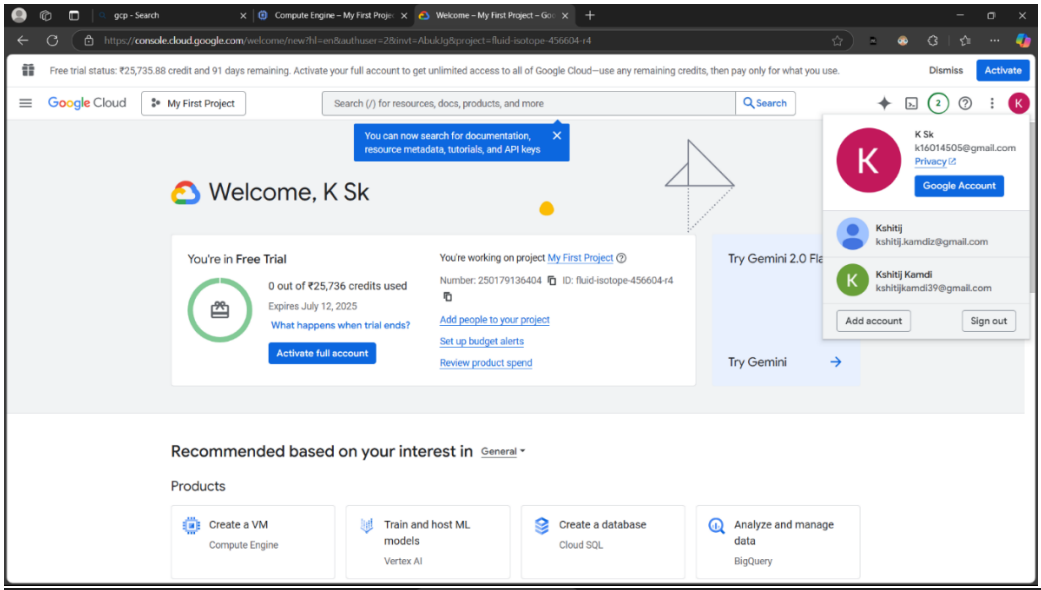
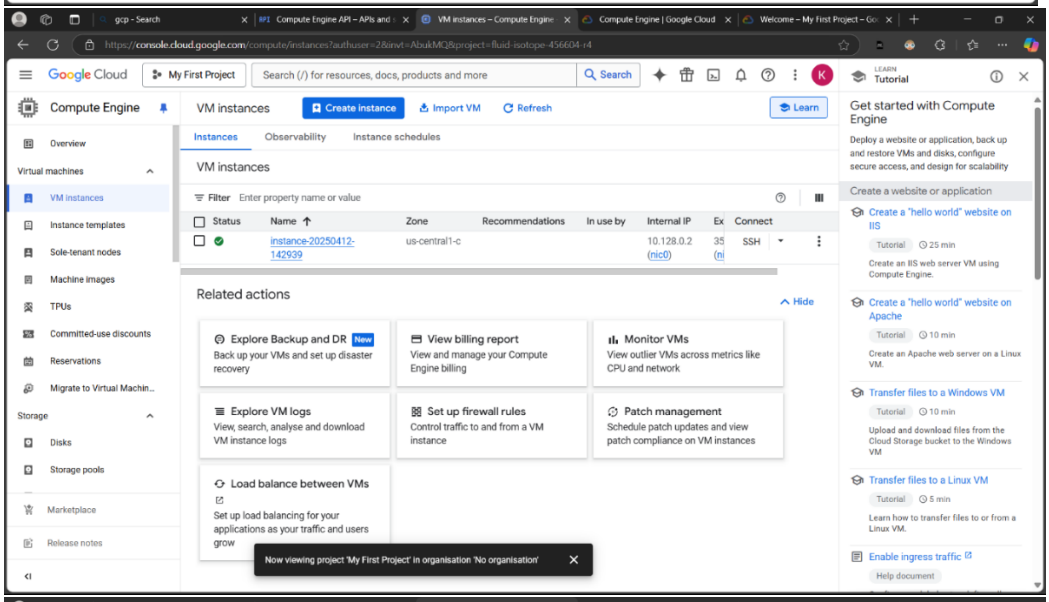
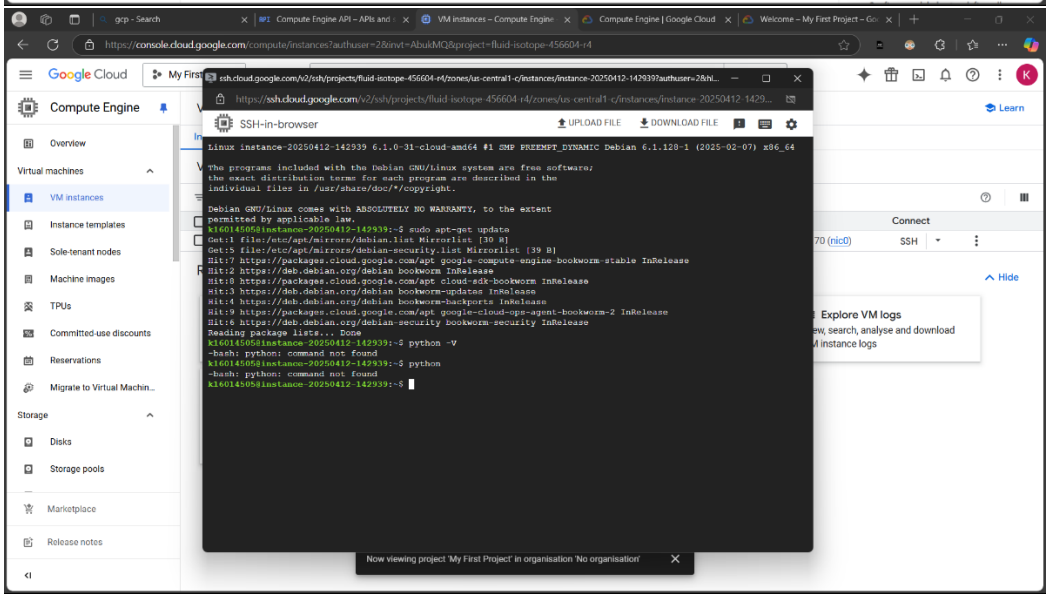
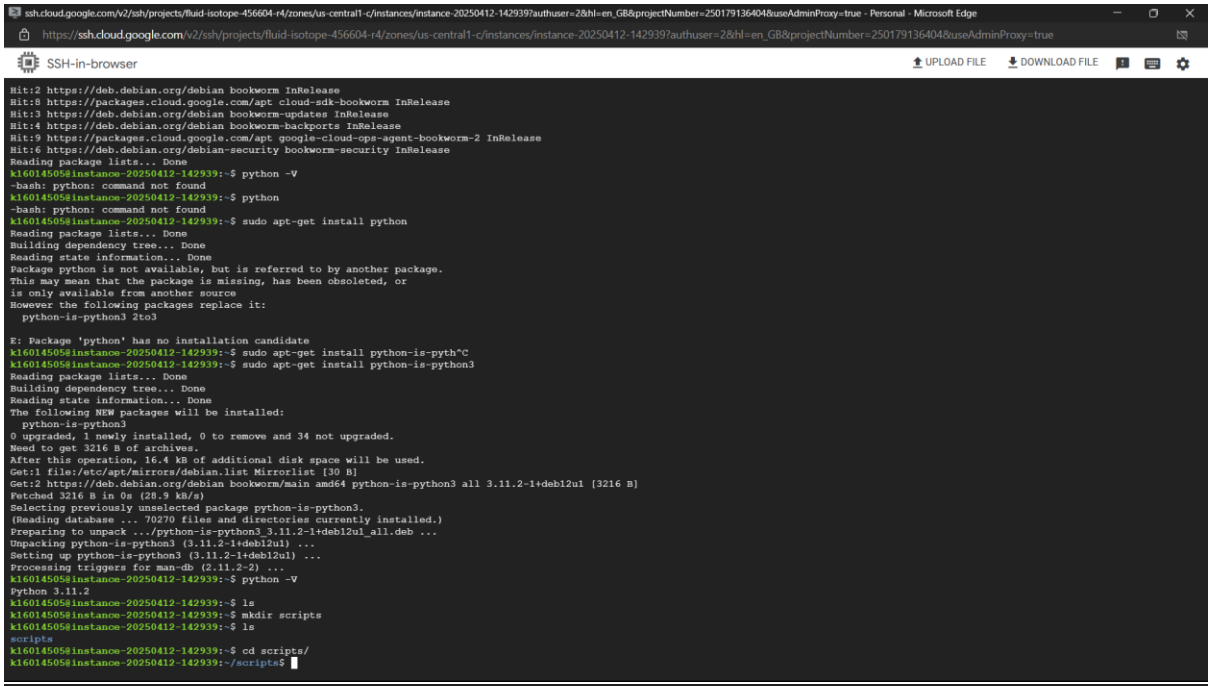
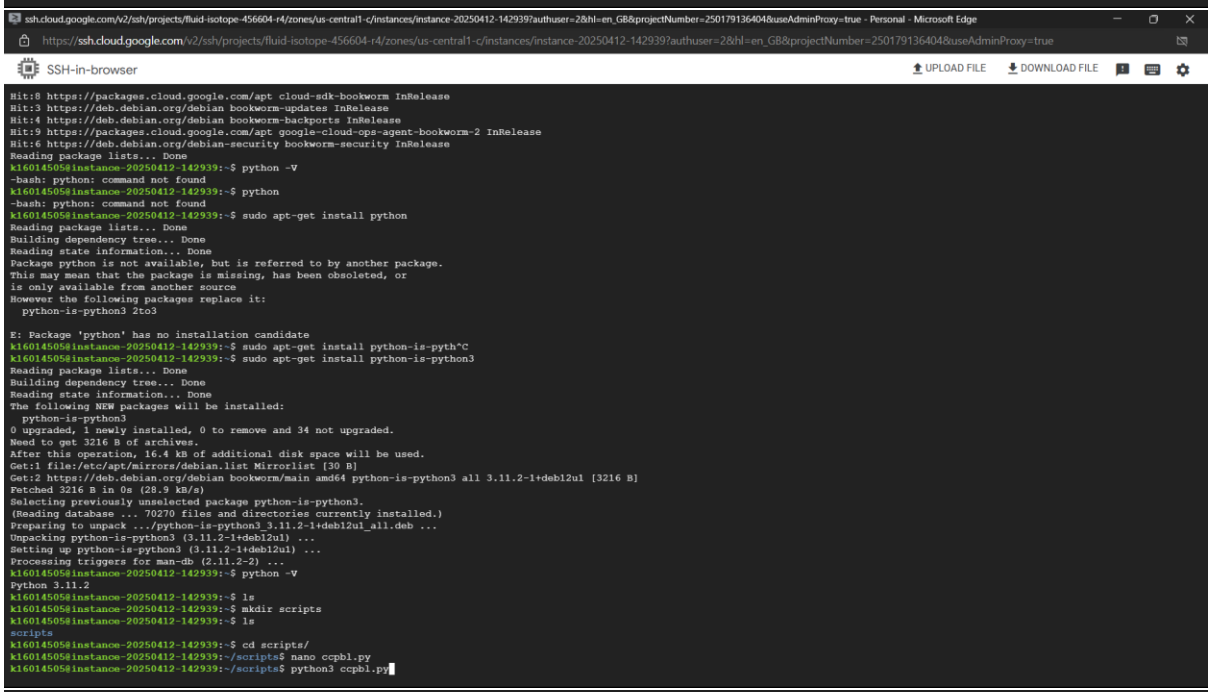
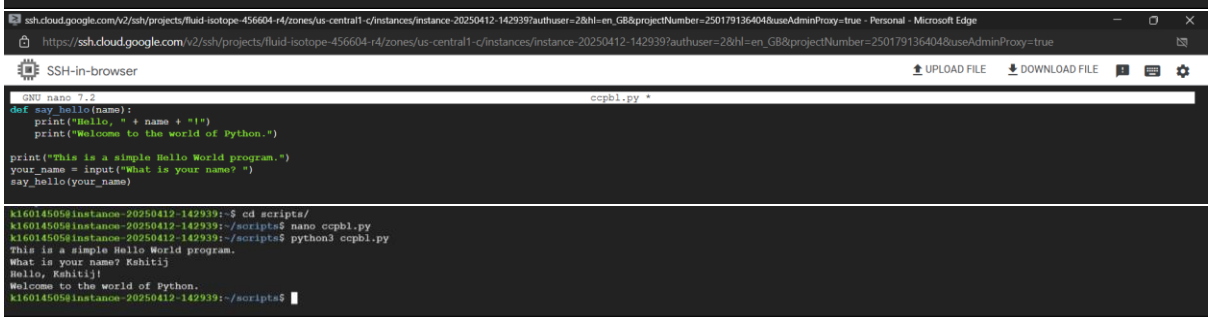

