# Google Cloud Platform (GCP)

**1. What is GCP?**

GCP stands for Google Cloud Platform. It is a cloud computing platform offered by Google that provides a wide range of services, such as computing, storage, networking, and security.

**2. What are the benefits of using GCP?**

GCP provides numerous benefits, such as effortless resource scaling, data security, cost-effectiveness, reliable platform performance, and a broad range of customizable services for diverse business requirements. These services help developers to build and deploy applications quickly and easily.

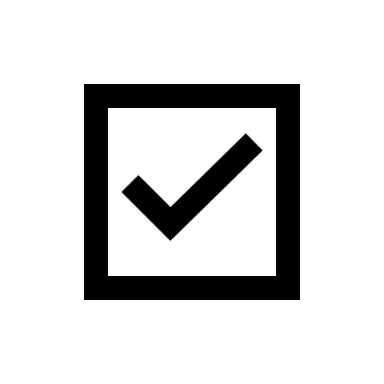
**3. What are the different types of GCP services?**

* **Compute Services:** These services provide the computing power needed to run applications and workloads. E.g. Compute Engine, App Engine
* **Storage Services:** These services are used for storing and managing data. E.g. Cloud Storage, Cloud SQL
* **Networking Services:** These services manage connectivity and traffic flow. E.g. Virtual Private Cloud (VPC), Cloud Load Balancing
* **Big Data Services:** These services are used for processing and analyzing massive datasets. E.g. BigQuery, Pub/Sub
* **AI & Machine Learning Services:** These services provide pre-trained models and platforms for building and deploying your own machine learning solutions. E.g. Vertex AI, Pre-trained APIs
* **Developer Tools and DevOps:** These services help developers build, deploy, and manage applications more efficiently. E.g. Cloud Build, Artifact Registry
* **Security & Identity:** These services help developers build, deploy, and manage applications more efficiently. E.g. Identity and Access Management (IAM), Secret Manager

**4. What are the 3 main GCP principles?**

* **Security 🛡️:** GCP ensures securing data and infrastructure through security measures like encryption, Identity and Access Management (IAM)
* **Scalability📈:** GCP can handle any amount of traffic without users having to manually adjust things. It allows users to effortlessly scale their resources up or down, ensuring seamless adaptability to fluctuating demands while maintaining cost efficiency.

If your website suddenly gets a lot of visitors, GCP will automatically add more power and resources to keep it running smoothly. Also, if traffic drops, it will scale down, so there is no need to pay for resources which don't need. This makes it easy to handle unexpected spikes in demand and keeps your costs in check.

* **Flexibility :** GCP's offers services and tools that provide users with freedom to select and configure solutions that align with their requirements.

**5. What are the 3 pillars of GCP?**

* **Compute:** GCP provide users with virtual machines (VMs) and Compute & App Engine, facilitating highly adaptable compute resources
* **Storage:** GCP delivers scalable storage solutions, such as Google Cloud Storage and BigQuery, for efficient data management, storage, and analysis
* **Networking:** GCP's networking services empower users to construct secure, high-performance networks and establish global connectivity for seamless operation across geographies/locations.

**6. What is the difference between GCP and AWS?**

GCP and AWS are both cloud computing platforms, but they differ in terms of the services they offer, their pricing models, and the level of support they provide. GCP is generally more user-friendly and developer-focused, while AWS is more geared toward enterprise customers.

**7. What is the difference between GCP, Microsoft Azure and AWS?**

* Compute: **AWS's** EC2 (Elastic Compute Cloud) is a powerful and popular option, while **Azure's** Virtual Machines are a natural fit for Windows-based workloads. **GCP's** Compute Engine is known for its custom machine types and per-second billing.
* Databases: **AWS** offers a vast array of database services, including its relational database service, RDS. **Azure** has seamless integration of SQL Server. **GCP** shines with its serverless data warehouse, BigQuery, which is highly optimized for large-scale data analytics.
* Serverless: All three offer robust serverless platforms (**AWS Lambda, Azure Functions, GCP Cloud Functions**), allowing developers to run code without managing servers

A screenshot of a computer

AI-generated content may be incorrect.

**8. Is GCP IaaS or PaaS or SaaS?**

GCP offers all, Infrastructure as a Service (IaaS) through a Compute Engine, Platform as a Service (PaaS) through App Engine, providing versatile cloud solutions and Software as a Service (SaaS) through products that we can use directly like Google Workspace, Drive, Google Meet, etc.

