# GCP Assignment 2

1. What is cloud computing, and how does it work?

Cloud computing is the delivery of different services through the Internet. These resources include tools and applications like data storage, servers, databases, networking, and software.

Rather than keeping files on a proprietary hard drive or local storage device, cloud-based storage makes it possible to save them to a remote database. As long as an electronic device has access to the web, it has access to the data and the software programs to run it.

Cloud computing is a popular option for people and businesses for a number of reasons including cost savings, increased productivity, speed and efficiency, performance, and security.

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. Large clouds often have functions distributed over multiple locations, each location being a data center.

The cloud is basically a decentralized place to share information through satellite networks. Every cloud application has a host, and the hosting company is responsible for maintaining the massive data centers that provide the security, storage capacity and computing power needed to maintain all of the information users send to the cloud.

The most prominent companies hosting the cloud are major players like Amazon (Amazon Web Services), Microsoft (Azure), Apple (iCloud) and Google (Google Drive), but there's also a plethora of other players, large and small. These hosting companies can sell the rights to use their clouds and store data on their networks, while also offering the end user an ecosystem that can communicate between devices and programs (e.g., download a song on your laptop and it's instantly synced to the iTunes software on your iPhone).

2. List all of the GCP deployment services available.

Google Cloud Platform services are robust. One way to navigate them is to consider which solutions are available based on your primary computing needs: infrastructure as a service (IaaS), platform as a service (PaaS), and software-as-a-service (SaaS).

• **IaaS** enables IT to run virtual machines without having to invest in or manage this computing infrastructure themselves. Often IT will opt for an IaaS solution when the workload is temporary, experimental, or subject to unexpected changes (e.g. sandbox projects).

• **PaaS** is the next step, building on the IaaS model. Customers opt for all of the benefits of IaaS, plus they get underlying infrastructure – like operating systems and middleware. Their vendor hosts and manages all of these elements.

• **SaaS** goes one more step – everything is available via the web: the provider hosts, manages, and delivers the entire infrastructure including applications. Users simply log in to access the resources the specific solution delivers, e.g. backup and recovery tools.

Another way to navigate Google Cloud Platform is by service-offering type. Core **service categories** include:

1. Compute
2. Networking
3. Storage and Databases
4. Artificial Intelligence (AI) / Machine Learning (ML)
5. Big Data
6. Identity and Security
7. Management Tools

Within these categories, **top services** include:

| Compute | Networking | Storage and Databases | AI and ML | Big Data | Identity and Security | Management Tools |
|---|---|---|---|---|---|---|
| Compute engine | Google Cloud Virtual Network | Cloud Storage | Cloud Machine Learning | Big Query | Google Cloud IAM | Stackdriver |

| | | | | | | |
|---|---|---|---|---|---|---|
| App Engine | Cloud Load Balancing | Cloud SQL | Cloud Vision API | Cloud Dataflow | Cloud Resource Manager | Deployment Manager |
| Container Engine | Cloud CDN | Bigtable | Cloud Speech API | Dataproc | Cloud Security Scanner | Cloud Shell |
| Container Registry | Google Cloud Interconnect | Cloud Datastore | Natural Language API | Cloud Datalab | | Google Cloud Billing API |
| Cloud Functions | Cloud DNS | Cloud Spanner | Translate API | Google Genomics | | |
| Cloud Pub/Sub | | Persistent Disk | | | | |
| | | Cloud Source Repositories | | | | |

1. Compute

**App Engine**: App Engine enables you to build and host applications on the same systems that power Google applications. App Engine offers fast development and deployment; simple administration, with no need to worry about hardware, patches or backups; and effortless scalability.

**Compute Engine**: Compute Engine offers scalable and flexible virtual machine computing capabilities in the cloud, with options to utilize certain CPUs, GPUs, or Cloud TPUs. You can use Compute Engine to solve large-scale processing and analytic problems on Google's computing, storage, and networking infrastructure.

**Google Cloud VMware Engine (GCVE)**: GCVE is a managed VMware-as-a-Service that is specifically designed for running VMware workloads on Google Cloud Platform. GCVE enables customers to run VMware virtual machines natively in a dedicated, private, software-defined data center.

2. Storage

**Cloud Storage**: Cloud Storage is a RESTful service for storing and accessing your data on Google's infrastructure. The service combines the performance and scalability of Google's cloud with advanced security and sharing capabilities.

**Persistent Disk**: Persistent Disk is a durable and high performance block storage service for Google Cloud Platform. Persistent Disk provides SSD and HDD storage that can be attached to instances running in either Compute Engine or Google Kubernetes Engine.

**Cloud Filestore**: Cloud Filestore is a scalable and highly available shared file service fully-managed by Google. Cloud Filestore provides persistent storage ideal for shared workloads. It is best suited for enterprise applications requiring persistent, durable, shared storage which is accessed by NFS or requires a POSIX compliant file system.

**Cloud Storage for Firebase**: Cloud Storage for Firebase adds customizable Google security (via Firebase Security Rules for Cloud Storage) to file uploads and downloads for your Firebase apps, as well as robust uploads and downloads regardless of network quality through the Firebase SDK. Cloud Storage for Firebase is backed by Cloud Storage, a service for storing and accessing your data on Google's infrastructure.

3. Databases

**Cloud Bigtable**: Cloud Bigtable is a fast, fully-managed, highly-scalable NoSQL database service. It is designed for the collection and retention of data from 1TB to hundreds of PB.

**Datastore**: Datastore is a fully-managed, schemaless, non-relational datastore. It provides a rich set of query capabilities, supports atomic transactions, and automatically scales up and down in response to load. It can scale to support an application with 1,000 users or 10 million users with no code changes.

**Firestore**: Firestore is a NoSQL document database for storing, syncing, and querying data for mobile and web apps. Its client libraries provide live synchronization and offline support, while its security features and integrations with Firebase and Google Cloud Platform accelerate building serverless apps.

**Memorystore**: Memorystore, which includes Memorystore for Redis and Memorystore for Memcached, provides a fully-managed in-memory data store service that allows customers to deploy distributed caches that provide sub-millisecond data access.

**Cloud Spanner**: Cloud Spanner is a fully-managed, mission-critical relational database service. It is designed to provide a scalable online transaction processing (OLTP) database with high availability and strong consistency at global scale.
**Cloud SQL**: Cloud SQL is a web service that allows you to create, configure, and use relational databases that live in Google's cloud. It is a fully-managed service that maintains, manages, and administers your databases, allowing you to focus on your applications and services.
4. Networking
**Cloud CDN**: Cloud CDN uses Google's globally distributed edge points of presence to cache HTTP(S) load balanced content close to your users.
**Cloud DNS**: Cloud DNS is a high performance, resilient, global, fully-managed DNS service that provides a RESTful API to publish and manage DNS records for your applications and services.
**Cloud IDS (Cloud Intrusion Detection System)**: Cloud IDS is a managed service that aids in detecting certain malware, spyware, command-and-control attacks, and other network-based threats.
**Cloud Interconnect**: Cloud Interconnect offers enterprise-grade connections to Google Cloud Platform using Google Services for Dedicated Interconnect, Partner Interconnect and Cloud VPN. This solution allows you to directly connect your on-premises network to your Virtual Private Cloud.
**Cloud Load Balancing**: Cloud Load Balancing provides scaling, high availability, and traffic management for your internet-facing and private applications.
**Cloud NAT (Network Address Translation)**: Cloud NAT enables instances in a private network to communicate with the internet.
**Cloud Router**: Cloud Router enables dynamic Border Gateway Protocol (BGP) route updates between your VPC network and your non-Google network.
**Cloud VPN**: Cloud VPN allows you to connect to your Virtual Private Cloud (VPC) network from your existing network, such as your on-premises network, another VPC network, or another cloud provider's network, through an IPsec connection using (i) Classic VPN, which supports dynamic (BGP) routing or static routing (route-based or policy-based), or (ii) HA (high-availability) VPN, which supports dynamic routing with a simplified redundancy setup, separate failure domains for the gateway interfaces, and a higher service level objective.

