Google Compute Engine	Google App Engine
Compute Engine delivers configurable virtual machines running in Google's data centers with access to high-performance networking infrastructure and block storage solutions.	App Engine is a fully managed, serverless platform for developing and hosting web applications at scale.
Delivered as Infrastructure-as-a-Service (IaaS)	Delivered as Platform-as-a-Service (PaaS)
Supported Languages: Any	Supported Languages: Go, Python, Java, Node.js, PHP, Ruby (.Net and Custom runtimes for Flexible Environment)
A machine type is a set of virtualized hardware resources available to a virtual machine (VM) instance, including the system memory size, virtual CPU (vCPU) count, and persistent disk limits. In Compute Engine, machine types are grouped and curated by families for different workloads. You can choose from general-purpose, memory-optimized, and compute-optimized families.	You can run your applications in App Engine using the flexible environment or standard environment. You can also choose to simultaneously use both environments for your application and allow your services to take advantage of each environment's benefits.
You can create a collection of virtual instances and manage them as a single entity by creating instance groups. Instance groups can be managed instance groups (MIGs) or unmanaged instance groups.	Instances are the basic building blocks of App Engine, providing all the resources needed to successfully host your application. App Engine can automatically create and shut down instances as traffic fluctuates, or you can specify the number of instances to run regardless of the amount of traffic.

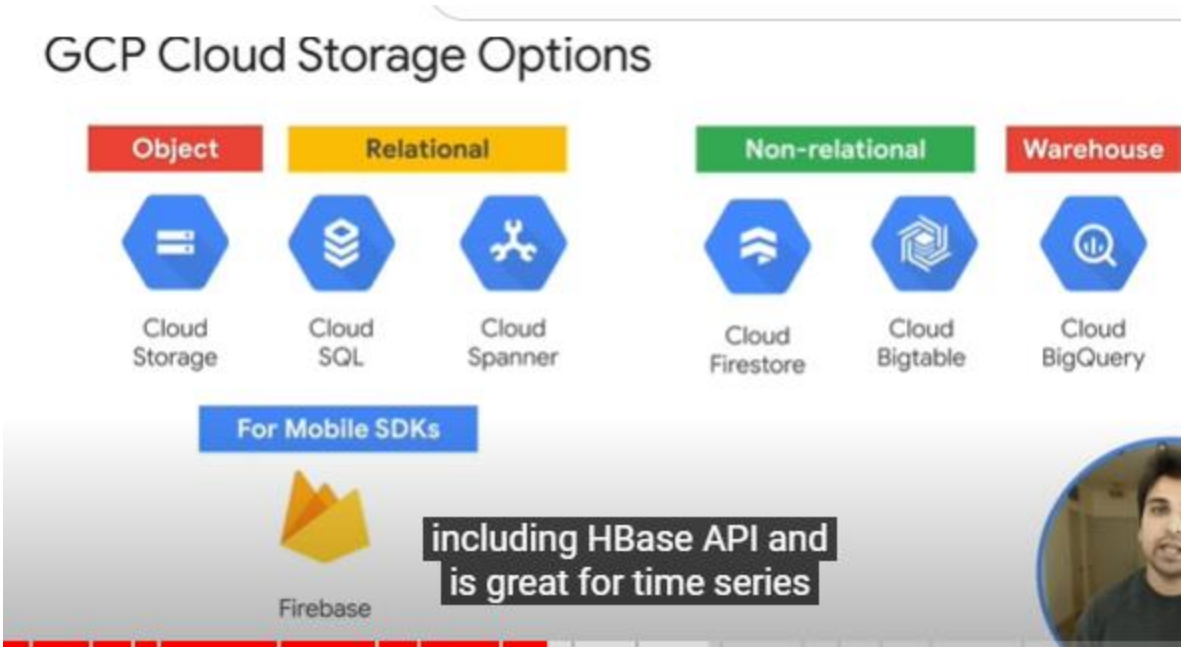
Compute Engine offers autoscaling to automatically add or remove VM instances from a managed instance group based on increases or decreases in load. Autoscaling lets your apps gracefully handle increases in traffic, and it reduces cost when the need for resources is lower.	You can specify what type of scaling you want to implement by the following -Basic Scaling-Automatic Scaling-Manual Scaling- App Engine can scale down to 0 instances when no one is using your application.
General Workloads, VM migration to Compute Engine, Genomics data processing, BYOL or use license-included images	Modern web applications, Scalable mobile back ends