Free trial status: ₹25,735.88 credit and 91 days remaining. Activate your full account to get unlimited access to all of Google Cloud—use any remaining credits, then pay only for what you use.
Dismiss
Activate
Google Cloud
My First Project
Search (/) for resources, docs, products, and more
Search
You can now search for documentation, resource metadata, tutorials, and API keys
Welcome, K Sk
You're in Free Trial
0 out of ₹25,736 credits used
Expires July 12, 2025
What happens when trial ends?
Activate full account
You're working on project My First Project
Number: 250179136404 ID: fluid-isotope-456604-r4
Add people to your project
Set up budget alerts
Review product spend
Try Gemini 2.0 Fla
Try Gemini
K Sk
k16014505@gmail.com
Privacy
Google Account
Kshitij
kshitij.kamdlz@gmail.com
Kshitij Kamdi
kshitijkamdi39@gmail.com
Add account
Sign out
Recommended based on your interest in General
Products
Create a VM
Compute Engine
Train and host ML models
Vertex AI
Create a database
Cloud SQL
Analyze and manage data
BigQuery

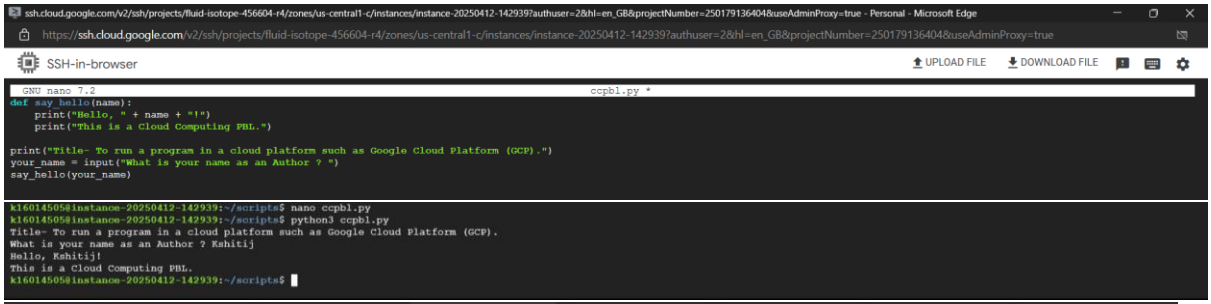

Compute Engine
VM instances
Create instance
Import VM
Refresh
Learn
Instances
Observability
Instance schedules
VM instances
Filter Enter property name or value
Status Name Zone Recommendations In use by Internal IP Ex Connect
instance-20250412-142939 us-central1-c 10.128.0.2 (nic0) SSH
Related actions
Hide
Explore Backup and DR New
Back up your VMs and set up disaster recovery
View billing report
View and manage your Compute Engine billing
Monitor VMs
View outlier VMs across metrics like CPU and network
Explore VM logs
View, search, analyse and download VM instance logs
Set up firewall rules
Control traffic to and from a VM instance
Patch management
Schedule patch updates and view patch compliance on VM instances
Load balance between VMs
Set up load balancing for your applications as your traffic and users grow
Now viewing project 'My First Project' in organisation 'No organisation'
Tutorial
Get started with Compute Engine
Deploy a website or application, back up and restore VMs and disks, configure secure access, and design for scalability
Create a website or application
Create a "hello world" website on IIS
Tutorial 25 min
Create an IIS web server VM using Compute Engine.
Create a "hello world" website on Apache
Tutorial 10 min
Create an Apache web server on a Linux VM.
Transfer files to a Windows VM
Tutorial 10 min
Upload and download files from the Cloud Storage bucket to the Windows VM
Transfer files to a Linux VM
Tutorial 5 min
Learn how to transfer files to or from a Linux VM.
Enable ingress traffic
Help document