**9. What are regions and zones in GCP?**

In GCP, regions are independent geographic locations/areas where Google's data centers are located, and Zones are smaller, isolated deployment areas within a region. Each zone has its own independent infrastructure.

**10. Why would you deploy an application across multiple zones?**

Deploying an application across multiple zones is a key strategy for achieving high availability and fault tolerance. If a single zone experiences an outage, application can automatically continue to run in the other zones. This ensures that a single point of failure doesn't bring your entire service down, providing your users with a more reliable and resilient experience.

**11. What is Google Compute Engine (GCE)? When would you choose it over another compute service like GKE or Cloud Run?**

GCE is Google's Infrastructure as a Service (IaaS), providing you with raw virtual machines (VMs) and full control over the OS and software. You'd choose it for legacy or highly customized applications that require specific OS configurations or direct VM access, as it offers the most control.

**12. What is Google App Engine (GAE) and how does it differ from Google Compute Engine (GCE)?**

GAE is a Platform as a Service (PaaS) service that allows us to deploy web applications and mobile backends without worrying about the underlying infrastructure. We just need to deploy the code, and Google manages the servers and scaling.

GCE is Infrastructure as a Service (IaaS), providing virtual machines (VMs) and offers full control over the operating system, software, and infrastructure configuration. GCE requires more manual management of scaling, patching, and other operational tasks.

GAE is a great choice for web applications when you want to focus entirely on writing code without any server or infrastructure management.

**13. What is container orchestration?**

Container orchestration is the process of automating the deployment, management, scaling, and networking of containers throughout their lifecycle, making it possible to deploy software consistently across many different environments at scale.

**14.** **What is the Google Kubernetes Engine (GKE)? Explain the difference between GKE Standard and GKE Autopilot.**

GKE is a managed service for deploying and scaling containerized applications using Kubernetes. GKE Standard requires us to manually manage the cluster nodes (VMs) and GKE Autopilot is a fully managed, serverless mode where Google handles the node management.

**15. Explain the concept of serverless computing on GCP. What are the key services that enable it?**

Serverless computing is a model where you run code without managing servers. GCP handles all the provisioning, scaling, and maintenance, and you pay only for the resources used during execution. Key services for this are Cloud Run, Cloud Functions, and BigQuery.

**16. What is Google Cloud Storage?**  
Google Cloud Storage is a service used for storing unstructured data. Store any amount of data and retrieve it as often as you like.

**17. Explain the different Google Cloud Storage classes (Standard, Nearline, Coldline, Archive) and their use cases.**

Standard is for frequently accessed data (e.g., website assets). Nearline is for data accessed less than once a month (e.g., backups). Coldline is for data accessed less than once a quarter (e.g., disaster recovery). Archive is for long-term storage of data you rarely access (e.g., compliance data).

**18. What is Google Cloud SQL?**

Google Cloud SQL is a service that provides fully managed relational databases in the cloud. It's designed to simplify setting up, maintaining, managing, and administering databases. Cloud SQL lets you focus on developing your applications while Google handles database management.

**19. What is Google Cloud Storage?**

Google Cloud Storage is a highly durable and available storage service that provides global access to your data. It suits various use cases, including data backup and archival, content delivery, and big data analytics.

**20. What is Google Cloud Spanner?**

Google Cloud Spanner is a fully managed, mission-critical database service that uniquely combines the benefits of a relational (SQL) database with the non-relational horizontal scalability of NoSQL databases on a global scale.

It provides strong consistency, high availability (up to 99.999% SLA), supports SQL queries and offers automatic data replication and scaling for demanding, high-transaction workloads.

**21. What is Google Cloud Dataflow?**

Google Cloud Dataflow is a fully managed, serverless service that unifies batch and stream data processing

**22.** **What is Google Cloud Pub/Sub?**

Google Cloud Pub/Sub is a messaging service that enables us to send and receive messages between independent applications. It operates on a publisher-subscriber model, where services that produce data (publishers) send messages to a central channel called a topic, and services that are interested in that data (subscribers) receive messages from a subscription to that topic. It guarantees the delivery of messages in the order they were published. It can handle high throughput and is highly available, making it an ideal choice for businesses that require reliable and scalable messaging services.