https://youtu.be/j274vq9a2Rs

https://jayendrapatil.com/google-cloud-gcloud-cheat-sheet/

https://cloud.google.com/certification/guides/cloud-engineer

https://docs.google.com/forms/d/e/1FAIpQLSfexWKtXT2OSFJ-obA4iT3GmzgiOCGvjrT9OfxilWC1yPtmfQ/viewform



- Kubernetes is a production grade open-source container orchestration service for automating deployment, scaling and managing containerized workloads and services.
- Google App Engine is a managed service by Google Cloud Platform for build and run applications in the form of containers.

Difference kubernetes and Google App Engine https://www.weave.works/blog/google-app-engine-vs-kubernetes-engine

Identity and Access Management (IAM) provides predefined roles that give fine-grained access to specific Google Cloud resources and help prevent unwanted access to other resources.

IAM also lets you create *custom IAM roles*. Custom roles help you enforce the principle of least privilege, because they help to ensure that the principals in your organization have only the permissions that they need.

In IAM, you don't directly grant permissions. Instead, you grant *roles*, which contain one or more permissions.

There are several kinds of roles in IAM: basic roles, predefined roles, and custom roles.

Basic roles include three roles that existed prior to the introduction of IAM: **Owner, Editor, and Viewer.**

Predefined roles are created and maintained by Google. Google automatically updates their permissions as necessary, such as when Google Cloud adds new features or services.

Custom roles are user-defined, and allow you to bundle one or more supported permissions to meet your specific needs. Custom roles are not maintained by Google; when new permissions, features, or services are added to Google Cloud, your custom roles will not be updated automatically.

Google IAM Member Types:

- Google account – individual (me@example.com)
- Google group – (team@example.com)
- G Suite domain – (@example.com)
- Cloud Identity domain – same as G Suite domain without Google services
- Service account – JSON or P12 file for program access

- **Principal:** Google account end user or service account application or identity that needs permission to access a Google Cloud resource.
- **Role:** A collection of permissions that determine which operations can be used on a specific resource.
- **Policy:** Role constraints that bind a principal or identity to a specific action for a resource. Policies define which identities have what kind of access to an attached, specified resource.

Roles

- **Basic(Primitive roles)**: IAM basic roles are the most limited form of GCP roles and include owners, editors, and viewers.
- **Predefined**: Predefined roles provide finer-grain access to specific services in the Google Cloud.
- **Custom**: Custom roles provide finer-grain access to an organization-specific list of permissions to meet specific needs.

Service name Description Kubernetes Engine A managed environment for deploying containerized applications Compute Engine A managed environment for deploying virtual machines App Engine A managed serverless platform for deploying applications Cloud Functions A managed serverless platform for deploying event-driven functions

images are a global resource,

persistent disks are either regional or zonal resources

virtual machine (VM) instances zone

regional static external IP address

Global resources

Global resources are accessible by any resource in any zone within the same project. When you create a global resource, you don't need to provide a scope specification. Global resources include:

Addresses

The Addresses collection contains any global static external IP addresses that you have reserved for your project. Global static external IP addresses are a global resource and are used for global load balancers.

Images

Images are used by any instance or disk resource in the same project as the image. Google provides preconfigured images that you can use to boot your instance. You can customize one of these images, or you can build your own image. Optionally, you can share images across projects.

Snapshots

Persistent disk snapshots are available to all disks within the same project as the snapshot. Optionally, you can share snapshots across projects.