**Google Cloud Armor**: Google Cloud Armor offers a policy framework and rules language for customizing access to internet-facing applications and deploying defenses against denial of service attacks as well as targeted application attacks. Components of Google Cloud Armor include: L3/L4 volumetric DDos Protection, preconfigured web-application firewall (WAF) rules, and custom rules language.
**Google Cloud Armor Managed Protection Plus**: Google Cloud Armor Managed Protection Plus is a managed application protection service subscription that bundles Google Cloud Armor WAF and DDoS Protection with additional services and capabilities including DDoS response support, DDoS bill protection, and Google Cloud Armor Adaptive Protection, which is Google's machine-learning based solution to protect internet-facing endpoints from network and application-based attacks.
**Network Connectivity Center**: Network Connectivity Center is a hub-and-spoke model for network connectivity management in Google Cloud that facilitates connecting a customer's resources to its cloud network.
**Network Intelligence Center**: Network Intelligence Center is Google Cloud's comprehensive network monitoring, verification, and optimization platform across the Google Cloud, multi-cloud, and on-prem environments.
**Network Service Tiers**: Network Service Tiers enable you to select different quality networks (tiers) for outbound traffic to the internet: the Standard Tier primarily utilizes third party transit providers while the Premium Tier leverages Google's private backbone and peering surface for egress.
**Service Directory**: Service Directory is a managed service that offers customers a single place to publish, discover and connect their services in a consistent way, regardless of their environment. Service Directory supports services in Google Cloud, multi-cloud, and on-prem environments and can scale up to thousands of services and endpoints for a single project.
**Traffic Director**: Traffic Director is Google Cloud Platform's traffic management service for open service meshes.
**Virtual Private Cloud**: Virtual Private Cloud provides a private network topology with IP allocation, routing, and network firewall policies to create a secure environment for your deployments.
5. Operations

**Cloud Debugger**: Cloud Debugger connects your application's production data to your source code by inspecting the state of your application at any code location in production without stopping or slowing down your requests.
**Cloud Logging**: Cloud Logging is a fully-managed service that performs at scale and can ingest application and system log data, as well as custom log data from thousands of VMs and containers. Cloud Logging allows you to analyze and export selected logs to long-term storage in real time. Cloud Logging includes the Error Reporting feature, which analyzes and aggregates the errors in your cloud applications and notifies you when new errors are detected.
**Cloud Monitoring**: Cloud Monitoring provides visibility into the performance, uptime, and overall health of cloud-powered applications. Cloud Monitoring collects metrics, events, and metadata from certain Services, hosted uptime probes, application instrumentation, alert management, notifications and a variety of common application components.
**Cloud Profiler**: Cloud Profiler provides continuous profiling of resource consumption in your production applications, helping you identify and eliminate potential performance issues.
**Cloud Trace**: Cloud Trace provides latency sampling and reporting for App Engine, including per-URL statistics and latency distributions.
6. Developer Tools
**Artifact Registry**: Artifact Registry is a service for managing container images and packages. It is integrated with Google Cloud tooling and runtimes and comes with support for native artifact protocols. This makes it simple to integrate it with your CI/CD tooling to set up automated pipelines.
**Container Registry**: Container Registry is a private Docker image storage system on Google Cloud Platform. The registry can be accessed through an HTTPS endpoint, so you can pull images from your machine, whether it's a Compute Engine instance or your own hardware.
**Cloud Build**: Cloud Build is a service that executes your builds on Google Cloud Platform infrastructure. Cloud Build can import source code from Cloud Storage, Cloud Source Repositories, GitHub, or Bitbucket; execute a build to your specifications; and produce artifacts such as Docker containers or Java archives.
**Cloud Source Repositories**: Cloud Source Repositories provides Git version control to support collaborative development of any application or service, including those that run on App Engine and Compute Engine.

***Firebase Test Lab**: Firebase Test Lab lets you test your mobile app using your test code or automatically on a wide variety of devices and device configurations hosted in a Google data center, with test results made available in the Firebase console.
**Google Cloud Deploy**: Google Cloud Deploy is a service for managing and performing application continuous delivery to Google Kubernetes Engine. It allows for process specification and control of application delivery.
**Test Lab**: Test Lab enables you to test mobile applications using physical and virtual devices in the cloud. It runs instrumentation tests and script-less robotic tests on a matrix of device configurations, and reports detailed results to help improve the quality of your mobile app.
7. Data Analytics
**BigQuery**: BigQuery is a fully-managed data analysis service that enables businesses to analyze Big Data. It features highly scalable data storage that accommodates up to hundreds of terabytes, the ability to perform ad hoc queries on multi-terabyte datasets, and the ability to share data insights via the web.
**Cloud Composer**: Cloud Composer is a managed workflow orchestration service that can be used to author, schedule, and monitor pipelines that span across clouds and on-premises data centers. Cloud Composer allows you to use Apache Airflow without the hassle of creating and managing complex Airflow infrastructure.
**Cloud Data Fusion**: Cloud Data Fusion is a fully-managed, cloud native, enterprise data integration service for quickly building and managing data pipelines. Cloud Data Fusion provides a graphical interface to help increase time efficiency and reduce complexity and allows business users, developers, and data scientists to easily and reliably build scalable data integration solutions to cleanse, prepare, blend, transfer, and transform data without having to wrestle with infrastructure.
**Cloud Life Sciences (formerly Google Genomics)**: Cloud Life Sciences provides services and tools for managing, processing, and transforming life sciences data.
**Data Catalog**: Data Catalog is a fully-managed and scalable metadata management service that empowers organizations to quickly discover, manage, and understand their data in Google Cloud. It offers a central data catalog across certain Google Cloud Services that allows organizations to have a unified view of their data assets.
**Dataplex**: Dataplex is an intelligent data fabric that helps customers unify distributed data and automate management and governance across that data to power analytics at scale.

**Dataflow**: Dataflow is a fully-managed service for strongly consistent, parallel data-processing pipelines. It provides an SDK for Java with composable primitives for building data-processing pipelines for batch or continuous processing. This service manages the life cycle of Compute Engine resources of the processing pipeline(s). It also provides a monitoring user interface for understanding pipeline health.
**Datalab**: Datalab is an interactive tool for exploration, transformation, analysis and visualization of your data on Google Cloud Platform. It runs in your cloud project and enables you to write code to use other Big Data and storage services using a rich set of Google-authored and third party libraries.
**Dataproc**: Dataproc is a fast, easy to use, managed Spark and Hadoop service for distributed data processing. It provides management, integration, and development tools for unlocking the power of rich open source data processing tools. With Dataproc, you can create Spark/Hadoop clusters sized for your workloads precisely when you need them.
**Dataproc Metastore**: Dataproc Metastore provides a fully-managed metastore service that simplifies technical metadata management and is based on a fully-featured Apache Hive metastore. Dataproc Metastore can be used as a metadata storage service component for data lakes built on open source processing frameworks like Apache Hadoop, Apache Spark, Apache Hive, Presto, and others.
**Datastream**: Datastream is a serverless change data capture (CDC) and replication service that enables data synchronization across heterogeneous databases, storage systems, and applications with minimal latency.
**Pub/Sub**: Pub/Sub is designed to provide reliable, many-to-many, asynchronous messaging between applications. Publisher applications can send messages to a "topic" and other applications can subscribe to that topic to receive the messages. By decoupling senders and receivers, Pub/Sub allows developers to communicate between independently written applications.
8. AI and Machine Learning
**AI Building Blocks**
**AutoML**: AutoML is a machine learning product suite that enables developers with limited machine learning expertise to provide their data sets and obtain access to quality trained models produced by Google's transfer learning and Neural Architecture Search (Google's technology for finding, generating, evaluating, and training numerous neural architectures to automatically select a solution for the customer's application):

- • **AutoML Natural Language**: AutoML Natural Language enables customers to categorize input text into their own custom defined labels (supervised classification). Users can customize models to their own domain or use case.
- • **AutoML Tables**: AutoML Tables enables your entire team of data scientists, analysts, and developers to automatically build and deploy state-of-the-art machine learning models on structured data at increased speed and scale.
- • **AutoML Translation**: AutoML Translation is a simple and scalable translation solution that allows businesses and developers with limited machine learning expertise to customize the Google Neural Machine Translation (GNMT) model for their own domain or use-case.
- • **AutoML Video**: AutoML Video is a simple and flexible machine learning service that lets businesses and developers easily train custom and scalable video models for their own domain or use cases.
- • **AutoML Vision**: AutoML Vision is a simple and flexible machine learning service that lets businesses and developers with limited machine learning expertise train custom and scalable vision models for their own use cases.
- • **Recommendations AI**: Recommendations AI enables you to build an end-to-end personalized recommendation system based on state-of-the-art deep learning ML models, without a need for expertise in ML or recommendation system architecture.

**Cloud Natural Language API**: Cloud Natural Language API provides powerful natural language understanding as an easy to use API. This API enables application developers to answer the following questions: 1) What are the entities referred to in the block of text?; 2) What is the sentiment (positive or negative) for this block of text?; 3) What is the language of this block of text?; and 4) What is the syntax for this block of text (including parts of speech and dependency trees)? Users can call this API by passing in a block of text or by referring to a document in Cloud Storage.