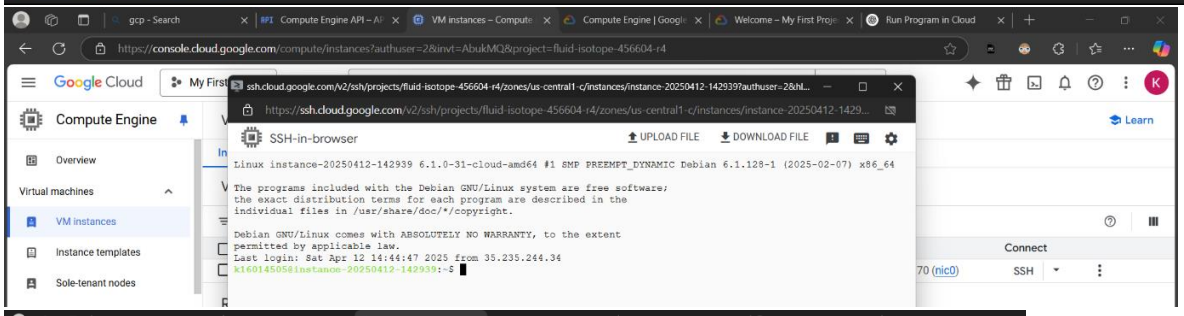

SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
Linux instance-20250412-142939 6.1.0-31-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.128-1 (2025-02-07) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
k16014505@instance-20250412-142939:~$ sudo apt-get update
Hit:1 file:/etc/apt/mirrors/debian.list Mirrorlist [30 B]
Get:5 file:/etc/apt/mirrors/debian-security.list Mirrorlist [39 B]
Hit:7 https://packages.cloud.google.com/apt google-compute-engine-bookworm-stable InRelease
Hit:2 https://deb.debian.org/debian bookworm InRelease
Hit:8 https://packages.cloud.google.com/apt cloud-sdk-bookworm InRelease
Hit:3 https://deb.debian.org/debian bookworm-updates InRelease
Hit:4 https://deb.debian.org/debian bookworm-backports InRelease
Hit:9 https://packages.cloud.google.com/apt google-cloud-ops-agent-bookworm-2 InRelease
Hit:6 https://deb.debian.org/debian-security bookworm-security InRelease
Reading package lists... Done
k16014505@instance-20250412-142939:~$ python -V
-bash: python: command not found
k16014505@instance-20250412-142939:~$ python
-bash: python: command not found
k16014505@instance-20250412-142939:~$
Now viewing project 'My First Project' in organisation 'No organisation'
Explore VM logs

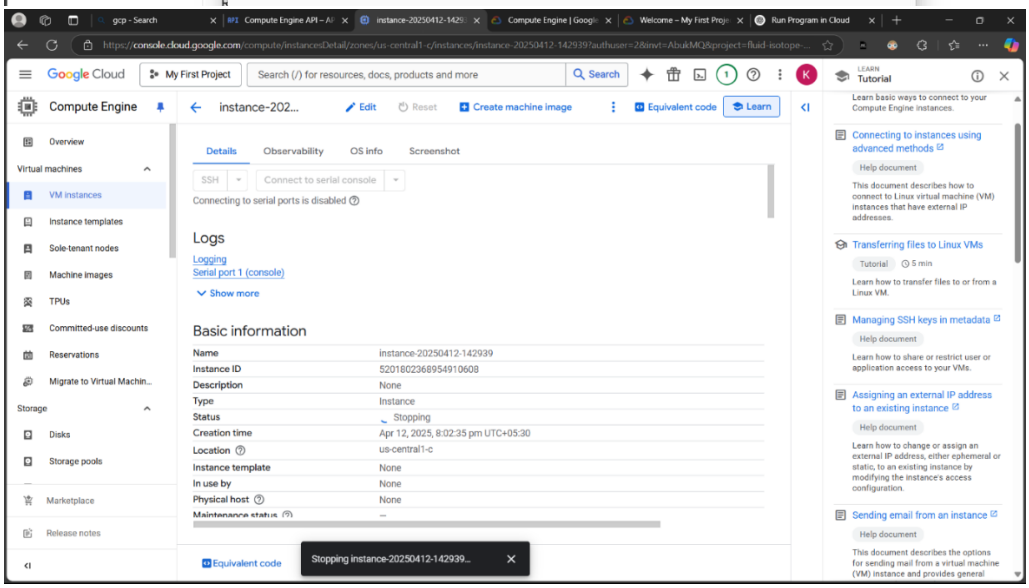

ssh.cloud.google.com/v2/ssh/projects/fluid-isotope-456604-r4/zones/us-central1-c/instances/instance-20250412-142939?authuser=2&hl=en_GB&projectNumber=250179136404&useAdminProxy=true - Personal - Microsoft Edge
https://ssh.cloud.google.com/v2/ssh/projects/fluid-isotope-456604-r4/zones/us-central1-c/instances/instance-20250412-142939?authuser=2&hl=en_GB&projectNumber=250179136404&useAdminProxy=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
Hit:2 https://deb.debian.org/debian bookworm InRelease
Hit:8 https://packages.cloud.google.com/apt cloud-sdk-bookworm InRelease
Hit:3 https://deb.debian.org/debian bookworm-updates InRelease
Hit:4 https://deb.debian.org/debian bookworm-backports InRelease
Hit:9 https://packages.cloud.google.com/apt google-cloud-ops-agent-bookworm-2 InRelease
Hit:6 https://deb.debian.org/debian-security bookworm-security InRelease
Reading package lists... Done
k16014505@instance-20250412-142939:~$ python -V
-bash: python: command not found
k16014505@instance-20250412-142939:~$ python
-bash: python: command not found
k16014505@instance-20250412-142939:~$ sudo apt-get install python
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package python is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
However the following packages replace it:
  python-is-python3 2to3
E: Package 'python' has no installation candidate
k16014505@instance-20250412-142939:~$ sudo apt-get install python-is-pyth^C
k16014505@instance-20250412-142939:~$ sudo apt-get install python-is-python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  python-is-python3
0 upgraded, 1 newly installed, 0 to remove and 34 not upgraded.
Need to get 3216 B of archives.
After this operation, 16.4 kB of additional disk space will be used.
Get:1 file:/etc/apt/mirrors/debian.list Mirrorlist [30 B]
Get:2 https://deb.debian.org/debian bookworm/main amd64 python-is-python3 all 3.11.2-1+deb12u1 [3216 B]
Fetched 3216 B in 0s (28.9 kB/s)
Selecting previously unselected package python-is-python3.
(Reading database ... 70270 files and directories currently installed.)
Preparing to unpack .../python-is-python3_3.11.2-1+deb12u1_all.deb ...
Unpacking python-is-python3 (3.11.2-1+deb12u1) ...
Setting up python-is-python3 (3.11.2-1+deb12u1) ...
Processing triggers for man-db (2.11.2-2) ...
k16014505@instance-20250412-142939:~$ python -V
Python 3.11.2
k16014505@instance-20250412-142939:~$ ls
k16014505@instance-20250412-142939:~$ mkdir scripts
k16014505@instance-20250412-142939:~$ ls
scripts
k16014505@instance-20250412-142939:~$ cd scripts/
k16014505@instance-20250412-142939:~/scripts$