**23. What are Google Cloud Functions?**

Google Cloud Functions is a service that allows you to run code without worrying about managing servers or infrastructure. It's a serverless computing service that supports different programming languages, making it easy to build applications. With Cloud Functions, you can focus on creating your application logic and deploying your code to the cloud. This feature makes developing and deploying your applications quicker and more efficient.

**24. What is Google Cloud Load Balancing?**

Google's Load Balancing is a service that distributes traffic across multiple instances or services. This service can balance the load across multiple regions or protocols, ensuring high availability and reliability.

**25. What is Google Cloud IAM?**

Google Cloud IAM (Identity and Access Management) is a service for managing access to resources in GCP. It allows you to define roles and permissions for users and groups, and it integrates with other GCP services for easy access control.

**26. Explain how Identity and Access Management (IAM) works in GCP. What are the key components of an IAM policy?**

IAM controls who has what access to which resources. The key components of an IAM policy are the Principal (who is requesting access, e.g., a user or service account), the Role (what permissions are being granted), and the Resource (the specific GCP asset).

**27. What are the Roles in IAM?**There are three primary types of IAM roles in GCP: Basic, Predefined, and Custom.

**Basic Roles:**

Owner: Has full administrative control over a project,

Editor: Can create and modify resources within a project but cannot manage roles or billing.

Viewer: Has read-only access to all resources.

**Predefined Roles:** These are roles created and managed by Google that provide fine-grained access to a specific Google Cloud service. For e.g. roles/run.admin for Cloud Run and roles/datastore.viewer to only read data from Datastore

**Custom Roles:** When predefined roles don't fit your exact needs, you can create a custom role by combining a specific set of permissions

**28. What is Google Cloud CDN?**

Google Cloud CDN (Content Delivery Network) is a service that distributes content/data from edge locations to end-users globally, reducing the load on origin servers. This feature means users can access content faster and more reliably, improving overall experience.

**29. What is Cloud Armor and what problem does it solve?**

Cloud Armor is a security service that provides Web Application Firewall (WAF) capabilities. It solves the problem of protecting applications from web-based threats like SQL injections and malicious attacks

**30. How do you manage secrets and sensitive data on GCP? Name a service you would use.**

GCP manage secrets and sensitive data using Secret Manager, a fully managed service that securely stores, manages, and provides access to sensitive data like API keys and passwords.

**31. How do you monitor and troubleshoot a GCP application? Name the services you would use.**

GCP provides the Cloud Operations Suite which provide service – i.e. Cloud Monitoring for metrics and alerting, and Cloud Logging for collecting, viewing, and analyzing logs to identify and troubleshoot issues

**32. How would you design a highly available and fault-tolerant application on GCP?**

We'd design it by deploying the application across multiple regions and zones, using a Global Load Balancer to distribute traffic, implementing managed instance groups with auto-scaling, and using managed services with built-in redundancy.

**33. What are service accounts and why are they important?**

A service account is a special Google account for applications and VMs, not human users. They are important because they provide an identity for your workloads to authenticate and access GCP resources without using user credentials

**34. How does GCP ensure the security of data?**

GCP uses multiple layers of security to protect data, including network security, access management, data encryption, and security audits. GCP also complies with multiple security certifications and standards

**35. Is GCP an ETL tool?**

GCP isn't primarily an ETL tool but offers services like Dataflow and BigQuery for ETL and data processing.

**36. Does GCP use SQL?**

GCP supports SQL through services like BigQuery for data querying and Cloud SQL for managing relational databases.

**37. What is a Virtual Private Cloud (VPC) in GCP, and how is it structured?**

A Virtual Private Cloud (VPC) in GCP is a global, software-defined network that provides a logically isolated section of the Google Cloud infrastructure. Think of it as your own private network within Google's cloud, where you can launch and manage your GCP resources like virtual machines, Kubernetes clusters, and databases.