**Cloud Translation (including Cloud Translation v2 or any subsequent general availability version/release)**: Cloud Translation is a RESTful API that automatically translates text from one language to another language (e.g. French to English). You can use the API to programmatically translate text in your webpages or apps.

**Cloud Vision**: Cloud Vision enables developers to understand the content of an image by encapsulating powerful machine learning models in an easy to use API. It quickly classifies images into thousands of categories (e.g., "sailboat", "lion", "Eiffel Tower"), detects individual objects and faces within images, and finds and reads printed words contained within images. You can build metadata on your image catalog, moderate offensive content, or enable new marketing scenarios through

image sentiment analysis. You can also analyze images uploaded in the request and integrate with your image storage on Cloud Storage.
**Contact Center AI (CCAI)**: CCAI is a solution for improving the customer experience in your contact centers using AI. CCAI encompasses Dialogflow Essentials, Dialogflow Customer Experience Edition (CX), Speech-to-Text, and Text-to-Speech, and Speaker ID.
**Contact Center AI Insights:** Contact Center AI Insights helps customers extract value from their contact center data. It provides a console to explore the data, find relevant information and take action on the data. Customers can run advanced analysis within the platform to extract sentiment, topics and highlight key areas from their data.
**Dialogflow Essentials(ES):** Dialogflow is a development suite for voice and text conversational apps including chatbots and voicebots. Dialogflow is cross-platform and can connect to your own apps (on the web, Android, iOS, and IoT) or existing platforms (e.g., Telephony platforms like Genesys, Avaya, Cisco and digital platforms like Actions on Google, Facebook Messenger, Slack). Dialogflow Essentials Edition is a paid enterprise tier of Dialogflow provided under the Google Cloud Platform Terms of Service. (The free tier of Dialogflow (Dialogflow Trial Edition) is not offered via the Google Cloud Platform Terms of Service and is instead provided under the Dialogflow Trial Edition Terms of Service).
**Dialogflow Customer Experience Edition (CX)**: Dialogflow CX is an advanced development suite for creating conversational AI applications including chatbots and voicebots. It includes a visual bot building platform, collaboration and versioning tools, bot modularization tools, advanced IVR feature support (like DTMF, barge-in, etc.), and is optimized for enterprise scale and complexity. Dialogflow CX is cross-platform and can connect to your own apps (on the web, Android, iOS, and IoT) or existing platforms (e.g., telephony platforms like Genesys, Avaya, Cisco and digital platforms). Dialogflow CX is provided under the Google Cloud Platform Terms of Service.
**Document AI**: Document AI classifies and extracts structured data from documents to help discover insights and automate business processes.
• • **Human-in-the-Loop AI**: Human-in-the-Loop AI provides a user interface and workflow tools for human verification of data extracted from documents using Document AI.

**Media Translation API**: Media Translation API is a gRPC API that automatically translates audio from one language to another language (e.g., French to English) and supports streaming real time. You can use the API to programmatically translate audio in your apps.

**Speaker ID**: Speaker ID allows customers to enroll user voice prints and later verify users against a previously enrolled voice print.
**Speech-to-Text**: Speech-to-Text allows developers to convert audio to text by applying powerful neural network models in an easy to use API.
**Text-to-Speech**: Text-to-Speech synthesizes human-like speech based on input text in a variety of voices and languages.
**Video Intelligence API**: Video Intelligence API makes videos searchable, and discoverable, by extracting metadata with an easy to use REST API. It quickly annotates videos stored in Cloud Storage, and helps you identify key noun entities of your video and when they occur within the video.
9. Vertex AI, AI Platform, and Accelerators
**AI Platform Data Labeling**: AI Platform Data Labeling is a service that helps developers obtain high quality data to train and evaluate their machine learning models. It supports labeling for image, video, text, and audio as well as management of all of your labeled data in one place.
**AI Platform Deep Learning Container**: AI Platform Deep Learning Container is a Docker image with the most popular AI frameworks. Machine learning developers and data scientists can customize AI Platform Deep Learning Container and use it with Notebooks, Google Kubernetes Engine (GKE), Vertex AI, Cloud Run, Compute Engine, Kubernetes, and Docker Swarm.
**AI Platform Neural Architecture Search (NAS)**: NAS is a managed service leveraging Google's neural architecture search technology to generate, evaluate, and train numerous model architectures for a customer's application. NAS training services facilitate management of large-scale experiments.
**AI Platform Training and Prediction**: AI Platform Training and Prediction is a managed service that enables you to easily build and use machine learning models. It provides scalable training and prediction services that work on large scale datasets.
**Notebooks**: Notebooks is a managed service that offers an integrated JupyterLab environment in which machine learning developers and data scientists can create instances running JupyterLab that come pre-installed with the latest data science and machine learning frameworks in a single click.
**Vertex AI**: Vertex AI is a service for managing the entire lifecycle of AI and machine learning development. With Vertex AI, you can (i) manage image, video, text, and

tabular datasets and associated labels; (ii) build machine learning pipelines to train and evaluate models using Google Cloud algorithms or custom training code; (iii) deploy models for online or batch use cases all on scalable managed infrastructure (including additional discovery points and API endpoints for functionality replacing the legacy services of AI Platform Data Labeling, AI Platform Training and Prediction, AI Platform Neutral Architecture Search (NAS), AutoML Natural Language, AutoML Video, AutoML Vision, and AutoML Tables); and (iv) manage your entire data science workflow using Vertex AI Workbench.

10. Industry Solutions

**Talent Solution**: Talent Solution offers access to Google's machine learning, enabling company career sites, job boards, ATS, staffing agencies, and other recruitment technology platforms to improve the talent acquisition experience.

**Retail Search**: Retail Search, powered by Google's Retail API, allows retailers to leverage Google's search capabilities on their own retail websites and mobile applications. With Retail Search, retailers receive fast, accurate, and high quality search results that help improve conversion and increase customer engagement.

11. API Management

**Apigee**: Apigee is a full-lifecycle API management platform that lets customers design, secure, analyze, and scale APIs, giving them visibility and control. Apigee is available as Apigee, a fully-managed service, Apigee hybrid, a hybrid model that's partially hosted and managed by the customer, or Apigee Private Cloud, an entirely customer hosted Premium Software solution.

**API Gateway**: API Gateway is a fully-managed service that helps you develop, deploy, and secure your APIs running on Google Cloud Platform.

**Cloud Endpoints**: Cloud Endpoints is a tool that helps you to develop, deploy, secure and monitor your APIs running on Google Cloud Platform.

12. Hybrid and Multi-cloud

**Anthos**: Anthos is a solution designed for building and managing modern applications running across hybrid cloud environments. Anthos is an integrated platform incorporating cloud-based services and software components, including:

- • **Anthos Config Management**: Anthos Config Management is a policy management solution for enabling consistent configuration across multiple Kubernetes clusters. Anthos Config Management allows you to specify one single source of truth and then enforce those policies on your cluster.
- • **Anthos Identity Service**: Anthos Identity Service is an authentication service that lets customers bring existing identity solutions for authentication to multiple Anthos environments. Users can log in to and access their Anthos clusters from the command line or from the Cloud Console, all using their existing identity providers.
- • **Anthos Integration with Google Cloud Platform Services**: Google Cloud Platform services and components may be used in connection with Anthos deployments, including Google Kubernetes Engine (GKE), Cloud Logging, Cloud Monitoring, Traffic Director, and Google Cloud Platform Marketplace.
- • **Anthos Premium Software**: Anthos includes the software components listed below as Premium Software.
- • **Anthos Service Mesh**: Anthos Service Mesh is a managed service mesh service that includes (i) a managed certificate authority that issues cryptographic certificates that identify customer workloads within the Anthos Service Mesh for mutual authentication, and (ii) telemetry for customers to manage and monitor their services. Customers receive details showing an inventory of services, can understand their service dependencies, and receive metrics for monitoring their services. For clarity this service does not include Anthos Service Mesh -- Software (see below regarding Premium Software).
- • **Google Kubernetes Engine**: Google Kubernetes Engine, powered by the open source container scheduler Kubernetes, enables you to run containers on Google Cloud Platform. Kubernetes Engine takes care of provisioning and maintaining the underlying virtual machine cluster, scaling your application, and operational logistics such as logging, monitoring, and cluster health management.
- • **Connect**: Connect is a service that enables both users and Google-hosted components to interact with clusters through a connection to the in-cluster Connect software agent.
- • **Hub**: Hub is centralized control-plane that enables a user to register clusters running in a variety of environments, including Google’s cloud, on premises in customer datacenters, or other third party clouds. Hub provides a way for customers to centrally manage features and services on customer-registered clusters.