Instance templates

An instance template can be used to create VM instances and managed instance groups. An instance template is a global resource. However, you can specify some zonal resources in an instance template, which restricts the use of that template to the location of the specified zonal resource.

Cloud Interconnects

A Cloud Interconnect is a highly available connection from your on-premises network to Google's network. This connection is a global resource. However, interconnect attachments, which run inside of this connection, are regional resources.

Cloud Interconnect locations

A Cloud Interconnect location is a physical connection point for Cloud Interconnect near your network. There is one Cloud Interconnect location for every available colocation facility and edge availability domain. Cloud Interconnect locations are read-only, global resources.

VPC network

A VPC network is a global resource, but individual subnets are regional resources.

Firewalls

Firewalls apply to a single VPC network and are considered a global resource because packets can reach them from other networks.

Routes

Routes let you create complex networking scenarios. You can manage how traffic is routed for a specific IP range. Routes are similar to how a router directs traffic within a local area network. Routes apply to VPC networks within a Google Cloud project and are considered global resources.

Global operations

Operations are a per-zone resource, a per-region resource, and a global resource. If you are performing an operation on a global resource, the operation is considered a global operation. For example, inserting an image is considered a global operation because images are a global resource.

Note: Operations are unique in that they span all three scopes: global resources, regional resources, and zonal resources. A request to list operations returns operations across all three scopes.

Regional resources

Regional resources are accessible by any resources within the same region. For example, if you reserve a static external IP address in a specific region, that static external IP address can only be assigned to instances within that region. Each region also has one or more zones. For a list of available regions and zones, see Regions and zones.

Regional resources include:

Addresses

The Addresses collection contains any regional static external IP addresses that you have reserved for your project. Static external IP addresses are a regional resource that are used by instances that are in the same region as the address, by regional forwarding rules for regional load balancers, and for protocol forwarding.

Cloud Interconnect attachments

An interconnect attachment allocates a VLAN on your Cloud Interconnect and connects that VLAN to a VPC network. An attachment is a regional resource, but a Cloud Interconnect connection is a global resource.

Subnets

Subnets regionally segment the network IP space into prefixes (subnets) and control which prefix an instance's internal IP address is allocated from.

Regional managed instance groups

Regional managed instance groups are collections of identical instances that span multiple zones. Regional managed instance groups let you spread app load across multiple zones, rather than confining your app to a single zone or having to manage multiple instance groups across different zones.

Regional persistent disks

Regional persistent disks provide durable storage and replication of data between two zones within the same region. In a failover situation, you can force-attach a regional persistent disk to another instance within the same region. You cannot force attach a zonal persistent disk to an instance. Optionally, you can share disk resources across projects, which lets other projects make images and snapshots from these disks but doesn't let instances in other projects attach the disks.

Regional operations

Operations are a per-zone resource, a per-region resource, and a global resource. If you are performing an operation on a regional resource, the operation is considered a per-region operation. For example, reserving an address is considered regional operation because addresses are a region-specific resource.

Note: Operations are unique in that they span all three scopes: global resources, regional resources, and zonal resources. A request to list operations returns operations across all three scopes.

Zonal resources

Resources that are hosted in a zone are called *per-zone resources*. Zone-specific resources, or per-zone resources, are unique to that zone and are only usable by other resources in the same zone. For example, an instance is a per-zone resource. When you create an instance, you must provide the zone where the instance is located. The instance can access other resources within the same zone, and can access global resources, but it can't access other per-zone resources in a different zone, such as a disk resource.

Instances

A virtual machine (VM) instance is located within a zone and can access global resources or resources within the same zone.

Persistent disks

Persistent disks are accessed by other instances within the same zone. You can attach a disk only to instances in the same zone as the disk. You can't attach a disk to an instance in another zone. Optionally, you can share disk resources across projects, which lets other projects make images and snapshots from these disks but doesn't let instances in other projects attach the disks.