**38. Difference between Compute Engine vs App Engine vs Cloud Run vs GKE. When do you use Java microservices?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service | Type | Management Level | Scaling Model | Ideal for Java Microservices |
| Cloud Run | Serverless Containers | Fully Managed | Scales from zero to thousands based on requests. | Default choice for new stateless microservices. |
| GKE | Managed Kubernetes | Managed (Control Plane) | Highly flexible, scales Pods and nodes automatically. | Complex, large-scale systems needing custom orchestration. |
| App Engine | Platform as a Service (PaaS) | Fully Managed | Scales instances based on traffic or tasks. | Simple, web-based microservices where you want zero infrastructure management. |
| Compute Engine | Infrastructure as a Service (IaaS) | Self-Managed | Manual or autoscaling via Instance Groups. | Legacy or highly specialized applications requiring OS-level control. |

**39. What is Google Cloud Build?**

Google Cloud Build is a CI/CD platform for building, testing, and deploying code. It integrates with Cloud Source Repositories, GitHub, Bitbucket, etc.

**40.** **What is a firewall rule in GCP, and how does it control traffic to and from VM instances?**

A[firewall rule in Google Cloud Platform (GCP)](https://www.geeksforgeeks.org/cloud-computing/how-to-open-port-in-gcp-vm/) is a set of criteria that dictates which incoming and outgoing network traffic is allowed to reach or leave VM instances. It controls traffic based on factors like IP addresses, protocols, and ports.

**41. What is the difference between Persistent Disk and Local SSD in GCP?**

GCP Persistent Disk offers strong block storage. Redundancy and high availability are the advantages it provides. Local SSD, on the other hand, offers temporary block storage, which is high-performance, low-latency, and linked to the actual hardware operating the virtual machine instance.

**42.** **What is a GCP diagram?**

A GCP diagram serves as a graphical representation of the cloud architecture within the Google Cloud Platform. These visualizations represent interconnections and interactions among various GCP services and components for planning and designing complex cloud deployments.

**43**. **What is Artifact Registry vs Container Registry?**

Both the services are used for storing container images.

Artifact Registry: Modern, recommended and Container Registry: Legacy

**44. What is the difference between Cloud SQL and Cloud Spanner? When would you use each?**

**Cloud SQL:** A managed relational database service. Traditional DBs - MySQL, PostgreSQL, SQL Server, etc. Ideal for transactional databases with a moderate amount of data, applications with a defined schema, and systems that don't require global scalability.

**Cloud Spanner:** A globally distributed, horizontally scalable, strongly consistent relational database. It combines the best traditional relational databases (SQL, strong consistency) with the horizontal scalability of NoSQL databases due to which it referred to as a "NewSQL" database. Used for mission-critical, high-availability applications that require a single database across multiple regions.

**45. What is gcloud and how is it used?**

gcloud is the CLI tool for GCP. Used to Deploy apps, Configure projects and Manage services like Cloud Run, GKE, and IAM

Sample commands –

**gcloud init:** This is the first command you'd use. It interactively guides you through a complete setup, including authenticating your account, setting the default project, and configuring your compute region and zone.

**gcloud auth login:** Authorizes the gcloud CLI to access your Google account. It opens a browser window for you to sign in.

**gcloud config list:** Displays your current active configuration, including the project, account, region, and zone

**gcloud compute instances list**: Lists all VM instances in your current project.  
**gcloud app deploy**: Deploys an application to App Engine

**gcloud run deploy**: Deploys a new container image or updates an existing Cloud Run service

**gcloud functions deploy**: Deploys a new Cloud Function.

**46.** **What is Pub/Sub, and when would you use it?**

Pub/Sub is a fully managed, real-time messaging service. I would use it for asynchronous communication between microservices to decouple them. For example, a "user creation" service could publish a message to a topic, and a separate "email notification" service could subscribe to that topic and send a welcome email.

**47. What are preemptible VMs?**

Preemptible VMs are Compute Engine instances offered at lower cost, but Google may shut them down (preempt) at short notice (after giving short warning).

Pros: Cost savings (often much cheaper. Good for batch jobs, non‑critical workloads, Development and Testing Environments.

Cons: No Uptime Guarantee. Can Be Terminated Anytime (GCP can stop the VM at any moment, with as little as 30 sec notice). They do not run longer than 24 hours.

**48. How do you manage identity and access control in GCP?**

Using IAM, service accounts, and the principle of least privilege.