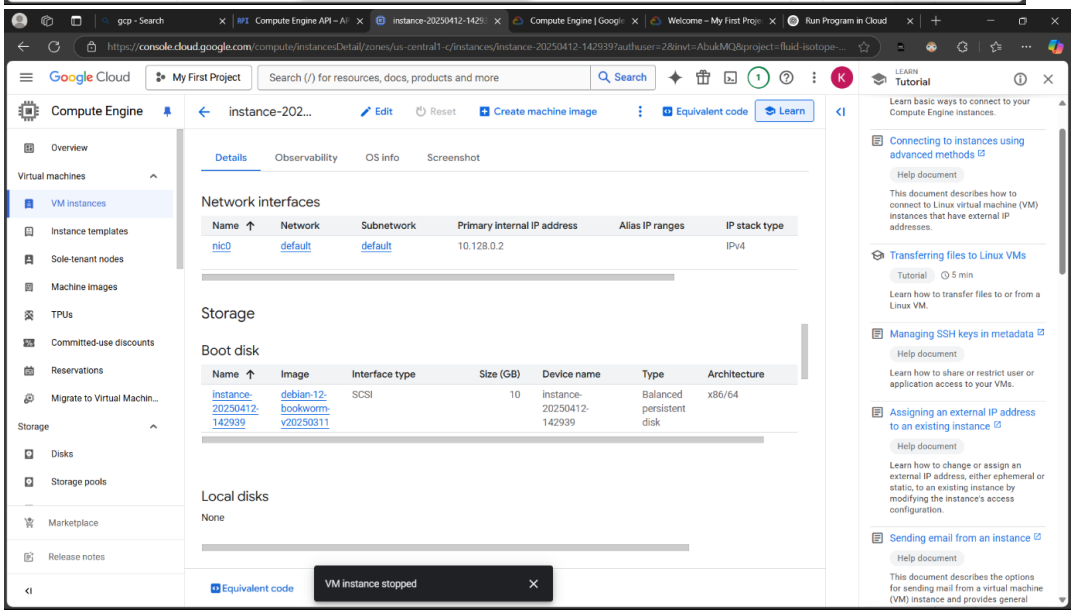

ssh.cloud.google.com/v2/ssh/projects/fluid-isotope-456604-r4/zones/us-central1-c/instances/instance-20250412-142939?authuser=2&hl=en_GB&projectNumber=250179136404&useAdminProxy=true - Personal - Microsoft Edge
https://ssh.cloud.google.com/v2/ssh/projects/fluid-isotope-456604-r4/zones/us-central1-c/instances/instance-20250412-142939?authuser=2&hl=en_GB&projectNumber=250179136404&useAdminProxy=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
Hit:8 https://packages.cloud.google.com/apt cloud-sdk-bookworm InRelease
Hit:3 https://deb.debian.org/debian bookworm-updates InRelease
Hit:4 https://deb.debian.org/debian bookworm-backports InRelease
Hit:9 https://packages.cloud.google.com/apt google-cloud-ops-agent-bookworm-2 InRelease
Hit:6 https://deb.debian.org/debian-security bookworm-security InRelease
Reading package lists... Done
k16014505@instance-20250412-142939:~$ python -V
-bash: python: command not found
k16014505@instance-20250412-142939:~$ python
-bash: python: command not found
k16014505@instance-20250412-142939:~$ sudo apt-get install python
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package python is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
However the following packages replace it:
  python-is-python3 2to3
E: Package 'python' has no installation candidate
k16014505@instance-20250412-142939:~$ sudo apt-get install python-is-pyth^C
k16014505@instance-20250412-142939:~$ sudo apt-get install python-is-python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  python-is-python3
0 upgraded, 1 newly installed, 0 to remove and 34 not upgraded.
Need to get 3216 B of archives.
After this operation, 16.4 kB of additional disk space will be used.
Get:1 file:/etc/apt/mirrors/debian.list Mirrorlist [30 B]
Get:2 https://deb.debian.org/debian bookworm/main amd64 python-is-python3 all 3.11.2-1+deb12u1 [3216 B]
Fetched 3216 B in 0s (28.9 kB/s)
Selecting previously unselected package python-is-python3.
(Reading database ... 70270 files and directories currently installed.)
Preparing to unpack .../python-is-python3_3.11.2-1+deb12u1_all.deb ...
Unpacking python-is-python3 (3.11.2-1+deb12u1) ...
Setting up python-is-python3 (3.11.2-1+deb12u1) ...
Processing triggers for man-db (2.11.2-2) ...
k16014505@instance-20250412-142939:~$ python -V
Python 3.11.2
k16014505@instance-20250412-142939:~$ ls
k16014505@instance-20250412-142939:~$ mkdir scripts
k16014505@instance-20250412-142939:~$ ls
scripts
k16014505@instance-20250412-142939:~$ cd scripts/
k16014505@instance-20250412-142939:~/scripts$ nano ccpb1.py
k16014505@instance-20250412-142939:~/scripts$ python3 ccpb1.py

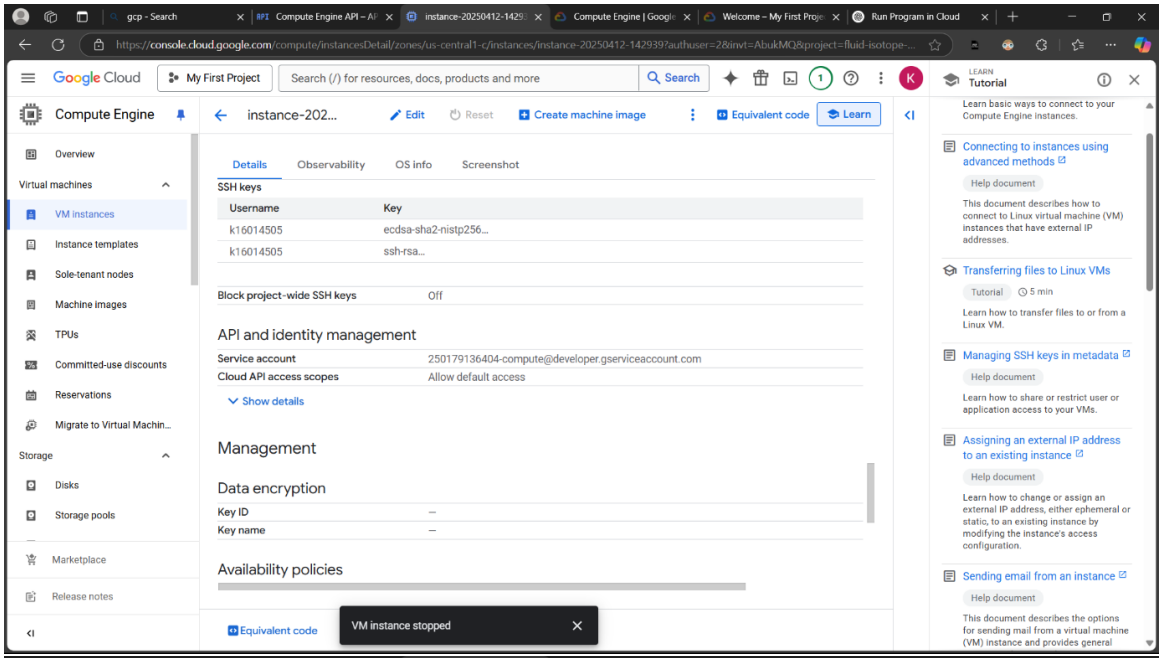

ssh.cloud.google.com/v2/ssh/projects/fluid-isotope-456604-r4/zones/us-central1-c/instances/instance-20250412-142939?authuser=2&hl=en_GB&projectNumber=250179136404&useAdminProxy=true - Personal - Microsoft Edge
https://ssh.cloud.google.com/v2/ssh/projects/fluid-isotope-456604-r4/zones/us-central1-c/instances/instance-20250412-142939?authuser=2&hl=en_GB&projectNumber=250179136404&useAdminProxy=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
GNU nano 7.2 ccpb1.py *
def say_hello(name):
print("Hello, " + name + "!")
print("Welcome to the world of Python.")
print("This is a simple Hello World program.")
your_name = input("What is your name? ")
say_hello(your_name)
k16014505@instance-20250412-142939:~$ cd scripts/
k16014505@instance-20250412-142939:~/scripts$ nano ccpb1.py
k16014505@instance-20250412-142939:~/scripts$ python3 ccpb1.py
This is a simple Hello World program.
What is your name? Kshitij
Hello, Kshitij!
Welcome to the world of Python.
k16014505@instance-20250412-142939:~/scripts$