**Cloud Run for Anthos**: Cloud Run for Anthos lets you run stateless containers on Anthos.

13. Google-Managed Multi-Cloud Services

***BigQuery Omni:** BigQuery Omni is a Google-managed multi-cloud analytics solution that enables analysts to access and analyze data stored on other supported public clouds from a singular BigQuery control-plane on GCP.

Bare Metal

**Bare Metal Solution**: Bare Metal Solution allows you to operate and manage dedicated bare metal hardware (servers and attached storage) in Google's subprocessors' data centers to run specialized workloads with low latency.

Migration

**BigQuery Data Transfer Service**: BigQuery Data Transfer Service automates data movement from SaaS applications to BigQuery on a scheduled, managed basis. With the BigQuery Data Transfer Service, you can transfer data to BigQuery from SaaS applications including Google Ads, Campaign Manager, Google Ad Manager, and YouTube.

**Database Migration Service**: Database Migration Service is a fully-managed migration service that makes it simple to perform high fidelity, minimal-downtime migrations at scale. You can use Database Migration Service to migrate from your on-premises environments, Compute Engine, and other clouds to certain Google Cloud-managed databases with minimal downtime.

**Migrate for Compute Engine V5.0 and up**: Migrate for Compute Engine V5.0 and up is a fully-managed migration service that enables you to migrate workloads at scale into Google Cloud Compute Engine with minimal down time by utilizing replication-based migration technology.

**Storage Transfer Service**: Storage Transfer Service enables you to import large amounts of online data into Cloud Storage, quickly and cost-effectively. With Storage Transfer Service, you can transfer data from locations reachable by the general internet (e.g., HTTP/HTTPS), including Amazon Simple Storage Service (Amazon S3), as well as transfer data between Google Cloud products (e.g., between two Cloud Storage buckets). You can also use Storage Transfer Service to move data between private data center storage (e.g., NFS) and Google Cloud products (e.g., transfer from NFS to Cloud Storage).

**Transfer Appliance**: Transfer Appliance is a solution that uses hardware appliances and software to transfer large amounts of data quickly and cost-effectively into Google Cloud Platform.

14. Security and Identity

Security

**Access Transparency**: Access Transparency captures near real-time logs of manual, targeted accesses by Google administrators, and serves them to customers via their Cloud Logging account.

**Assured Workloads:** Assured Workloads provides functionality to create security controls that are enforced on your cloud environment. These security controls can assist with your compliance requirements (for example, FedRAMP Moderate).

**Binary Authorization**: Binary Authorization helps customers ensure that only signed and explicitly-authorized workload artifacts are deployed to their production environments. It offers tools for customers to formalize and codify secure supply chain policies for their organizations.

**Certificate Authority Service:** Certificate Authority Service is a cloud-hosted certificate issuance service that lets customers issue and manage certificates for their cloud or on-premises workloads. Certificate Authority Service can be used to create certificate authorities using Cloud KMS keys to issue, revoke, and renew subordinate and end-entity certificates.

**Certificate Manager**: Certificate Manager provides a central place for customers to control where certificates are used and how to obtain certificates, and to see the state of the certificates.

**Cloud Asset Inventory**: Cloud Asset Inventory is an inventory of cloud assets with history. It enables users to export cloud resource metadata at a given timestamp or cloud resource metadata history within a time window.

**Cloud Data Loss Prevention:** Cloud Data Loss Prevention is a fully-managed service designed to help you discover, classify, and protect your most sensitive data. You can inspect, mask, and de-identify sensitive data like personally identifiable information (PII).

**Cloud External Key Manager (Cloud EKM):** Cloud EKM lets you encrypt data in Google Cloud Platform with encryption keys that are stored and managed in a third-party key management system deployed outside Google's infrastructure.

**Cloud HSM:** Cloud HSM (Hardware Security Module) is a cloud-hosted key management service that lets you protect encryption keys and perform cryptographic operations within a managed HSM service. You can generate, use, rotate, and destroy various symmetric and asymmetric keys.
**Cloud Key Management Service:** Cloud Key Management Service is a cloud-hosted key management service that lets you manage cryptographic keys for your cloud services the same way you do on premises. You can generate, use, rotate, and destroy AES256, RSA 2048, RSA 3072, RSA 4096, EC P256, and EC P384 cryptographic keys.
**Event Threat Detection**: Event Threat Detection helps detect threats in log data. Threat findings are written to Security Command Center and optionally to Cloud Logging.
**Key Access Justifications (KAJ):** KAJ provides a justification for every request sent through Cloud EKM for an encryption key that permits data to change state from at-rest to in-use.
**Risk Manager**: Risk Manager allows customers to scan their cloud environments and generate reports around their compliance with industry-standard security best practices, including CIS benchmarks. Customers then have the ability to share these reports with insurance providers and brokers.
**Security Command Center:** Security Command Center is Google Cloud's centralized vulnerability and threat reporting service. Security Command Center provides asset inventory and discovery and allows you to identify misconfigurations, vulnerabilities and threats, helping you to mitigate and remediate risks.
**VPC Service Controls:** VPC Service Controls provide administrators the ability to configure security perimeters around resources of API based cloud services (such as Cloud Storage, BigQuery, Bigtable) and limit access to authorized VPC networks, thereby mitigating data exfiltration risks.
**Secret Manager:** Secret Manager provides a secure and convenient method for storing API keys, passwords, certificates, and other sensitive data.
**Web Security Scanner**: Web Security Scanner is a web application security scanner that enables developers to easily check for a subset of common web application vulnerabilities in websites built on App Engine and Compute Engine.
15. Identity & Access

**Access Approval**: Access Approval allows customers to approve eligible manual, targeted accesses by Google administrators to their data or workloads before those accesses happen.
**Access Context Manager**: Access Context Manager allows Google Cloud organization administrators to define fine-grained, attribute based access control for projects, apps and resources.
**BeyondCorp Enterprise**: BeyondCorp Enterprise is a solution designed to enable zero-trust application access to enterprise users and protect enterprises from data leakage, malware and phishing attacks. BeyondCorp Enterprise is an integrated platform incorporating cloud-based services and software components, including:
• • **On-premises Connector**, which forwards Identity-Aware Proxy traffic from Google Cloud Platform to applications and VMs deployed in non-Google Cloud Platform environments.
• • **Endpoint Verification**, which allows administrators to build an inventory of devices and set the security posture of the devices.
• • **Threat and Data Protection Services**, which are a set of security services that work by aggregating threat intelligence and are designed to protect enterprise users from malware transfers, phishing, malicious site visits, and sensitive data leakage.
• • **BeyondCorp Enterprise Integration with Chrome Browser Cloud Management**, which enables malware, phishing, and data leakage protection for managed Chrome browsers.
• • Other features listed at https://cloud.google.com/beyondcorp-enterprise/pricing or a successor URL.

**Cloud Identity Services**: Cloud Identity Services are the services and editions as described at: https://cloud.google.com/terms/identity/user-features.html or such other URL as Google may provide.
***Firebase Authentication**: Firebase Authentication provides a service as part of the Firebase platform to authenticate and manage users in your applications. It supports authentication using email & password, phone number and popular federated identity providers like Google and Facebook.
**Google Cloud Identity-Aware Proxy**: Google Cloud Identity-Aware Proxy is a tool that helps control access, based on a user's identity and group membership, to applications running on Google Cloud Platform.

**Identity & Access Management (IAM)**: IAM provides administrators the ability to manage cloud resources centrally by controlling who can take what action on specific resources.
**Identity Platform**: Identity Platform provides you with functionality and tools to manage your users' identities and access to your applications. Identity Platform supports authentication and management of users with a variety of methods, including email & password, phone number, and popular federated identity providers like Google and Facebook.
**Managed Service for Microsoft Active Directory (AD)**: Managed Service for Microsoft Active Directory is a Google Cloud service running Microsoft AD that enables you to deploy, configure and manage cloud-based AD-dependent workloads and applications. It is a fully-managed service that is highly available, applies network firewall rules, and keeps AD servers updated with Operating System patches.
**Resource Manager API**: Resource Manager API allows you to programmatically manage Google Cloud Platform container resources (such as Organizations and Projects), that allow you to group and hierarchically organize other Google Cloud Platform resources. This hierarchical organization lets you easily manage common aspects of your resources such as access control and configuration settings.
16. User Protection Services
**reCAPTCHA Enterprise**: reCAPTCHA Enterprise helps detect fraudulent activity on websites.
**Web Risk API**: Web Risk API is a Google Cloud service that lets client applications check URLs against Google's constantly updated lists of unsafe web resources.
Serverless Computing
**Cloud Run**: Cloud Run (fully-managed) lets you run stateless containers on a fully-managed environment.
**Cloud Functions**: Cloud Functions is a lightweight, event-based, asynchronous compute solution that allows you to create small, single-purpose functions that respond to cloud events without the need to manage a server or a runtime environment.
***Cloud Functions for Firebase**: Cloud Functions for Firebase lets you write code that responds to events and invokes functionality exposed by other Firebase

features, once you deploy JavaScript code in a hosted, private, and scalable Node.js environment that requires no maintenance.