**49. What are the differences between the Standard and Flexible environments of App Engine? When would you choose one over the other for a Java application?**

Standard environment

* Application instances that run in a sandbox, using the runtime environment of a supported language only.
* Sandbox restricts what the application can do
  + only allows the app to use a limited set of binary libraries
  + app cannot write to disk
  + limits the CPU and memory options available to the application
* Rapid scaling and Zero downscaling is possible, can scale from zero instances up to thousands very quickly.
* Choose Standard for a lightweight, stateless microservice or a simple web application with unpredictable traffic

Flexible environment

* Application instances run within Docker containers on Compute Engine virtual machines (VM).
* Must have at least one instance running for each active version and can take longer to scale up in response to traffic.
* Choose Flexible for a monolithic or complex application with specific runtime needs, consistent traffic, or long running processes

**50. What is the difference between Cloud SQL, Cloud Spanner, and Cloud Firestore?**

A screenshot of a computer

AI-generated content may be incorrect.

***Application Development & Deployment***

**51. How do you create a new project in GCP?**

Go to console.cloud.google.com to get logged into Google Cloud Console.

* Select "New Project" from the option list located at the very top of the page following click on it.
* Select a billing account, enter the project name, and specify the location or organization.
* To finish the configuration of the new project, click "Create."

**52. How would you containerize and deploy a Spring Boot application on GCP?**

1. **Containerize**: Use a Dockerfile to create a Docker image of the Spring Boot application. Build the project and create JAR. The Dockerfile copies the JAR and specifies the ENTRYPOINT to run it.
2. **Push to Artifact Registry:** Tag the Docker image and push it to Google Artifact Registry (GCP's private container registry).
3. **Deploy:** Deploy the container to a managed service like Cloud Run for a serverless approach, or to Google Kubernetes Engine (GKE) for more control and orchestration. For simple deployments, Cloud Run is the preferred choice as it handles scaling, traffic routing, and infrastructure management automatically.

**53. How do you authenticate your Java application securely to GCP services?**

Authenticating a Java application to GCP services is most securely done using Application Default Credentials (ADC). This strategy automatically finds credentials without hardcoding them in your code.

ADC works by searching for credentials in a specific, prioritized order:

1. GOOGLE\_APPLICATION\_CREDENTIALS environment variable (for local dev).
2. gcloud CLI user credentials (also for local dev).
3. The default service account of the hosting GCP resource (for production on GCP).

**54. How do you create service accounts in GCP?**  
We can create a service account in Google Cloud Console, within the IAM & Admin section.

1. Navigate to IAM & Admin in the main navigation menu.
2. Click on Service Accounts.
3. Click + CREATE SERVICE ACCOUNT.
4. Follow the prompts to name the service account and assign it a role

**55. How can you manage secrets like database passwords in a Spring Boot application deployed on GCP?**

By avoiding hardcoding secrets in code or configuration files. The recommended approach is to use Google Cloud Secret Manager. We can configure the Spring Boot application to retrieve secrets from the Secret Manager at startup. The application's service account (which it uses to authenticate with GCP) must have the correct IAM role to access the secrets.

**56. How to set up a Cloud SQL instance.**

To set up a Cloud SQL instance:

1. Navigate to Google Cloud Console.
2. Choose the project whereby the instance is to be generated.
3. Click "Create Instance" after choosing SQL in the menu to the left.
4. Choose the instance type, database engine, and options for configuration.
5. To start your [Cloud SQL instance,](https://www.geeksforgeeks.org/devops/google-cloud-sql/) click "Create."

**57. How can you connect a Spring Boot application to Cloud SQL?**

1. Use Cloud SQL Socket Factory dependency in pom.xml or gradle.dependencies file
2. Configure application.properties

|  |
| --- |
| spring.datasource.url=<JDBC MySQL or ORACLE> :///<DB\_NAME>?cloudSqlInstance=<INSTANCE\_CONNECTION\_NAME>&socketFactory=com.google.cloud.sql.mysql.SocketFactory |

**58. How to enable Secret Manager in GCP for Java applications?**

Configure application.properties

|  |
| --- |
| spring.cloud.gcp.secretmanager.enabled=true |

**59. How do you set environment-specific configs in GCP deployments?**

* Use Spring Profiles
* Pass environment variables in:
  + Cloud Run (via --set-env-vars)
  + Kubernetes (via env: in YAML)
  + App Engine (via app.yaml)

**60. What GCP services would you use for logging and monitoring your microservices / application?**

* **Logging**: Cloud Logging is the centralized logging service. Spring Boot's default logging frameworks can be configured to write logs to stdout, which Cloud Logging automatically ingests and makes searchable.
* **Monitoring**: Cloud Monitoring provides metrics and dashboards to track the performance and health of your services. We can set up custom metrics and create alerts based on specific thresholds (e.g., CPU usage, latency).
* **Tracing**: Cloud Trace is used to analyze application latency and distributed requests across applications or microservices. It helps visualize the flow of a request and identify bottlenecks.
* **Profiling**: Cloud Profiler is a continuous profiling service that helps you identify and optimize performance bottlenecks in your code (e.g., high CPU or memory usage).