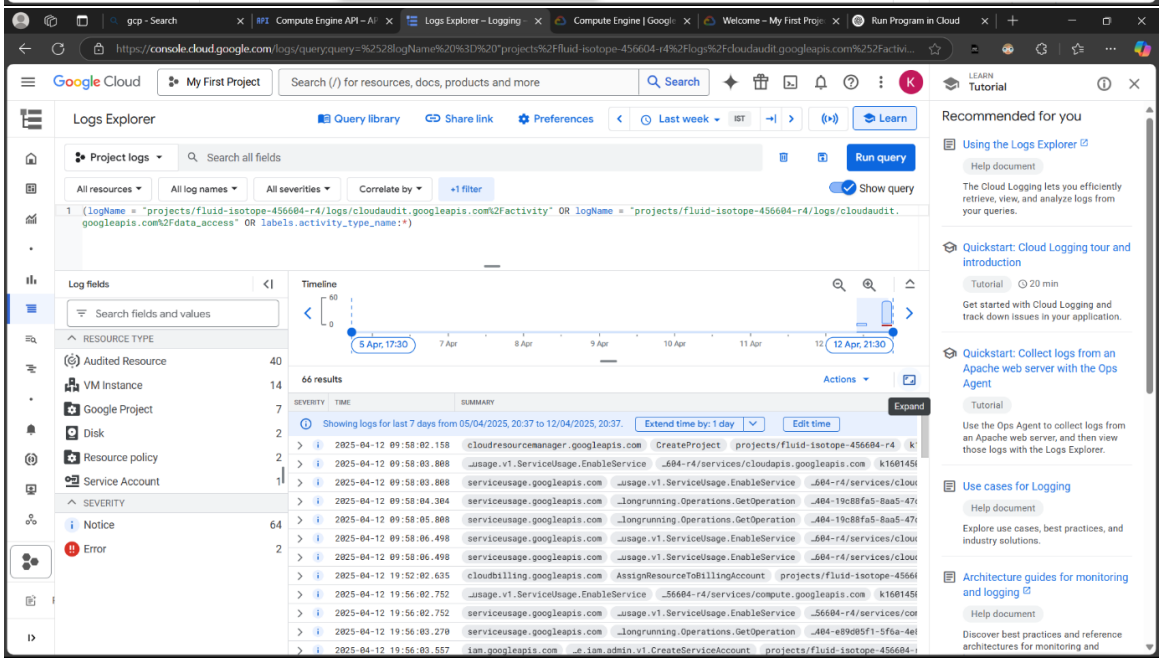

ssh.cloud.google.com/v2/ssh/projects/fluid-isotope-456604-r4/zones/us-central1-c/instances/instance-20250412-142939?authuser=2&hl=en_GB&projectNumber=250179136404&useAdminProxy=true - Personal - Microsoft Edge
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
GNU nano 7.2 ccpbl.py *
def say_hello(name):
print("Hello, " + name + "!")
print("This is a Cloud Computing PBL.")
print("Title- To run a program in a cloud platform such as Google Cloud Platform (GCP).")
your_name = input("What is your name as an Author ? ")
say_hello(your_name)
k16014505@instance-20250412-142939:~/scripts$ nano ccpbl.py
k16014505@instance-20250412-142939:~/scripts$ python3 ccpbl.py
Title- To run a program in a cloud platform such as Google Cloud Platform (GCP).
What is your name as an Author ? Kshitij
Hello, Kshitij!
This is a Cloud Computing PBL.
k16014505@instance-20250412-142939:~/scripts$


Google Cloud
Compute Engine
Overview
Virtual machines
VM instances
Instance templates
Sole-tenant nodes
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
Linux instance-20250412-142939 6.1.0-31-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.128-1 (2025-02-07) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Apr 12 14:44:47 2025 from 35.235.244.34
k16014505@instance-20250412-142939:~$
Connect
SSH


Google Cloud
My First Project
Search (/) for resources, docs, products and more
Search
Compute Engine
instance-202...
Edit
Reset
Create machine image
Equivalent code
Learn
Details
Observability
OS info
Screenshot
SSH
Connect to serial console
Connecting to serial ports is disabled
Logs
Logging
Serial port 1 (console)
Show more
Basic information
Name instance-20250412-142939
Instance ID 5201802368954910608
Description None
Type Instance
Status Stopping
Creation time Apr 12, 2025, 8:02:35 pm UTC+05:30
Location us-central1-c
Instance template None
In use by None
Physical host None
Stopping instance-20250412-142939...
Tutorial
Connecting to instances using advanced methods
Transferring files to Linux VMs
Managing SSH keys in metadata
Assigning an external IP address to an existing instance
Sending email from an instance


Google Cloud
My First Project
Search (/) for resources, docs, products and more
Search
Compute Engine
instance-202...
Edit
Reset
Create machine image
Equivalent code
Learn
Details
Observability
OS info
Screenshot
Network interfaces
Name Network Subnetwork Primary internal IP address Alias IP ranges IP stack type
nic0 default default 10.128.0.2 IPv4
Storage
Boot disk
Name Image Interface type Size (GB) Device name Type Architecture
instance-20250412-142939 debian-12-bookworm-v20250311 SCSI 10 instance-20250412-142939 Balanced persistent disk x86/64
Local disks
None
VM instance stopped
Tutorial
Connecting to instances using advanced methods
Transferring files to Linux VMs
Managing SSH keys in metadata
Assigning an external IP address to an existing instance
Sending email from an instance


Google Cloud
My First Project
Search (/) for resources, docs, products and more
Search
Compute Engine
instance-202...
Edit
Reset
Create machine image
Equivalent code
Learn
Overview
Virtual machines
VM instances
Instance templates
Sole-tenant nodes
Machine images
TPUs
Committed-use discounts
Reservations
Migrate to Virtual Machin...
Storage
Disks
Storage pools
Marketplace
Release notes
Details
Observability
OS info
Screenshot
SSH keys
Username
Key
k16014505
ecdsa-sha2-nistp256...
k16014505
ssh-rsa...
Block project-wide SSH keys
Off
API and identity management
Service account
250179136404-compute@developer.gserviceaccount.com
Cloud API access scopes
Allow default access
Show details
Management
Data encryption
Key ID
Key name
Availability policies
Equivalent code
VM instance stopped
LEARN
Tutorial
Learn basic ways to connect to your Compute Engine instances.
Connecting to instances using advanced methods
Help document
This document describes how to connect to Linux virtual machine (VM) instances that have external IP addresses.
Transferring files to Linux VMs
Tutorial
5 min
Learn how to transfer files to or from a Linux VM.
Managing SSH keys in metadata
Help document
Learn how to share or restrict user or application access to your VMs.
Assigning an external IP address to an existing instance
Help document
Learn how to change or assign an external IP address, either ephemeral or static, to an existing instance by modifying the instance's access configuration.
Sending email from an instance
Help document
This document describes the options for sending mail from a virtual machine (VM) instance and provides general


Google Cloud
My First Project
Search (/) for resources, docs, products and more
Search
Logs Explorer
Query library
Share link
Preferences
Last week
Learn
Project logs
Search all fields
Run query
All resources
All log names
All severities
Correlate by
+1 filter
Show query
(logName = "projects/fluid-isotope-456604-r4/logs/cloudaudit.googleapis.com%2Factivity" OR logName = "projects/fluid-isotope-456604-r4/logs/cloudaudit.googleapis.com%2Fdata_access" OR labels.activity_type_name:*)
Log fields
Search fields and values
RESOURCE TYPE
Audited Resource 40
VM Instance 14
Google Project 7
Disk 2
Resource policy 2
Service Account 1
SEVERITY
Notice 64
Error 2
Timeline
5 Apr, 17:30
12 Apr, 21:30
66 results
Actions
Expand
Showing logs for last 7 days from 05/04/2025, 20:37 to 12/04/2025, 20:37.
Extend time by: 1 day
Edit time
Tutorial
Recommended for you
Using the Logs Explorer
Help document
The Cloud Logging lets you efficiently retrieve, view, and analyze logs from your queries.
Quickstart: Cloud Logging tour and introduction
Tutorial
20 min
Get started with Cloud Logging and track down issues in your application.
Quickstart: Collect logs from an Apache web server with the Ops Agent
Tutorial
Use the Ops Agent to collect logs from an Apache web server, and then view those logs with the Logs Explorer.
Use cases for Logging
Help document
Explore use cases, best practices, and industry solutions.
Architecture guides for monitoring and logging
Help document
Discover best practices and reference architectures for monitoring and

1. Introduction

In the modern era of cloud computing, organizations and developers seek scalable, flexible, and reliable platforms to host, develop, and manage applications. **Google Cloud Platform (GCP)**, developed by Google, is one of the leading cloud service providers globally, alongside Amazon Web Services (AWS) and Microsoft Azure.

GCP is a **comprehensive suite of cloud computing services** that operates on the same global infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, YouTube, and Google Drive. It provides a broad range of services including computing, data storage, data analytics, networking, machine learning, artificial intelligence, and more.

Key Features of GCP:

- **Global Infrastructure**: GCP operates in multiple regions and zones across the world, ensuring high availability, redundancy, and performance.
- **Scalability**: Services can scale up or down automatically depending on the application's load and traffic.
- **Security**: GCP offers enterprise-grade security measures, including encryption at rest and in transit, identity and access management, and compliance with global standards.
- **Pay-as-You-Go Pricing**: Users pay only for the resources they consume, making it cost-effective for both startups and large enterprises.

2. Objectives

The goal of this project report is to provide a **comprehensive, step-by-step conceptual understanding** of how to run a program on the Google Cloud Platform (GCP). Since GCP is a vast ecosystem of services and tools, it is important to clearly define the core learning objectives that will guide this exploration.

2.1 Understand the Basic Structure of GCP

To effectively use GCP, it is essential to grasp the foundational architecture and organization of the platform. GCP is structured into:

- **Projects**: The fundamental organizational unit in GCP. All resources, APIs, and services are linked to a specific project.
- **Regions and Zones**: These are geographical locations where services are hosted. Understanding how to choose the appropriate region/zone impacts latency, availability, and cost.
- **Resources**: These include virtual machines, storage buckets, databases, and more, which are provisioned and managed within a project.
- **Billing and IAM (Identity and Access Management)**: GCP projects are tied to billing accounts, and user access is controlled through IAM roles and permissions.