Machine types

Machine types are per-zone resources. Instances and disks can only use machine types that are in the same zone.

Zonal managed instance groups

A zonal managed instance group uses an instance template to create a group of identical instances within a single zone. You manage VM instances in a managed instance group as a single entity, rather than managing individual instances.

Cloud TPUs

> TPUs are zonal resources. For information about the zones in which TPUs are available, see Availability.

Per-zone operations

> Operations are a per-zone resource, a per-region resource, and a global resource. If you are performing an operation on a zone-specific resource, the operation is considered a per-zone operation. For example, inserting an instance is considered a per-zone operation because the operation is being performed on a zone-specific resource, an instance.

gcp-examquestions.com/gcp-associate-cloud-engineer-practice-exam-part-1/

https://cloud.google.com/iam/docs/understanding-roles

++

Google Compute Engine

Cloud is a collection of services that helps developers focus on their project rather than on the infrastructure that powers it.

Amazon Elastic Compute Cloud (EC2) or Google Compute Engine (GCE), which provide APIs to provision virtual servers.

Google, Amazon, Microsoft, Rackspace, DigitalOcean, and more. With so many competitors in the space, each of these companies must have its own take on how to best serve customers.

#A zone is the smallest unit in which a resource can exist. a single facility that holds lots of computers (like a single data center)

#a collection of zones is called a region, and this corresponds loosely to a city. multiple zones in a single region.
For example, if the zone is **us-central1-a,** the region would be **us-central1**.

Multiregional—A multiregional service is a composition of several different regional services. If some sort of catastrophe occurs that takes down an entire region, your service should still continue to run with minimal downtime

#Global—A global service is a special case of a multiregional service. With a global service, you

typically have deployments in multiple regions, example (VPC

#Compute Engine offers scale, performance, and value that lets you easily launch large compute clusters on Google's infrastructure.

#**Machine type**: Every machine series has predefined machine types that provide a set of resources for your VM.

- General-purpose —best price-performance ratio for a variety of workloads.

- Compute-optimized —highest performance per core on Compute Engine and optimized for compute-intensive workloads.
- Memory-optimized —ideal for memory-intensive workloads, offering more memory per core than other machine families, with up to 12 TB of memory.
- Accelerator-optimized —ideal for massively parallelized Compute Unified Device Architecture (CUDA) compute workloads, such as machine learning (ML) and high performance computing (HPC). This family is the best option for workloads that require GPUs.

https://cloud.google.com/compute/docs/machine-types

https://cloud.google.com/compute/docs/instances/instance-life-cycle

A VM can be in one of the following states:

- **PROVISIONING**: resources are allocated for the VM. The VM is not running yet.
- **STAGING**: resources are acquired, and the VM is preparing for first boot.
- **RUNNING**: the VM is booting up or running.
- **STOPPING**: the VM is being stopped. You requested a stop, or a failure occurred. This is a temporary status after which the VM enters the **TERMINATED** status.
- **REPAIRING**: the VM is being repaired. Repairing occurs when the VM encounters an internal error or the underlying machine is unavailable due to maintenance. During this time, the VM is unusable. If repair succeeds, the VM returns to one of the above states.
- **TERMINATED**: the VM is stopped. You stopped the VM, or the VM encountered a failure. You can restart or delete the VM.
- **SUSPENDING**: The VM is in the process of being suspended. You suspended the VM.
- **SUSPENDED**: The VM is in a suspended state. You can resume the VM or delete it.

Network of 34 regions, 103 zones in 200+ countries and territories with uptime of 99.99%

#Use a single VPC to span multiple regions without communicating across the public internet.