**61. How would you store user-uploaded files or images in GCP?**

I would use Cloud Storage, which is Google's object storage service. It's highly scalable, durable, and provides a simple API for storing and retrieving files.

**62. How do you optimize costs when running an application in GCP?**

To optimize costs when running an application in GCP, focus on right-sizing resources, automating scaling, and leveraging cost-saving pricing models.

**Optimize Compute Costs:**

1. Choose the Right Service: For stateless microservices, Cloud Run is the most cost-effective option. It's a serverless platform that scales to zero when not in use, so you only pay for the duration your code is actively handling a request. For stateful applications or those needing more control, use Google Kubernetes Engine (GKE).

2. Leverage Auto-Scaling: Use auto-scaling to automatically adjust the number of instances based on traffic. This ensures you're not paying for idle resources during low-traffic periods. For GKE, you can configure the Horizontal Pod Autoscaler option based on CPU or custom metrics.

3. Utilize Preemptible VMs: For fault-tolerant workloads like batch processing or non-critical development environments, use Preemptible VMs (also known as Spot VMs).

**Optimizing Storage Costs:**

1. Choose the Right Storage Class: GCP's Cloud Storage offers different storage classes, each with a distinct pricing model based on access frequency.

**Application-Level Optimizations:**

* + 1. Optimize the JVM: Use a modern, container-aware JVM (like

OpenJDK 11+)

2. Right-Size Resources: Use GCP's Recommendation Hub to find oversized VMs & Underutilized services. Regularly review your resource usage and adjust the size of your compute instances (VMs, GKE nodes, or Cloud Run instance memory) to match your workload's actual needs

**63. How do you implement CI/CD pipelines in GCP?**

To implement continuous integrations and continuous deployments (CI/CD) pipelines in usage under GCP:

* Source Code Management: Make advantage of Google Cloud Source Repositories or [GitHub/Bitbucket](https://www.geeksforgeeks.org/git/difference-between-bitbucket-and-github/) connectivity.
* Continuous Integration: Automate the code packaging, testing, and deployment using the [Google Cloud Build.](https://www.geeksforgeeks.org/cloud-computing/google-cloud-platform-building-ci-cd-pipeline-for-package-delivery/)
* Artifact Storage: Build artifacts may be kept in Google Cloud Storage, Artifact Registry or the Container Registry.
* Continuous Deployment: Use the Google Cloud Deploy or [Cloud Run](https://www.geeksforgeeks.org/cloud-computing/guide-to-deploy-containers-on-google-cloud-run/) for automatic deployment to GKE, the App Engine, or Cloud Runs.
* Monitoring: Using Google Cloud Monitor and Logging to keep tabs on the performance and health of your cloud deployment.

**64. What is the principle of least privilege and how would you apply it in GCP for a Java service account?**

The Principle of Least Privilege (PoLP) is a security concept that states that a user, program, or service account should only have the minimum permissions necessary to perform its intended function. Granting more permissions than required increases the potential attacks.

**Avoid Default Service Accounts:** When you create a new GCP project default service accounts are automatically created. These often come with overly broad permissions (e.g., "Editor" role) that violate PoLP. Instead, We should Create a new, dedicated service account for our application.

**Grant the Least Permissive Roles:** Avoid Basic Roles like Owner, Editor, and Viewer which grants access for entire project. Instead create a custom role that contains only the specific permissions our application needs.

**65. How would you troubleshoot a slow-running application?**

* Start with Cloud Monitoring to find the performance bottleneck (e.g., high CPU, high memory).
* Use Cloud Profiler service to find the specific code causing high CPU or memory usage.
* Use Cloud Trace service to analyze latency issues across services.
* Use Cloud Debugger service to inspect the application's state in production if you need a deeper look.