2.2 Learn How to Prepare and Deploy a Program on GCP

A key objective is to guide users through the **logical sequence of preparing, uploading, and deploying a program** on the GCP infrastructure. This includes:

- Setting up the development environment and necessary tools (such as the Google Cloud Console, Cloud Shell, and SDK).
- Choosing the right compute service (e.g., Compute Engine for virtual machines, App Engine for serverless apps, Cloud Run for containerized services).
- Uploading the application files or containers to the cloud.
- Managing deployment configurations such as environment variables, memory/CPU allocations, and auto-scaling policies.

2.3 Identify the Key Services Involved in Running a Cloud-Based Application

GCP provides a variety of services categorized under compute, storage, networking, and more. This report aims to help users **recognize and understand the purpose of major services** such as:

- **Compute Engine**: For running applications on virtual machines.
- **App Engine**: For automatically managed applications using supported runtimes.

- **Cloud Functions**: For event-driven, serverless applications.
- **Cloud Run**: For deploying containerized applications that scale automatically.
- **Cloud Storage**: For storing static files and application data.
- **Cloud Monitoring & Logging**: For observing the performance and health of deployed services.

2.4 Outline the Steps from Project Creation to Execution

A final and crucial objective is to outline a **clear, high-level roadmap** of the entire process, from creating a new GCP project to running a deployed program. These steps include:

1. **Creating a Google Cloud account and project.**
2. **Enabling billing and required APIs.**
3. **Setting up IAM roles and permissions.**
4. **Selecting appropriate compute and storage services.**
5. **Uploading and deploying the application.**
6. **Testing and monitoring the live application.**
7. **Managing updates, scaling, and security post-deployment.**

3. Overview

Google Cloud Platform (GCP) is designed to provide a **robust, flexible, and secure cloud environment** that supports a wide range of applications—from small websites to enterprise-level, globally distributed systems. At its core, GCP follows a modular architecture, offering different services that can be combined and scaled based on specific application needs.

GCP's architecture is built on the same infrastructure that powers Google's own services (like Gmail, YouTube, and Google Search), ensuring reliability, high availability, and performance at a global scale.

Below are the **four major components** of GCP's architecture:

3.1 Compute Services

Compute services are the backbone of any cloud platform. They provide the **computing power** needed to run programs, applications, and services. GCP offers multiple compute options to cater to different use cases:

- **Compute Engine (IaaS)**
    - Allows you to create and run virtual machines (VMs) on demand.
    - Provides complete control over the operating system, software stack, and configurations.
    - Ideal for custom environments, legacy application migration, and intensive compute tasks.
- **App Engine (PaaS)**
    - A fully managed serverless platform for developing and hosting applications.
    - Automatically handles infrastructure, scaling, patching, and load balancing.
    - Developers only need to write code and deploy—it abstracts away system administration.
- **Cloud Functions (FaaS)**
    - Lightweight, event-driven serverless compute platform.
    - Executes single-purpose functions in response to events from services like HTTP requests, Cloud Storage uploads, or database changes.
    - Great for automation, data processing, and backend tasks.
- **Cloud Run**
    - A serverless compute platform for running containerized applications.
    - Offers the flexibility of containers with the simplicity of serverless.
    - Automatically scales up or down based on request volume.

3.2 Storage Services

Storage is essential for managing application data, files, backups, and logs. GCP offers diverse storage options to meet various needs, from unstructured object storage to transactional databases:

- **Cloud Storage**
    - Object storage designed for unstructured data such as images, videos, and backups.
    - Offers multiple storage classes (Standard, Nearline, Coldline, Archive) based on frequency of access and cost.
    - Highly durable (11 nines of durability) and globally available.
- **Cloud Firestore**
    - A scalable NoSQL document database for real-time applications.
    - Supports automatic synchronization across devices and offline capabilities.
    - Ideal for mobile, web, and IoT applications.
- **Cloud SQL and Cloud Spanner**
    - Cloud SQL: Managed relational database service supporting MySQL, PostgreSQL, and SQL Server.
    - Cloud Spanner: Globally distributed relational database combining traditional RDBMS features with horizontal scalability.
- **Persistent Disks**
    - Block storage used with Compute Engine VMs.
    - Can be zonal or regional for redundancy and performance.

3.3 Networking Services

Networking in GCP connects all the components of your application—users, databases, services, and external systems. GCP provides a **global, high-performance network infrastructure**, enabling secure and fast communication.

- **Virtual Private Cloud (VPC)**
    - A private network space in GCP where you can launch your compute resources.
    - Provides IP allocation, subnets, firewalls, and routing control.
    - Supports hybrid cloud setups and inter-region communication.
- **Cloud Load Balancing**
    - Distributes incoming traffic across multiple instances or regions.

    - Ensures high availability and performance by routing requests to the closest and healthiest backend.
    - Supports HTTP(S), TCP/SSL, and UDP load balancing.
- **Cloud CDN (Content Delivery Network)**
    - Delivers static and dynamic content closer to users using Google's globally distributed edge locations.
    - Reduces latency and improves user experience.
- **Cloud Interconnect and VPN**
    - Connect on-premises networks to GCP through secure, high-bandwidth connections.

3.4 Identity and Access Management (IAM)

Security is a foundational element of cloud architecture. GCP's **Identity and Access Management (IAM)** system controls who (users or services) can access what resources and how.

- **Granular Permissions**
    - Roles can be assigned at the project, folder, or resource level.
    - Predefined roles (Viewer, Editor, Owner) and custom roles can be created.
- **Service Accounts**
    - Special Google accounts used by applications or virtual machines to interact with GCP services securely.
    - Prevents unauthorized access and ensures scoped privileges.
- **Multi-Factor Authentication (MFA)**
    - Additional security layer for user accounts.
- **Audit Logging**
    - Tracks user and service activities to ensure transparency, compliance, and troubleshooting.

4. GCP Services

Google Cloud Platform offers multiple services that enable users to run applications in the cloud. Each service provides a different level of abstraction, control, and automation depending on the nature and requirements of the program being deployed. From full control with virtual machines to fully serverless architectures, GCP provides developers with the flexibility to choose the right tool for the job.

Below is a detailed overview of four primary GCP services commonly used to run programs:

4.1 Compute Engine

Type: Infrastructure-as-a-Service (IaaS)
Best For: Full control over infrastructure and complex workloads

Google Compute Engine (GCE) is a virtual machine (VM) service that provides users with highly customizable compute infrastructure. With Compute Engine, users can **launch and manage virtual machines** that run on Google's infrastructure, similar to how they would use a physical server.

Key Features:

- **Customizable VM instances**: Choose machine type, CPU, memory, GPU, and disk configurations.
- **Operating System Flexibility**: Supports various OS images including Linux, Windows, and custom images.
- **Persistent Disks and Snapshots**: Reliable data storage that can be attached or detached from VMs.
- **Scalability and Load Balancing**: Supports auto-scaling and global load balancing for large-scale applications.
- **Preemptible VMs**: Cost-effective options for batch jobs or short-lived workloads.

Use Cases:

- Running legacy applications or software that require specific configurations.
- Hosting large-scale enterprise applications like databases and analytics engines.
- High-performance computing and machine learning workloads.

4.2 App Engine

Type: Platform-as-a-Service (PaaS)
Best For: Rapid application deployment with minimal infrastructure management

Google App Engine provides a fully managed environment for developing and hosting applications. It allows developers to focus solely on writing code, while GCP **automatically handles infrastructure concerns** such as provisioning servers, scaling, and load balancing.

Key Features:

- **Automatic Scaling**: Automatically adjusts computing resources based on traffic.
- **Multiple Language Support**: Supports popular languages such as Python, Java, Node.js, Go, PHP, and Ruby.
- **Integrated Developer Tools**: Seamless deployment from the command line or CI/CD pipelines.
- **Standard and Flexible Environments**:
    - **Standard**: Uses pre-configured runtimes with fast scaling and lower costs.
    - **Flexible**: Allows use of custom runtimes in Docker containers.

Use Cases:

- Web and mobile backend applications.
- APIs and microservices that require auto-scaling.
- Startups and developers who need to deploy applications quickly without managing servers.

4.3 Cloud Functions

Type: Function-as-a-Service (FaaS) / Serverless Computing
Best For: Lightweight event-driven applications and microservices

Google Cloud Functions is a **serverless compute service** that runs small pieces of code (functions) in response to events. Developers do not need to provision or manage any infrastructure—GCP automatically allocates resources as needed.