VPC network	**VPC can automatically set up your virtual topology, configuring prefix ranges for your subnets and network policies, or you can configure your own. You can also expand CIDR ranges without downtime.**
VPC flow logs	**Flow logs capture information about the IP traffic going to and from network interfaces on Compute Engine. VPC flow logs help with network monitoring, forensics, real-time security analysis, and** expense optimization. Google Cloud flow logs are updated every five seconds, providing immediate visibility.
Bring your own Ips	Bring your own IP addresses to Google's network across all regions to minimize downtime during migration and reduce your networking infrastructure cost. After you bring your own IPs, Google Cloud will advertise them globally to all peers. Your prefixes can be broken into blocks as small as 16 addresses (/28), creating more flexibility with your resources.
VPC peering	Configure private communication across the same or different organizations without bandwidth bottlenecks or single points of failure.
Firewall	Segment your networks with a globally distributed firewall to restrict access to instances. VPC Firewall Rules Logging lets you audit, verify, and analyze the effects of your firewall rules. It logs firewall access and denies events with the same responsiveness of VPC flow logs.
Routes	Forward traffic from one instance to another instance within the same network, even across subnets, without requiring external IP addresses.
Shared VPC	Configure a VPC network to be shared across several projects in your organization. Connectivity routes and firewalls associated are managed centrally. Your developers have their own projects with separate billing and quota, while they simply connect to a shared private network, where they can communicate.
Packet mirroring	Troubleshoot your existing VPCs by collecting and inspecting network traffic at scale, providing

	intrusion detection, application performance monitoring, and compliance controls with Packet Mirroring.
VPN	Securely connect your existing network to a VPC network over IPsec.
Private access	Get private access to Google services, such as storage, big data, analytics, or machine learning, without having to give your service a public IP address. Configure your application's front end to receive internet requests and shield your backend services from public endpoints, all while being able to access Google Cloud services.
VPC Service Controls	Mitigate data exfiltration risks by enforcing a security perimeter to isolate resources of multi-tenant Google Cloud services. Configure private communications between cloud resources from VPC networks spanning cloud and on-premise deployments. Keep sensitive data private and take advantage of the fully managed storage and data processing capabilities.

VPC Network Peering lets you build software as a service (SaaS) ecosystems in Google Cloud, making services available privately across different VPC networks, whether the networks are in the same project, different projects, or projects in different organizations.

With VPC Network Peering, all communication happens by using internal IP addresses. Subject to firewall rules, VM instances in each peered network can communicate with one another without using external IP addresses.

Cloud VPN lets you connect your VPC network to your physical, on-premises network or another cloud provider by using a secure virtual private network
Cloud Interconnect lets you connect your VPC network to your on-premises network by using a high speed physical connection.

VPC Network Peering lets you build software as a service (SaaS) ecosystems in Google Cloud, making services available privately across different VPC networks, whether the networks are in the same project, different projects, or projects in different organizations. Peered networks automatically exchange subnet routes for private IP address ranges. VPC Network Peering lets you configure whether the following types of routes are exchanged:

==========Book==========

Cloud Audit Logs maintain three audit logs: Admin Activity logs, Data Access logs, and System Event logs.

Constraints are the standard

Google Cloud Platform is a suite of cloud computing services that includes compute, storage, and networking services designed to meet the needs of a wide range of cloud computing customers.

A container is like a lightweight VM that isolates processes running in one container from processes running in another container on the same server.

Serverless computing is an approach that allows developers and application administrators to run their code in a computing environment that does not require setting up VMs or kubernetes clusters.

Google Cloud Platform has two serverless computing options: App Engine and Cloud Functions. App Engine is used for **applications and containers that run for extended periods of time, such as a website backend, point-of-sale system**, or custom business application. Cloud Functions is a platform for **running code in response to an event, such as uploading a file or adding a message to a message queue.** This serverless option works well when you need to **respond to an event by running a short process coded in a function** or by calling a longer-running application that might be running on a VM, managed cluster, or App Engine.