**Cloud Scheduler**: Cloud Scheduler is a fully-managed enterprise-grade cron job scheduler. It allows you to schedule virtually any job, including batch, big data jobs, cloud infrastructure operations, and more. You can automate everything, including retries in case of failure to reduce manual toil and intervention. Cloud Scheduler even acts as a single pane of glass, allowing you to manage all your automation tasks from one place.

**Cloud Tasks**: Cloud Tasks is a fully-managed service that allows you to manage the execution, dispatch, and delivery of a large number of distributed tasks. Using Cloud Tasks, you can perform work asynchronously outside of a user or service-to-service request. Cloud Tasks provides all the benefits of a distributed task queue such as task offloading wherein heavyweight, background and long running processes can be dispatched to a task queue, loose coupling between microservices allowing them to scale independently, and enhanced system reliability as tasks are persisted in storage and retried automatically, making your infrastructure resilient to intermittent failures.

**Eventarc**: Eventarc is a fully-managed service for eventing on Google Cloud Platform. Eventarc connects various Google Cloud services together, allowing source services (e.g., Cloud Storage) to emit events that are delivered to target services (e.g., Cloud Run or Cloud Functions).

**Workflows**: Workflows is a fully-managed service for reliably executing sequences of operations across microservices, Google Cloud services, and HTTP-based APIs.

17. Internet of Things (IoT)

**IoT Core**: IoT Core is a fully-managed service that allows you to easily and securely connect, manage, and ingest data from internet connected devices. It permits utilization of other Google Cloud services for collecting, processing, analyzing, and visualizing IoT data in real time.

Management Tools

**Cloud Console App**: Cloud Console App is a native mobile app that enables customers to manage key Google Cloud services. It provides monitoring, alerting, and the ability to take actions on resources.

**Cloud Deployment Manager**: Cloud Deployment Manager is a hosted configuration tool which allows developers and administrators to provision and manage their infrastructure on Google Cloud Platform. It uses a declarative model which allows

users to define or change the resources necessary to run their applications and will then provision and manage those resources.
**Cloud Shell**: Cloud Shell is a tool that provides command-line access to cloud resources directly from your browser. You can use Cloud Shell to run experiments, execute Cloud SDK commands, manage projects and resources, and do lightweight software development via the built-in web editor.
**Recommenders**: Recommenders automatically analyze your usage patterns to provide recommendations and insights across services to help you use Google Cloud Platform in a more secure, cost-effective, and efficient manner.
**Service Infrastructure:** Service Infrastructure is a foundational platform for creating, managing, securing, and consuming APIs and services. It includes:

- • Service Management API, which lets service producers manage their APIs and services;
- • Service Consumer Management API, which lets service producers manage their relationships with their service consumers; and
- • Service Control API, which lets managed services integrate with Service Infrastructure for admission control and telemetry reporting functionality.
- • Service Usage API, which lets service consumers manage their usage of APIs and services.

18. Healthcare and Life Sciences
**Cloud Healthcare**: Cloud Healthcare is a fully-managed service to send, receive, store, query, transform, and analyze healthcare and life sciences data and enable advanced insights and operational workflows using highly scalable and compliance-focused infrastructure.
***Healthcare Data Engine (HDE)**: HDE is a solution that enables (1) harmonization of healthcare data to the Fast Healthcare Interoperability Resources (“FHIR”) standard and (2) streaming of healthcare data to an analytic environment.
19. Media and Gaming
**Game Servers**: Game Servers is a managed service that enables game developers to deploy and manage their dedicated game servers across multiple Agones clusters around the world through a single interface.

**Transcoder API**: Transcoder API can batch convert media files into optimized formats to enable streaming across web, mobile, and living room devices. It provides fast, easy to use, large-scale processing of advanced codecs while utilizing Google’s storage, networking, and delivery infrastructure.

3. What is the GCP pub-sub with examples?

Pub/Sub allows services to communicate asynchronously, with latencies on the order of 100 milliseconds.

Pub/Sub is used for streaming analytics and data integration pipelines to ingest and distribute data. It is equally effective as a messaging- oriented middleware for service integration or as a queue to parallelize tasks.

Pub/Sub enables you to create systems of event producers and consumers, called **publishers** and **subscribers**. Publishers communicate with subscribers asynchronously by broadcasting events, rather than by synchronous remote procedure calls (RPCs).

Publishers send events to the Pub/Sub service, without regard to how or when these events are to be processed. Pub/Sub then delivers events to all services that need to react to them. Compared to systems communicating through RPCs, where publishers must wait for subscribers to receive the data, such asynchronous integration increases the flexibility and robustness of the overall system.

- **Ingestion user interaction and server events.** To make use of user interaction events from end-user apps or server events from your system, you may forward them to Pub/Sub and then use a stream processing tool (such as Dataflow) which delivers them to BigQuery, Bigtable, Cloud Storage and other databases. Pub/Sub allows you to gather events from many clients simultaneously.
- **Real-time event distribution.** Events, raw or processed, may be made available to multiple applications across your team and organization for real time processing. This supports an "enterprise event bus" and event-driven application design patterns. Pub/Sub allows you to integrate with many Google systems that export events to Pub/Sub.
- **Replicating data among databases.** Pub/Sub is commonly used to distribute change events from databases. These events can be used to construct a view of the database state and state history in BigQuery and other data storage systems.
- **Parallel processing and workflows.** You can efficiently distribute a large number of tasks among multiple workers--such as compressing text files, sending email notifications, evaluating AI models, or reformatting images--by using Pub/Sub messages to connect to Cloud Functions.
- **Enterprise event bus.** You can create an enterprise-wide real-time data sharing bus, distributing business events, database updates, and analytics events across your organization.
- **Data streaming from applications, services, or IoT devices.** For example, a SaaS application can publish a real-time feed of events or a residential sensor can

stream data to Pub/Sub for use in other Google Cloud products through a Dataflow.

- **Refreshing distributed caches.** For example, an application can publish invalidation events to update the IDs of objects that have changed.
- **Load balancing for reliability.** For example, instances of a service may be deployed on Compute Engine in multiple zones but subscribe to a common topic. When the service fails in any zone, the others can pick up the load automatically.

4. Make a script to manage files stored on Google Drive.

To create and run a script in Drive, follow these steps:

1. Create/open a Google Doc or Sheet. From **Tools** menu, select **Script editor**.

2. Remove existing code and add your code. Go to **File** > **Save** > add script name > Click **OK**.

3. To execute, click ►, or from the **Run** menu, select a function. If running the script for first time, it will ask for your authentication. Check the required permissions and click *Allow*.

4. A yellow bar will appear at the centre-top, to indicate that the script is running.

5. Explain how Google's cloud billing system works?

A **Cloud Billing account** is set up in Google Cloud and is used to define who pays for a given set of Google Cloud resources and Google Maps Platform APIs. Access control to a Cloud Billing account is established by IAM roles. A Cloud Billing account is connected to a **Google payments profile**. Your Google payments profile includes a payment instrument to which costs are charged.

| Cloud Billing account | Payments Profile |
|---|---|
| A Cloud Billing account:<br>• Is a cloud-level resource managed in the Cloud Console.<br>• Tracks all of the costs (charges and usage credits) incurred by your Google Cloud usage<br>  • A Cloud Billing account can be linked to one or more projects.<br>  • Project usage is charged to the linked Cloud Billing account. | A Google Payments Profile:<br>• Is a Google-level resource managed at payments.google.com.<br>• Connects to *ALL* of your Google services (such as Google Ads, Google Cloud, and Fi phone service).<br>• Processes payments for *ALL* Google services (not just Google Cloud). |

| | |
|---|---|
| • Results in a single invoice per Cloud Billing account<br>• Operates in a single currency<br>• Defines who pays for a given set of resources<br>• Is connected to a Google Payments Profile, which includes a payment instrument, defining how you pay for your charges<br>• Has *billing-specific* roles and permissions to control accessing and modifying billing-related functions (established by IAM roles) | • Stores information like name, address, and tax ID (when required legally) of who is responsible for the profile.<br>• Stores your various payment instruments (credit cards, debit cards, bank accounts, and other payment methods you've used to buy through Google in the past.)<br>• Functions as a document center, where you can view invoices, payment history, and so on.<br>• Controls who can view and receive invoices for your various Cloud Billing accounts and products. |



6. What are the different parts of the Google Cloud Platform?

Core **service categories** include:

1. Compute
2. Networking
3. Storage and Databases
4. Artificial Intelligence (AI) / Machine Learning (ML)
5. Big Data
6. Identity and Security
7. Management Tools

7. What is Google Cloud App Engine, and how does it work? What is Google Cloud App Engine's purpose?

App Engine is a fully managed, serverless platform for developing and hosting web applications at scale. You can choose from several popular languages, libraries, and frameworks to develop your apps, and then let App Engine take care of provisioning servers and scaling your app instances based on demand.