Key Features:

- **Event-driven execution**: Triggered by HTTP requests, file uploads, database changes, etc.
- **Supports multiple runtimes**: JavaScript (Node.js), Python, Go, Java, etc.
- **Pay-per-use model**: Charges only for the time the function runs.
- **Integrated with other GCP services**: Easily responds to events from Cloud Storage, Pub/Sub, Firestore, and more.

Use Cases:

- Backend logic for web or mobile apps.
- Processing data from cloud storage or IoT devices.

- Lightweight APIs and automation tasks.

4.4 Cloud Run

Type: Serverless Containers
Best For: Deploying and managing containerized applications with ease

Google Cloud Run is a **serverless platform** designed to run **containers**. It gives developers the flexibility to build applications in any language and framework, package them into containers, and deploy them with a **fully managed, auto-scaling service**.

Key Features:

- **Deploy directly from container images** stored in Google Container Registry or Artifact Registry.
- **Stateless request handling** with automatic scaling from zero to thousands of instances.
- **Custom environments**: Since containers are used, developers have full control over the runtime environment.
- **Concurrency control**: Configure how many requests each instance handles.

Use Cases:

- APIs or web services packaged in containers.
- Applications requiring specific software or runtime dependencies.
- Migrating Docker-based applications to the cloud with minimal changes.Each of these services offers unique advantages depending on the project requirements. Understanding their capabilities helps users choose the most suitable environment to run their applications effectively on GCP.

5. Process

Deploying a program on Google Cloud Platform (GCP) involves several sequential steps, from account setup to monitoring the running application. This structured workflow ensures that your program runs efficiently and securely in the cloud environment. Below is a comprehensive step-by-step guide, free from code, to understand how a program can be prepared, deployed, and executed on GCP.

Step 1: Create a Google Cloud Account

Before using any of GCP's services, you need to create an account:

- **Visit** https://cloud.google.com.
- **Sign in** using your Google credentials or create a new Google account.
- **Set up billing**: Add a valid billing method to activate your account. GCP typically provides a **free trial with $300 credits** for new users.
- **Accept terms and conditions** and proceed to the Cloud Console.

Step 2: Create a New Project

Projects are the **primary containers** for all your GCP resources:

- Go to the **GCP Console Dashboard**.
- Click on the **project selector dropdown** (top menu bar) and click **"New Project."**
- Enter the following:
    - **Project Name** (e.g., my-cloud-app)
    - **Billing Account** (choose one if already linked)
    - **Organization** (optional for personal accounts)
- Click **"Create"** to initialize your project.

Step 3: Enable APIs and Services

To use any GCP service (e.g., Compute Engine or App Engine), the associated APIs must be enabled.

- In the navigation menu, go to **APIs & Services > Library**.
- Search and **enable the APIs** needed for your application. Common ones include:
    - **Compute Engine API**
    - **App Engine Admin API**

    - **Cloud Functions API**
    - **Cloud Run API**
- Repeat for other services based on your application's requirements.

Step 4: Set Up IAM & Admin Permissions

Identity and Access Management (IAM) ensures only authorized users can access specific resources.

- Navigate to **IAM & Admin > IAM** in the menu.
- Click **"Add"** to include new users or team members.
- Assign appropriate **roles** based on the principle of least privilege. For example:
    - **Viewer**: View resources without editing.
    - **Editor**: Create and modify resources.
    - **Owner**: Full administrative control.
- Ensure you (or your team) have at least **Editor** or **Owner** role to manage deployment and configurations.

Step 5: Prepare the Environment

Choose the most suitable GCP service depending on your application's structure and requirements:

- **App Engine**:
    - For applications written in supported languages like Python, Java, or Node.js.
    - Best when you want automatic scaling and minimal configuration.
- **Compute Engine**:
    - For full control over OS, networking, and software.
    - Ideal for legacy applications or workloads needing custom configurations.
- **Cloud Run**:
    - For containerized applications built using Docker.
    - Offers the flexibility of containers with the ease of serverless deployment.

Step 6: Upload Your Program

Once the environment is chosen, the program must be uploaded for execution:

- **App Engine**:
    - Create an App Engine application in your selected region.

    - Upload program files using either the **Cloud Console** or the **Cloud SDK**.
- **Compute Engine**:
    - Launch a **VM instance** and connect via **SSH**.
    - Upload files using:
        - Direct command-line file transfer (gcloud compute scp)
        - Upload via **Cloud Storage** and then access within the VM.
- **Cloud Run**:
    - Package your application in a **Docker container**.
    - Push it to the **Container Registry or Artifact Registry**.
    - Deploy from there directly to Cloud Run.|

Step 7: Configure the Runtime Environment

After uploading, define how the program should run by configuring environment settings:

- **Environment Variables**: Useful for storing secrets, configurations, or credentials.
- **Resource Settings**:
    - Choose VM types, CPU cores, memory, and disk size (especially for Compute Engine).
    - Define concurrency limits and request timeouts (for Cloud Run or App Engine).
- **Scaling Policies**:
    - **Automatic Scaling**: Automatically adds or removes instances based on traffic.
    - **Manual or Fixed Scaling**: Maintain a specific number of instances.
- **Network and Security Settings**:
    - Open necessary ports (e.g., port 80 for web apps).
    - Configure firewall rules and access scopes.

Step 8: Deploy the Application

With configuration complete, deploy your program:

- **App Engine**:
    - Deploy using Cloud Console or gcloud app deploy.
- **Compute Engine**:
    - Start the VM and manually run the application, or set it to start on boot.
- **Cloud Run**:

    - Deploy from container image using gcloud run deploy or Cloud Console.
- **Deployment Tools**:
    - **Cloud Shell**: Web-based terminal accessible from GCP Console.
    - **Google Cloud SDK**: Command-line tool for deployment and management.

Step 9: Testing and Monitoring

Once deployed, it's critical to validate and monitor the application's performance:

- **Testing Access**:
    - App Engine and Cloud Run provide an **auto-generated URL**.
    - Compute Engine VMs use an **external IP address** (with firewall settings configured).
- **Cloud Monitoring**:
    - Track metrics such as CPU usage, memory, network traffic, and uptime.
- **Cloud Logging**:
    - View logs for errors, warnings, and informational messages.
- **Cloud Trace**:
    - Analyze application latency and request tracing.
- **Cloud Profiler**:
    - Understand application behavior and optimize performance.Continuous monitoring ensures reliability and helps in scaling or debugging as needed.

6. Advantages of using GCP

Google Cloud Platform (GCP) offers a comprehensive set of tools, services, and infrastructure that make it an ideal choice for running modern applications and programs. Organizations ranging from startups to large enterprises choose GCP not just for its computing power, but also for its reliability, scalability, and integration with cutting-edge technologies. This section highlights the core advantages of using GCP, explained in detail:

6.1 Scalability

One of the most significant advantages of GCP is its ability to **scale resources dynamically** based on real-time application demands.

- **Automatic Scaling**: Services like App Engine, Cloud Functions, and Cloud Run automatically scale the number of instances up or down depending on traffic levels. This ensures optimal performance during peak usage while minimizing costs during low activity.
- **Global Load Balancing**: GCP's infrastructure includes powerful global load balancers that distribute traffic across multiple regions and zones, enhancing responsiveness and availability.
- **Elastic Infrastructure**: Compute Engine allows you to scale manually or automatically based on custom metrics like CPU usage or request count.

Example: An e-commerce website hosted on GCP can handle traffic surges during holiday sales without any manual intervention, ensuring smooth customer experience.

6.2 Reliability

Google Cloud is built on the **same infrastructure that powers Google Search, Gmail, and YouTube**, ensuring top-tier reliability and uptime.

- **High Availability Zones**: GCP's architecture includes **multiple data centers (zones)** in each region, enabling developers to design applications that stay online even during localized failures.
- **Service-Level Agreements (SLAs)**: Google provides strict SLAs for many services, ensuring guaranteed uptime percentages (often 99.95% or higher).
- **Redundancy and Failover**: Systems like Cloud Storage and BigQuery offer data redundancy and disaster recovery features by default.

GCP's reliable infrastructure ensures that applications can remain operational even in the face of hardware or network issues.

6.3 Security

Security is a foundational element of GCP, with **built-in features and global compliance standards** designed to protect users, data, and applications.

- **Encryption by Default**: All data stored in GCP is encrypted at rest and in transit using advanced encryption standards.
- **IAM (Identity and Access Management)**: Fine-grained permission controls ensure that users and services access only what they're authorized to.
- **DDoS Protection**: GCP includes protections against distributed denial-of-service attacks, leveraging Google's global infrastructure to detect and mitigate threats.
- **Compliance Certifications**: GCP complies with major security standards like ISO/IEC 27001, SOC 1/2/3, HIPAA, GDPR, and more.