Storage

Public clouds offer a few types of storage services that are useful for a wide range of application requirements. These types include the following:

■ **Object storage** ■ File storage ■ Block storage ■ Caches

Objects can be uploaded without concern for the amount of space available on a disk. Multiple copies of objects are stored to improve availability and durability. In some cases, copies of objects may be stored in different regions to ensure availability even if a region becomes inaccessible.

advantage of object storage is that it is serverless. Google Cloud Platform's object storage, called Cloud Storage, is accessible from servers running in GCP as well as from other devices with Internet access.

Access controls can be applied at the object level. This allows users of cloud storage to control which users can access and update objects.

File storage services provide a hierarchical storage system for files. File systems storage provides network shared file systems. Google Cloud Platform has a file storage service called Cloud Filestore, which is based on the Network File System (NFS) storage system. **File storage is suitable for applications that require operating system–like file access to files**.

Block Storage Block storage uses a fixed-size data structure called a block to organize data. block storage system, you can install file systems on top of the block storage, or you can run applications that

access blocks directly. Some relational databases can be designed to access blocks directly rather working through file systems.

Block storage is available on disks that are attached to VMs in Google Cloud Platform. Block storage can be either persistent or ephemeral. A persistent disk continues to exist and store data even if it is detached from a virtual server or the virtual server to which it is attached shuts down. Ephemeral disks exist and store data only as long as a VM is running. Ephemeral disks store operating system files and other files and data that are deleted when the VM is shut down. Persistent disks are used when you want data to exist on a block storage device independent of a VM. These disks are good options when you have data that you want available independent of the lifecycle of a VM, and support fast operating system– and file system–level access.

Persistent disks can be attached to VMs in Google Compute Engine (GCE) and Google Kubernetes Engine (GKE). Since persistent disks are block storage devices, you can create file systems on these devices. Persistent disks are not directly attached to physical servers hosting your VMs but are network accessible. VMs can have locally attached solid-state drives (SSDs), but the data on those drives is lost when the VM is terminated.

Both SSD and HDD disks can be up to 64TB Persistent disks automatically encrypt data on the disk.

Caches Caches are in-memory data stores that maintain fast access to data. The time it takes to retrieve data is called latency. The latency of in-memory stores is designed to be submillisecond. To give you a comparison, here are some other latencies: ■ Making a main memory reference takes 100 nanoseconds, or 0.1 microsecond ■ Reading 4KB randomly from an SSD takes 150 microseconds ■ Reading 1MB sequentially from memory takes 250 microseconds.

“Object storage is designed for highly reliable and durable storage of objects, such as images or data sets. Object storage has more limited functionality than file system–based storage systems. File system–based storage provides hierarchical directory storage for files and supports common operating system and file system functions. File system services provide network-accessible file systems that can be accessed by multiple servers. Block storage is used for storing data on disks. File systems and databases make use of block storage systems. Block storage is used with persistent storage devices, such as SSDs and HDDs. Caches are in-memory data stores used to minimize the latency of retrieving data. They do not provide persistent storage and should never be considered a “system of truth.””

External IP addresses can be either static or ephemeral. Static addresses are assigned to a device for extended periods of time. Ephemeral external IP addresses are attached to VMs and released when the VM is stopped.

Elastic Resource Allocation differentiator between on-premise and public cloud computing is the ability to add and remove compute and storage resources on short notice.

Managed services and serverless options are good choices when you do not need control over the computing environment and will get more value from not having to manage compute resources.

Users can choose the operating system to run, which packages to install, and when to back up and perform other maintenance operations. This type of computing service is typically referred to as

infrastructure as a service (IaaS). An alternative model is called platform as a service (PaaS), which provides a runtime environment to execute applications without the need to manage underlying servers, networks, and storage systems. IaaS computing product is called Compute Engine, and the PaaS offerings are App Engine and Cloud Functions

VM preemptible, means you may be charged significantly less for the VM than normal (around 80 percent less), but your VM could be shut down at any time by Google. It will frequently be shut down if the preemptible VM has run for at least 24 hours.

