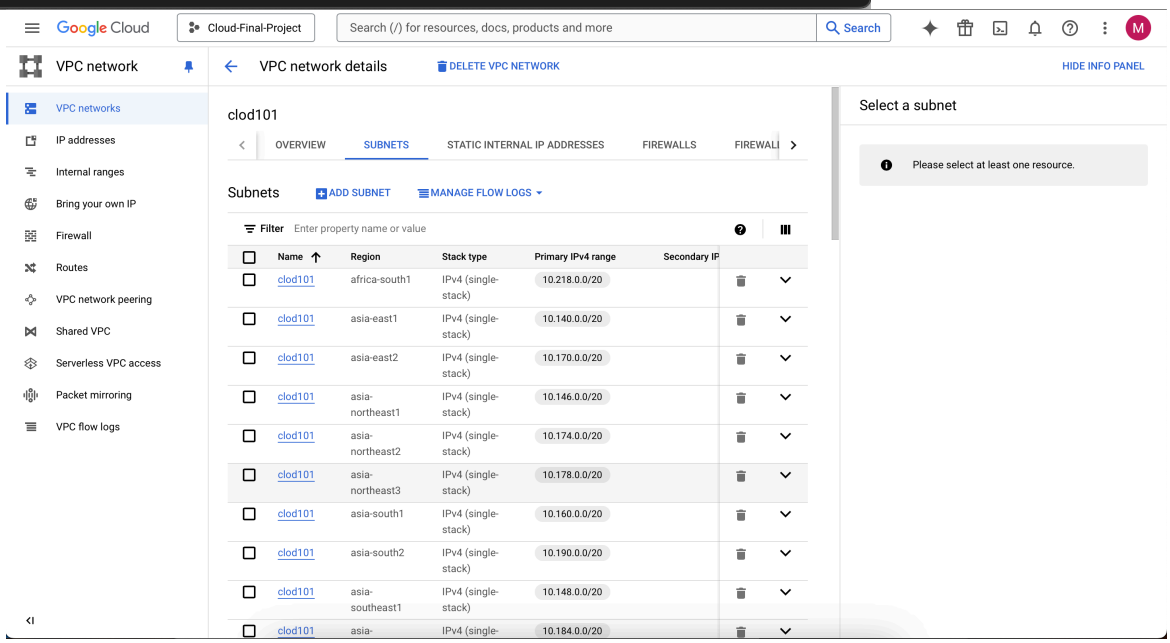
After following all the steps, I was able to get mainly 3 things:

- Access to a Linux OS computer's terminal with which the company can perform its compute needs.
- A fully functional website that can be accessed with the IP address of the frontend load balancer.
- A secure network over which the company can perform its tasks.

Frontend

Protocol ↑	IP:Port	Certificate	Certificate map	SSL Policy	Network Tier ?
HTTP	34.110.141.90:80	-			Premium





Challenges and Solutions

The main challenge faced was that the website was not getting hosted due to some access issues in the bucket. But later with some studies and help from ChatGPT, I was able to solve the issue and the website is hosted fine.

```
▼<Error>
   <Code>AccessDenied</Code>
   <Message>Access denied.</Message>
 </Error>
```

Sources: ChatGPT, QA Platform , Claude