These features make GCP a trusted platform for hosting sensitive or regulated workloads such as healthcare, finance, or government applications.

6.4 Flexibility

GCP supports a wide variety of **languages, frameworks, operating systems, and development environments**, giving developers complete flexibility.

- **Multi-language Support**: Services like App Engine and Cloud Functions support Python, Java, Go, Node.js, PHP, Ruby, and more.
- **Container and VM Support**: With Cloud Run and Compute Engine, users can run custom applications in containers or virtual machines, allowing any software stack to be deployed.
- **Hybrid and Multi-Cloud Support**: GCP tools like Anthos enable running workloads across GCP, on-premises environments, or other cloud platforms like AWS and Azure.

Developers are free to choose the tools and workflows they prefer, enhancing productivity and reducing learning curves.

6.5 Cost-Effectiveness

GCP follows a **pay-as-you-go pricing model**, meaning users are billed only for the resources they consume.

- **Per-Second Billing**: Unlike many cloud providers, GCP charges by the second (after a one-minute minimum), making it highly economical for short-lived tasks.
- **Sustained Use Discounts**: Automatic discounts are applied when services like Compute Engine are used for extended periods within a month.
- **Free Tier and Credits**: New users receive $300 in credits to explore GCP services. Several services also offer always-free usage limits.
- **Detailed Billing Reports**: GCP provides cost analysis tools and budgeting alerts to help users monitor and control spending.

7. Use Cases

Google Cloud Platform (GCP) offers a versatile set of tools and services that support a wide range of applications across different industries and domains. Whether you're developing a simple website or deploying an enterprise-scale data processing system, GCP provides scalable and reliable infrastructure. This section presents detailed examples of common use cases where GCP is effectively used to run programs and deploy solutions.

7.1 Web Applications

Description:
Web applications are dynamic platforms that run in a web browser and deliver content or services over the internet. These can range from portfolio websites to fully-featured online platforms like e-commerce or social networking sites.

How GCP Supports It:

- **App Engine**: Automatically manages the infrastructure, handling deployment, traffic splitting, scaling, and patching.
    - Ideal for applications developed in popular languages like Python, Java, PHP, Node.js, and Go.
    - Developers simply upload their code; GCP handles the rest.
- **Cloud Run**: Suitable for containerized web applications.
    - Offers greater flexibility with runtime environments.
    - Automatically scales the container up or down based on user requests.

Example: An online booking system for a local business can be built using Node.js and deployed on App Engine. If more flexibility is needed (such as using custom dependencies), the same system can be containerized using Docker and deployed on Cloud Run.

7.2 Data Processing Pipelines

Description:
Data processing involves collecting, cleaning, transforming, and analyzing large volumes of data. This is critical for business intelligence, reporting, and predictive analytics.

How GCP Supports It:

- **Cloud Functions**: Allows for event-driven execution of small pieces of code in response to triggers like file uploads or database changes.
    - Example: Automatically process and clean a CSV file uploaded to Cloud Storage.
- **BigQuery**: A fully managed, serverless data warehouse that supports fast SQL queries over massive datasets.

- Enables transformation, aggregation, and analysis of structured data.

Workflow Example:

1. User uploads a data file to **Cloud Storage**.
2. A **Cloud Function** triggers to clean and normalize the data.
3. Processed data is loaded into **BigQuery**.
4. Analysts use SQL queries to generate insights or dashboards.

GCP simplifies the creation of automated and scalable ETL (Extract, Transform, Load) pipelines.

7.3 Machine Learning Models

Description:
Machine learning (ML) applications involve training models on data to make predictions or classifications. These are widely used in image recognition, language processing, fraud detection, and recommendation systems.

How GCP Supports It:

- **Vertex AI**: An integrated platform for building, training, deploying, and managing ML models.
  - Supports AutoML for beginners and custom model development for experienced data scientists.
  - Includes features like data labeling, model monitoring, and hyperparameter tuning.
- **Compute Engine**: Provides customizable virtual machines with support for GPUs and TPUs (Tensor Processing Units), ideal for training resource-intensive models.

Example: A company wants to predict customer churn. They use Vertex AI to train a classification model using customer interaction data. The model is then deployed as an API endpoint for integration with the company's CRM software.

GCP enables both beginners and experts to develop machine learning workflows with high efficiency.

7.4 Microservices Architecture

Description:
Microservices architecture is a way of designing applications as a collection of loosely coupled services, each responsible for a specific function. It offers better modularity, scalability, and maintainability compared to monolithic applications.

How GCP Supports It:

- **Google Kubernetes Engine (GKE)**: Manages the deployment, scaling, and operation of containerized applications using Kubernetes.

    - Supports rolling updates, service discovery, and resource management.
    - Allows integration with monitoring and logging tools.

Example: An e-commerce application is split into multiple microservices:

- User authentication service
- Product catalog service
- Payment processing service
- Order management service

Each service is containerized and deployed on GKE, allowing independent development, scaling, and deployment. If the payment service needs more resources during peak times, GKE can scale it without affecting other parts of the application.

GKE enables modern DevOps practices and supports complex, production-grade applications efficiently.

8.Challenges & Considerations

8.1 Learning Curve

Overview:
GCP is a feature-rich and highly technical platform. For beginners or those unfamiliar with cloud computing concepts, understanding how to use its services can initially be overwhelming.

Key Points:

- **Complex Terminology**: Terms like VPCs, IAM, Load Balancers, and Container Registries require some foundational knowledge.
- **Service Variety**: With numerous overlapping services (e.g., Compute Engine vs. Cloud Run), choosing the right one may be confusing at first.
- **Tools and Interfaces**: Users must become comfortable with various tools such as the Cloud Console, Cloud Shell, and the gcloud CLI.

Mitigation:

- Google offers extensive **official documentation**, tutorials, and quick-start guides.
- Platforms like **Qwiklabs** and **Coursera** offer hands-on labs and courses.
- Starting with **simpler services like App Engine or Cloud Run** can help ease the learning curve.

8.2 Billing and Cost Management

Overview:
GCP operates on a **pay-as-you-go** pricing model, which, while flexible, can result in unexpected charges if resource usage is not closely monitored.

Key Points:

- **Hidden Costs**: Services like data egress (outbound traffic), sustained CPU usage, and network usage can add to the bill unexpectedly.
- **Idle Resources**: Forgetting to shut down unused VMs, databases, or Cloud Functions can accumulate unnecessary charges.
- **Pricing Complexity**: Different services have unique pricing structures that can be difficult to fully understand.

Mitigation:

- Set **budget alerts and cost caps** in the GCP Billing section.
- Use **Billing Reports and Cost Breakdown** tools to monitor usage in real-time.
- Take advantage of **free tier limits** and **sustained-use discounts**.

8.3 Service Limits and Quotas

Overview:
GCP imposes certain **default quotas** on services to protect users and prevent abuse. These quotas can sometimes limit scalability or functionality if not properly managed.

Key Points:

- **API Rate Limits**: Many APIs have usage limits that can throttle or reject requests if exceeded.
- **Region-Based Restrictions**: Not all services or features are available in every GCP region or zone.
- **Resource Caps**: Compute instances, storage, or Cloud Functions may have limits on memory, CPU, or simultaneous executions.

Mitigation:

- Monitor and manage quotas through the **Quotas Dashboard**.
- Request **quota increases** via the GCP Console when needed.
- Carefully choose **regions and zones** based on feature availability and performance requirements.

8.4 Security and Configuration

Overview:
While GCP provides robust security features, misconfigurations can expose applications and data to serious vulnerabilities.

Key Points:

- **IAM Misconfigurations**: Assigning overly permissive roles (e.g., granting 'Editor' access to all users) can lead to unauthorized access.
- **Public Access Risks**: Accidentally exposing Cloud Storage buckets or virtual machines to the internet can create security risks.
- **Data Protection**: Failure to encrypt sensitive data or configure firewall rules can lead to breaches.

Mitigation:

- Follow **the principle of least privilege** when assigning roles in IAM.
- Use **Security Command Center** to scan for vulnerabilities and misconfigurations.
- Implement **firewall rules**, **VPC networks**, and **encryption policies** properly.

9. Conclusion

In conclusion, running a program on Google Cloud Platform involves selecting the appropriate compute service, configuring the project environment, deploying the code, and efficiently managing the underlying infrastructure. GCP provides a robust and scalable platform that allows developers to build and deploy applications with ease, ensuring reliability, security, and seamless scalability. This makes it an ideal choice for modern, cloud-native development, catering to a wide range of use cases from simple web applications to complex data processing pipelines and machine learning workflows.