App Engine manages the underlying computing and network infrastructure. There is no need to configure VMs or harden networks to protect your application. App Engine is well suited for web and mobile backend applications

App Engine is available in two types: standard and flexible

Cloud Functions Google Cloud Functions is a lightweight computing option that is well suited to event-driven processing. Cloud Functions runs code in response to an event, like a file being uploaded to Cloud Storage or a message being written to a message queue. The code that executes in the Cloud Functions environment must be short-running—this computing service is not designed to execute long-running code. If you need to support long-running applications or jobs, consider Compute Engine, Kubernetes Engine, or App Engine.

Persistent disks are storage service that are attached to VMs in Compute Engine or Kubernetes Engine. Persistent disks provide block storage on solid-state drives (SSDs) and hard disk drives (HDDs). SSDs are often used for low-latency applications where persistent disk performance is important. SSDs cost more than HDDs,

Persistent disks can be up to 64TB in size using either SSDs or HDDs.

Cloud Storage for Firebase Mobile app developers may find Cloud Storage for Firebase to be the best combination of cloud object storage and the ability to support uploads and downloads from mobile devices with sometimes unreliable network connections.

Cloud Filestore service provides a shared file system for use with Compute Engine and Kubernetes Engine. Filestore can provide high numbers of input-output operations per second (IOPS) as well as variable storage capacity. File system administrators can configure Cloud Filestore to meet their specific IOPS and capacity requirements. Filestore implements the Network File System (NFS) protocol so system administrators can easily mount shared file systems on virtual servers.

========== Databases==========

Cloud SQL

Cloud SQL is GCP's managed relational database service that allows users to set up MySQL or PostgreSQL databases on VMs without having to attend to database administration tasks, such as backing up databases or patching database software.

This database service includes management of replication and allows for automatic failover, providing for highly available databases.

Relational databases are well suited to applications with relatively consistent data structure requirements. For example, a banking database may track account numbers, customer.

Cloud Spanner Cloud Spanner is Google's globally distributed relational database that combines the key benefits of relational databases, such as strong consistency and transactions, with the ability to scale horizontally like a NoSQL database. Spanner is a high availability database with a 99.999 percent availability Service Level Agreements (SLA), making it a good option for enterprise applications that demand scalable, highly available relational database services. Cloud Spanner also has enterprise-grade security with encryption at rest and encryption in transit, along with identity-based access controls. Cloud Spanner supports ANSI 2011 standard SQL.

========== NoSQL model==========

Cloud Bigtable Cloud Bigtable is designed for petabyte-scale applications that can manage up to billions of rows and thousands of columns. It is based on a NoSQL model known as a wide-column data model, and unlike Cloud SQL that supports relational databases.

Bigtable is suited for applications that require low-latency write and read operations. It is designed to support millions of operations per second. Bigtable integrates with other Google Cloud services, such as Cloud Storage, Cloud Pub/Sub, Cloud Dataflow, and Cloud Dataproc. It also supports the Hbase API, which is an API for data access in the Hadoop big data ecosystem. Bigtable also integrates with open source tools for data processing, graph analysis, and time-series analysis.

Cloud Datastore Cloud Datastore is a NoSQL document database. This kind of database uses the concept of a document, or collection of key-value pairs, as the basic building block. Documents allow for flexible schemas. For example, a document about a book may have key-value pairs listing author, title, and date of publication. Some books may also have information about companion websites and translations into other languages. Cloud Datastore is accessed via a REST API that can be used from applications running in Compute Engine, Kubernetes Engine, or App Engine. This database will scale automatically based on load. It will also shard, or partition, data as needed to maintain performance. Since Cloud Datastore is a managed service, it takes care of replication, backups, and other database administration tasks.

Cloud Firestore is another GCP-managed NoSQL database service designed as a backend for highly scalable web and mobile applications. A distinguishing feature of Cloud Firestore is its client libraries that provide offline support, synchronization, and other features for managing data across mobile devices, IoT devices, and backend data stores. For example, applications on mobile devices can be updated in real time as data in the backend changes.