Google App Engine (often referred to as GAE or simply App Engine) is a cloud computing platform as a service for developing and hosting web applications in Google-managed data centers. Applications are sandboxed and run across multiple servers.

8. Use an example to demonstrate how to use Google Cloud App Engine.

## 1. Overview

In this we will learn how to deploy a Python Flask web application to the App Engine flexible environment. The example application allows a user to upload a photo of a person's face and learn how likely it is that the person is happy. The application uses Google Cloud APIs for Vision, Storage, and Datastore.

### About App Engine

Google App Engine applications are easy to create, easy to maintain, and easy to scale as your traffic and data storage needs change. With App Engine, there are no servers to maintain. You simply upload your application and it's ready to go.

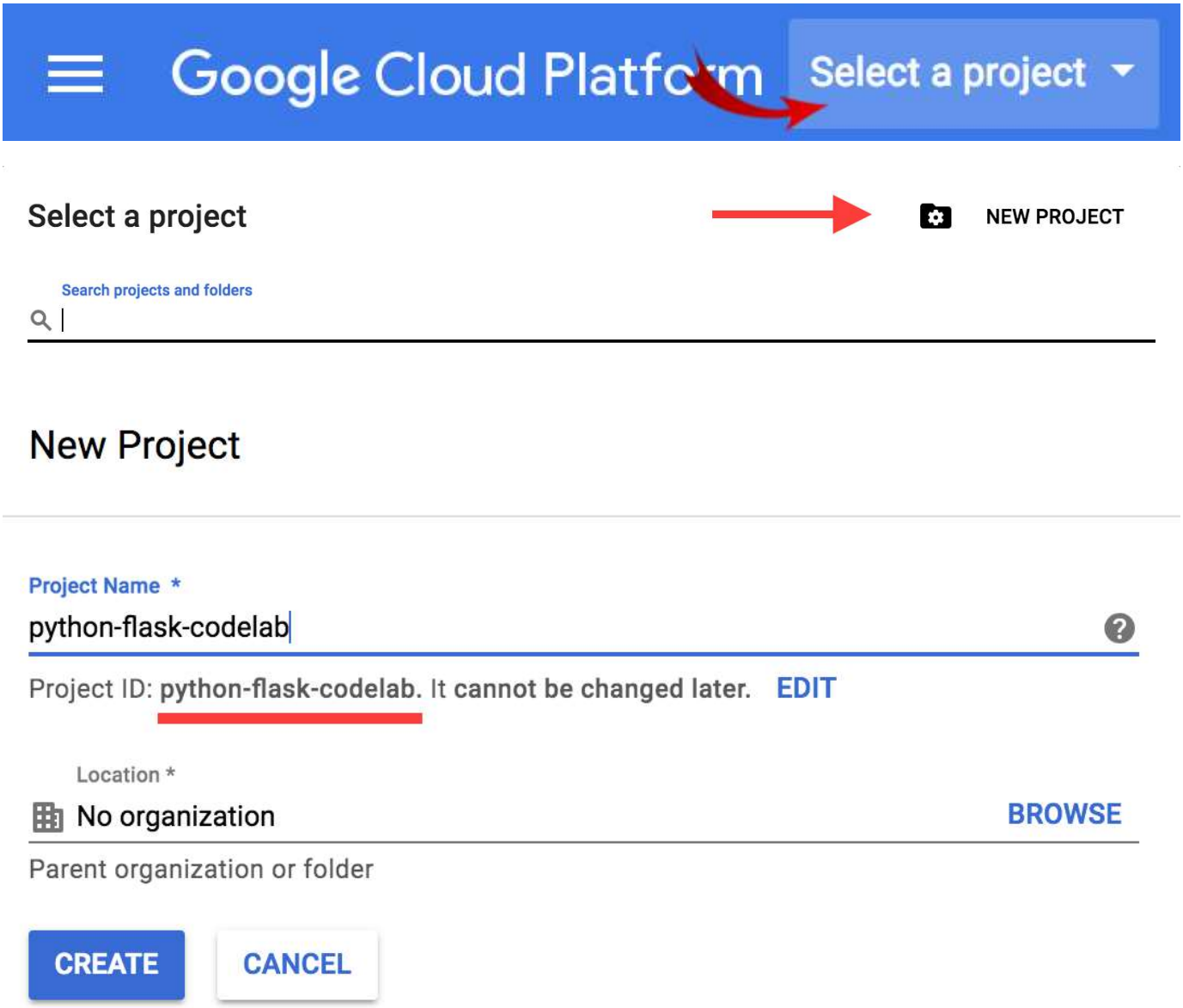
App Engine applications automatically scale based on incoming traffic. Load balancing, microservices, authorization, SQL and NoSQL databases, traffic splitting, logging, search, versioning, roll out and roll backs, and security scanning are all supported natively and are highly customizable.

App Engine's Flexible Environment supports all of the following programming languages: C#, Go, Java, Node.js, PHP, Python, and Ruby. App Engine Flexible runs your application within Docker containers running on Google Compute Engine virtual machines. App Engine's Standard Environment is an alternative option for certain languages including Python. App Engine Standard runs your application in a more restrictive sandbox environment. Read Choosing an App Engine Environment for more information.

## 2. Setup and Requirements

### Project Creation

If you don't already have a Google Account (Gmail or Google Apps), you must create one. Sign-in to Google Cloud Platform console (console.cloud.google.com) and create a new project:




Remember the project ID, a unique name across all Google Cloud projects (the name above has already been taken and will not work for you, sorry!). It will be referred to later in this codelab as *PROJECT_ID*.

### Billing

Next, you'll need to enable billing in the Cloud Console in order to use Google Cloud resources.

Running through this codelab shouldn't cost you more than a few dollars, but it could be more if you decide to use more resources or if you leave them running.

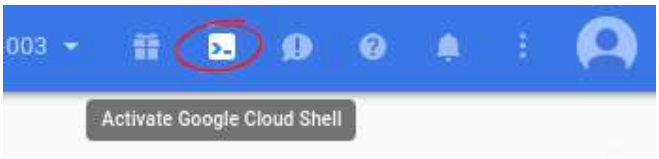
New users of Google Cloud Platform are eligible for a $300 free trial.

## 3. Google Cloud Shell

While Google Cloud can be operated remotely from your laptop, in this codelab we will be using Google Cloud Shell, a command line environment running in the Cloud. This Debian-based virtual machine is loaded with all the development tools you'll need (*gcloud*, *python*, *virtualenv*, *pip* and more), it offers a persistent 5GB home directory,

and runs on the Google Cloud, greatly enhancing network performance and authentication. This means that all you will need for this codelab is a browser (yes, it works on a Chromebook).