# Core Terms

**Cloud Services Models**

* **Infrastructure as a Service** ([IaaS](https://www.geeksforgeeks.org/cloud-computing/infrastructure-as-a-service-iaas/)): You rent the basic building blocks like virtual machines, storage, and networking, giving you the most control over the environment.
* **Platform as a Service** ([PaaS](https://www.geeksforgeeks.org/cloud-computing/platform-as-a-service-paas-and-its-types/)): You're provided with a complete platform to build, run, and manage applications, so you can focus on coding without worrying about the underlying infrastructure.
* **Software as a Service** ([SaaS](https://www.geeksforgeeks.org/software-engineering/software-as-a-service-saas/)): You simply use a complete, ready-to-use software application over the internet, with no need to manage any infrastructure, platform, or software.

Reference: https://www.geeksforgeeks.org/software-engineering/difference-between-iaas-paas-and-saas/

**Compute Services: Provide the computing power needed to run applications and workloads**

* **Compute Engine:** This is GCP's Infrastructure-as-a-Service (IaaS) offering, provide customizable virtual machines (VMs). You have full control over the operating system and software.
* **App Engine:** A Platform-as-a-Service (PaaS) that allows to deploy web applications & mobile backends without worrying about the underlying infrastructure.
* **Google Kubernetes Engine (GKE):** A managed environment for deploying, managing, and scaling containerized applications using Kubernetes. It handles a lot of the heavy lifting of container orchestration.
* **Cloud Run:** A serverless compute platform for running containerized applications. It automatically scales based on traffic and scales down to zero when not in use, making it very cost-effective.
* **Cloud Functions:** A serverless, event-driven compute service that lets you run small pieces of code in response to events, such as a file upload to a storage bucket or a message in a queue.

**Storage and Database Services: Used for storing and managing data**

* **Cloud Storage:** A scalable, durable, and highly available storage service for storing unstructured data (like images, videos, and documents).
* **Cloud SQL:** A fully managed relational database service for MySQL, PostgreSQL, and SQL Server.
* **Cloud Spanner:** A globally distributed, horizontally scalable, and strongly consistent relational database service for mission-critical applications.
* **Cloud Firestore:** A NoSQL document database for mobile, web, and serverless applications.
* **Cloud Bigtable:** A fast, fully managed, and highly scalable NoSQL database service for large analytical and operational workloads.

**Networking Services: Manage connectivity and traffic flow**

* **Virtual Private Cloud (VPC):** A private, isolated network within google cloud for your resources.
* **Cloud Load Balancing:** A managed service that distributes incoming traffic across multiple instances to ensure high availability and performance.
* **Cloud CDN (Content Delivery Network):** A service that caches web content at Google's edge locations worldwide to deliver it faster to users.
* **Cloud DNS (Domain Name System):** A high-performance, resilient, and global DNS service for managing domain name resolution.

**Developer Tools and DevOps: Helps developers build, deploy, and manage applications efficiently**

* **Cloud SDK:** A set of tools for interacting with GCP services from the command line.
* **Cloud Build:** A managed service for running builds and creating a CI/CD pipeline.
* **Artifact Registry:** A single place to manage all your container images and language packages.
* **Cloud Source Repositories:** A private Git repository hosted on GCP.
* **Cloud Monitoring and Cloud Logging:** Services for monitoring application performance and collecting logs.

**AI and Machine Learning: Provide pre-trained models and platforms for building and deploying machine learning solutions**

* **Vertex AI:** A unified platform for building, deploying, and managing machine learning models.
* **Pre-trained APIs:** Services like Vision AI, Natural Language API, and Speech-to-Text provide pre-built models you can use with a simple API call.

**Big Data and Analytics Services: Used for processing and analyzing massive datasets**

* **BigQuery:** A serverless, highly scalable, and cost-effective data warehouse for running fast SQL queries on massive datasets.
* **Cloud Dataflow:** A fully managed service for processing both batch and stream data.
* **Pub/Sub:** A messaging service for ingesting and distributing event data in real time.

**Security and Identity: For protecting your data and resources**

* **Identity and Access Management (IAM):** A service that lets you manage who has access to your GCP resources.
* **Secret Manager:** A service for storing and managing sensitive data like API keys and passwords.

More..

sdjvskdqkhvkhfvqkhebf