To activate Google Cloud Shell, from the developer console simply click the button on the top right-hand side (it should only take a few moments to provision and connect to the environment):





Once connected to the cloud shell, you should see that you are already authenticated and that the project is already set to your *PROJECT_ID*:

```
gcloud auth list
Credentialed accounts:
- <myaccount>@<mydomain>.com (active)
```

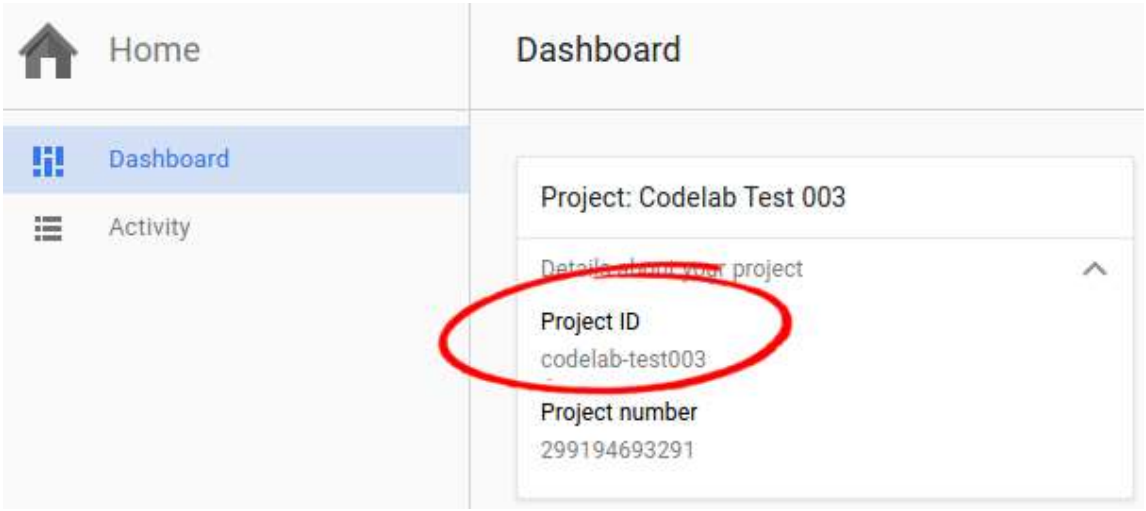
**Note:** *gcloud* is the new powerful unified command-line tool for Google Cloud Platform. Full documentation is available from https://cloud.google.com/sdk/gcloud. It comes pre-installed on CloudShell and you will surely enjoy its support for tab-completion.

```
gcloud config list project
[core]
Project = <PROJECT_ID>
```

If for some reason the project is not set, simply issue the following command:

```
gcloud config set project <PROJECT_ID>
```

Looking for your *PROJECT_ID*? Check out what Project ID you used in the setup steps or look it up in the console dashboard:



## 4. Get the sample code

In Cloud Shell on the command-line, run the following command to clone the Github repository:

```
git clone https://github.com/GoogleCloudPlatform/python-docs-samples.git
```

Change directory into *python-docs-samples/codelabs/flex_and_vision*:

```
cd python-docs-samples/codelabs/flex_and_vision
```

## 5. Enable the APIs

Before we can begin using the Vision, Storage, and Datastore APIs, you must enable the APIs with the following commands:

```
gcloud services enable vision.googleapis.com
gcloud services enable storage-component.googleapis.com
gcloud services enable datastore.googleapis.com
```

## 6. Authenticate API Requests

In order to make requests to the Vision, Storage, and Datastore APIs, you will need service account credentials. Service account credentials from your project can be generated using the *gcloud* tool.

**Note:** If you forgot your project ID, run `gcloud config list project`.

Set an environment variable for your *PROJECT_ID*, replacing *[YOUR_PROJECT_ID]* with your own project ID:

```
export PROJECT_ID=[YOUR_PROJECT_ID]
```

Create a Service Account to access the Google Cloud APIs when testing locally:

```
gcloud iam service-accounts create codelab \
  --display-name "My Codelab Service Account"
```

Give your newly created Service Account appropriate permissions:

```
gcloud projects add-iam-policy-binding ${PROJECT_ID} \
--member serviceAccount:codelab@${PROJECT_ID}.iam.gserviceaccount.com \
--role roles/owner
```

After creating your Service Account, create a Service Account key:

```
gcloud iam service-accounts keys create ~/key.json \
--iam-account codelab@${PROJECT_ID}.iam.gserviceaccount.com
```

This command generates a service account key stored in a JSON file named *key.json* in your home directory.

Using the absolute path of the generated key, set an environment variable for your service account key in the *Cloud Shell*:

```
export GOOGLE_APPLICATION_CREDENTIALS="/home/${USER}/key.json"
```

# 7. Testing the Application Locally

## Starting Your Virtual Environment and Installing Dependencies

Create an isolated Python 3 environment named *env* with *virtualenv*:

```
virtualenv -p python3 env
```

Enter your newly created *virtualenv* named *env*:

```
source env/bin/activate
```

**Note:** When you are ready to exit your virtual environment at the end of the codelab, just type deactivate into Cloud Shell.

Use *pip* to install dependencies for your project from the *requirements.txt* file:

```
pip install -r requirements.txt
```

The *requirements.txt* file is a list of package dependencies you need for your project. The above command downloaded all of these listed package dependencies to the *virtualenv*.

## Creating an App Engine App

Next, create an App Engine instance by using:

```
gcloud app create
```

A prompt will be displayed with a list of regions. Select a Region which supports App Engine Flexible for Python. You can read more about Regions and Zones here.

## Creating a Storage Bucket

First, set the environment variable *CLOUD_STORAGE_BUCKET* equal to the name of your *PROJECT_ID*. (It is generally recommended to name your bucket the same as your *PROJECT_ID* for convenience purposes).

```
export CLOUD_STORAGE_BUCKET=${PROJECT_ID}
```

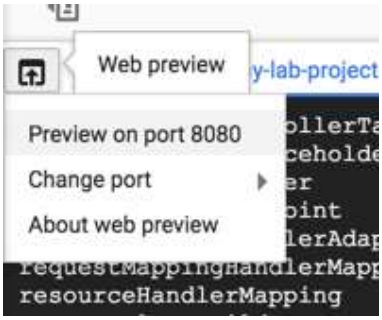
Our application uses a Cloud Storage bucket, which you will need to create from Cloud Shell with a tool called *gsutil*. Run the following command, which creates a bucket with the same name as your *PROJECT_ID*.

```
gsutil mb gs://${PROJECT_ID}
```

## Running the Application

```
python main.py
```

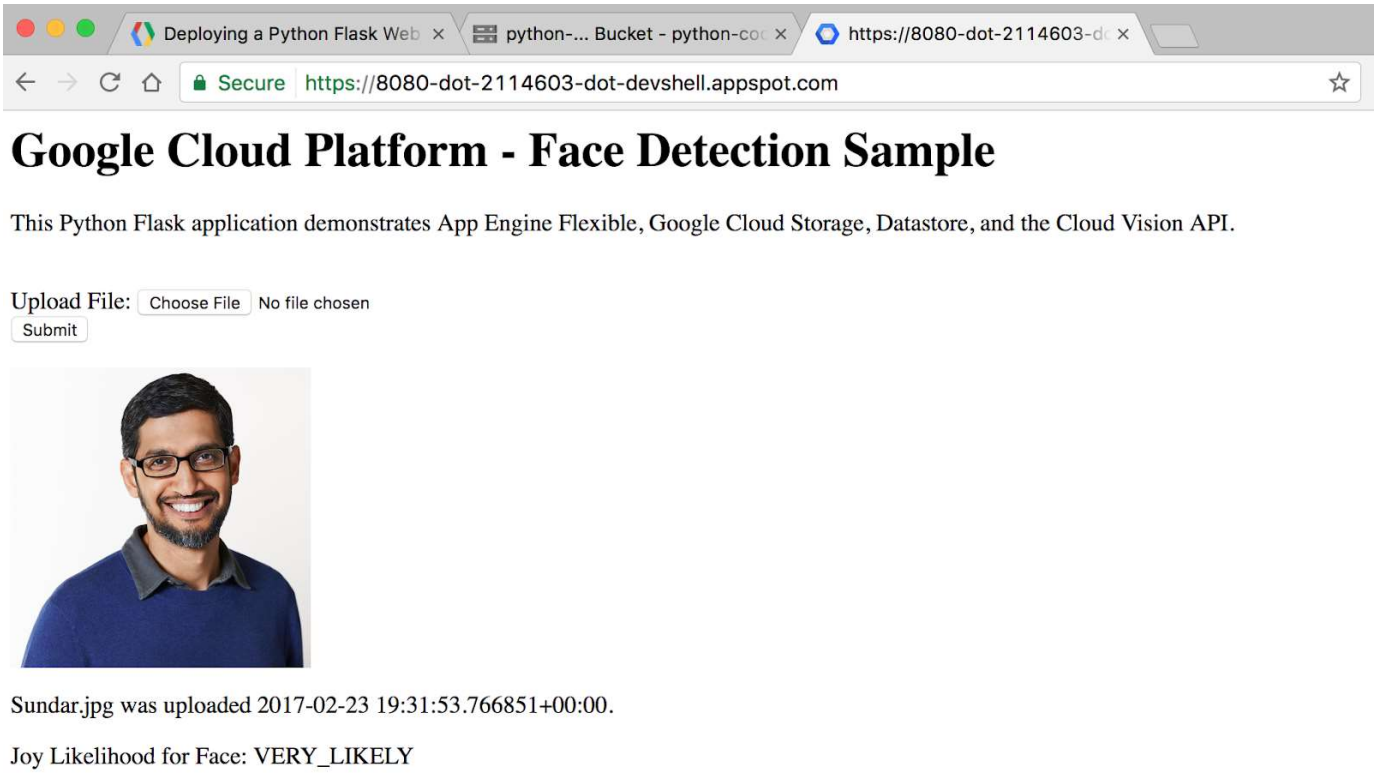
Once the application starts, click on the Web Preview icon in the Cloud Shell toolbar and choose "Preview on port 8080."




A tab in your browser opens and connects to the server you just started. You should see something like this:


Try uploading a photo that contains a human face. Click the *Choose File* button, choose an image from your computer, and then click *Submit*.

After uploading a photo, you should see something like this:



**Note:** When you are done testing your application locally, press *Control-C* on the Cloud Shell command line to shut down the local web server.

## 8. Exploring the Code

### Sample Code Layout

The sample has the following layout:

```
templates/
  homepage.html   /* HTML template that uses Jinja2 */
app.yaml          /* App Engine application configuration file */
main.py           /* Python Flask web application */
requirements.txt  /* List of dependencies for the project */
```

### main.py

This Python file is a Flask web application. The application allows users to submit photos (preferably of faces), which are stored in Cloud Storage and analyzed using the face detection feature of the Cloud Vision API. Key information about each photo is stored in Datastore, Google Cloud Platform's NoSQL database, where it is accessed each time a user visits the website.

This application uses the Google Cloud Platform client libraries for Storage, Datastore, and Vision. These client libraries make it easy to access Cloud APIs from your favorite programming languages.

Let's take a look at some key snippets of the code.

The imports section at the top is where we import the various packages we need for our code. This is how we import our Google Cloud client libraries for Datastore, Storage, and Vision:

```
from google.cloud import datastore
from google.cloud import storage
from google.cloud import vision
```

Here is the code for what happens when a user visits the root URL of the website. We create a Datastore client object, which is used to access the Datastore client library. We then run a query on Datastore for entities of kind *Faces*. Finally, we render our HTML template, passing in the *image_entities* we extract from Datastore as a variable.

```
@app.route('/')
def homepage():
    # Create a Cloud Datastore client.
    datastore_client = datastore.Client()

    # Use the Cloud Datastore client to fetch information from Datastore about
    # each photo.
    query = datastore_client.query(kind='Faces')
    image_entities = list(query.fetch())

    # Return a Jinja2 HTML template and pass in image_entities as a parameter.
    return render_template('homepage.html', image_entities=image_entities)
```

Let's take a look at how entities are saved to Datastore. Datastore is Google Cloud's NoSQL database solution. Data is stored in objects called *entities*. Each entity is assigned a unique identifying *key*, which can be created using a *kind* and a *key name* string. A *kind* is an organizational bucket for what type of *entity* it is. For example, we might want to set up *kinds* for Photos, People, and Animals.

Each *entity* can have multiple developer-defined *properties*, which can have values of a number of types, including integers, floats, strings, dates, or binary data.

```
# Create a Cloud Datastore client.
datastore_client = datastore.Client()

# Fetch the current date / time.
current_datetime = datetime.now()

# The kind for the new entity.
kind = 'Faces'

# The name/ID for the new entity.
name = blob.name

# Create the Cloud Datastore key for the new entity.
key = datastore_client.key(kind, name)

# Construct the new entity using the key. Set dictionary values for entity
# keys blob_name, storage_public_url, timestamp, and joy.
entity = datastore.Entity(key)
entity['blob_name'] = blob.name
entity['image_public_url'] = blob.public_url
entity['timestamp'] = current_datetime
entity['joy'] = face_joy

# Save the new entity to Datastore.
datastore_client.put(entity)
```

The Storage and Vision client libraries can be accessed programmatically in a similar manner to Datastore. You can open the *main.py* file yourself using *vim*, *emacs*, or *nano* to explore all of the sample code.

Learn more about Flask at http://flask.pocoo.org/.

Learn more about Client Libraries at https://googlecloudplatform.github.io/google-cloud-python/.

## homepage.html

The Flask web framework leverages Jinja2 as a template engine. This allows us to pass in variables and expressions from *main.py* into *homepage.html* that get replaced with values once the page is rendered.

Learn more about Jinja2 at http://jinja.pocoo.org/docs/2.9/templates/.

This Jinja2 HTML template displays a form for users to submit photos to the database. It also displays each previously submitted image along with its file name, upload date/time, and the likelihood that the face detected by the Vision API is happy.

homepage.html

```
<h1>Google Cloud Platform - Face Detection Sample</h1>

<p>This Python Flask application demonstrates App Engine Flexible, Google Cloud
Storage, Datastore, and the Cloud Vision API.</p>

<br>

<html>
 <body>
  <form action="upload_photo" method="POST" enctype="multipart/form-data">
   Upload File: <input type="file" name="file"><br>
   <input type="submit" name="submit" value="Submit">
  </form>

 </body>
</html>
```

## 9. Deploying the App to App Engine Flexible

App Engine Flexible uses a file called *app.yaml* to describe an application's deployment configuration. If this file is not present, App Engine will try to guess the deployment configuration. However, it is a good idea to provide this file.

Next, you will modify *app.yaml* using an editor of your choice *vim*, *nano*, or *emacs*. We will use the *nano* editor:

```
nano app.yaml
```

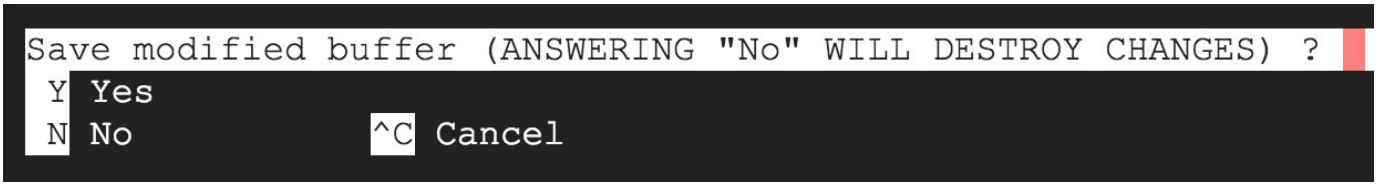
**app.yaml**

```
runtime: python
env: flex
entrypoint: gunicorn -b :$PORT main:app

runtime_config:
  python_version: 3

env_variables:
  CLOUD_STORAGE_BUCKET: <your-cloud-storage-bucket>
```

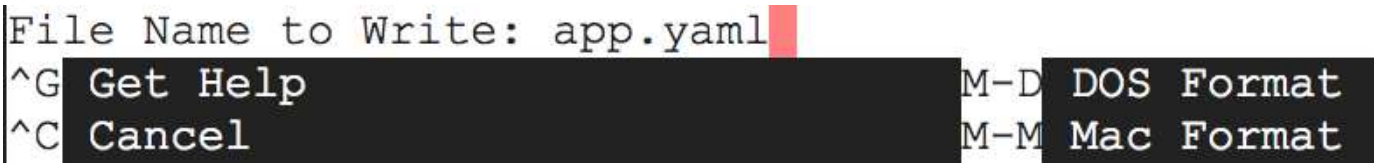
This is the basic configuration needed to deploy a Python 3 App Engine Flex application. You can learn more about configuring App Engine here.

Once you have *app.yaml* open, replace *< your-cloud-storage-bucket >* with the name of your Cloud Storage bucket. (If you forgot the name of your Cloud Storage bucket, copy the *GCP Project ID* from Qwiklabs, which is the same).
The *env_variables* section sets up environment variables that will be used in *main.py* once the application is deployed.

You can now close save and close the file in *nano* by using (**Ctrl + x**), which will prompt:




Type a letter *y* and then press the ***ENTER*** key one more time to confirm the filename for the following prompt:




Deploy your app on App Engine by using *gcloud*:

```
gcloud app deploy
```

First time deployment may take several minutes. This is because App Engine Flexible environment will automatically provision a Google Compute Engine virtual machine for you behind the scenes, and then install the application, and start it. If your deployment timed out, please let the instructor know.

After the application is deployed, you can visit it by opening the URL **https://< PROJECT_ID >.appspot.com** in your web browser.

If you forgot your *PROJECT_ID*, run `gcloud config list project` from the Cloud Shell command line.

## Summary

In this step, you set up a Python web application and deployed it to the App Engine Flexible environment.

You learned how to write and deploy your first App Engine Flexible web application!

## Clean Up

To avoid incurring charges to your Google Cloud Platform account for the resources used in this quickstart:

- Go to the Cloud Platform Console.
- Select the project you want to shut down, then click 'Delete' at the top: this schedules the project for deletion.