Cloud Memorystore Cloud Memorystore is an in-memory cache service. Other databases offered in GCP are designed to store large volumes of data and support complex queries, but Cloud Memorystore is a

managed Redis service for caching frequently used data in memory. Caches like this are used to reduce the time needed to read data into an application. Cloud Memorystore is designed to provide submillisecond access to data.

==========VPC==========

Understand virtual private clouds. A VPC is a logical isolation of an organization's cloud resources within a public cloud. In GCP, VPCs are global; they are not restricted to a single zone or region. All traffic between GCP services can be transmitted over the Google network without the need to send traffic over the public Internet. Understand load balancing. Load balancing is the process of distributing a workload across a group of servers. Load balancers can route workload based on network-level or application-level rules. GCP load balancers can distribute workloads globally. A distinguishing feature of GPC is that a VPC can span the globe without relying on the public Internet. Traffic from any server on a VPC can be securely routed through the Google global network to any other point on that network. Another advantage of the Google network structure is that your backend servers can access Google services, such as machine learning or IoT services, without creating a public IP address for backend servers.

VPC is global, enterprises can use separate projects and billing accounts to manage different departments or groups within the organization. Firewalls can be used to restrict access to resources on a VPC as well.

Cloud Load Balancing Google provides global load balancing to distribute workloads across your cloud infrastructure. Using a single multicast IP address, Cloud Load Balancing can distribute the workload within and across regions, adapt to failed or degraded servers, and autoscale your compute resources to accommodate changes in workload.

Cloud Load Balancing is a software service that can load-balance HTTP, HTTPS, TCP/ SSL, and UDP traffic.

Cloud Armor Services exposed to the Internet can become targets of distributed denial-of-service (DDoS) attacks. Cloud Armor is a Google network security service that builds on the Global HTTP(s) Load Balancing service. Cloud Armor features include the following: ■ Ability to allow or restrict access based on IP address ■ Predefined rules to counter cross-site scripting attacks ■ Ability to counter SQL injection attacks ■ Ability to define rules at both level 3 (network) and level 7 (application) ■ Allows and restricts access based on the geolocation of incoming traffic.

Cloud CDN With content delivery networks (CDNs), users anywhere can request content from systems distributed in various regions. CDNs enable low-latency response to these requests by caching content on a set of endpoints across the globe. Google currently has more than 90 CDN endpoints that are managed as a global resource, so there is no need to maintain region-specific configurations.

Cloud Interconnect Cloud Interconnect is a set of GCP services for connecting your existing networks to the Google network. Cloud Interconnect offers two types of connections: interconnects and peering. Interconnect with direct access to networks uses the Address Allocation for Private Internets standard (RFC 1918) to connect to devices in your VPC. A direct network connection is maintained between an on-premise or hosted data center and one of Google's colocation facilities, which are located in North America, South America, Europe, Asia,

Cloud DNS is designed to automatically scale so customers can have thousands and millions of addresses without concern for scaling the underlying infrastructure. Cloud DNS also provides for private zones that allow you to create custom names for your VMs if you need those.

==========Identity Management==========

GCP's Cloud Identity and Access Management (IAM) service enables customers to define fine-grained access controls on resources in the cloud. IAM uses the concepts of users, roles, and privileges.

Identities are abstractions about users of services, such as a human user. After an identity is authenticated by logging in or some other mechanism, the authenticated user can access resources and perform operations based on the privileges granted to that identity.

==========Management Tools ==========

The following are some of the most important tools in the management tools category:

Stackdriver This is a service that collects metrics, logs, and event data from applications and infrastructure and integrates the data so DevOps engineers can monitor, assess, and diagnose operational problems. Monitoring This extends the capabilities of Stackdriver by collecting performance data from GCP, AWS resources, and application instrumentation, including popular open source systems like NGINX, Cassandra, and Elasticsearch.

Logging This service enables users to store and analyze and alert on log data from both GCP and AWS logs.

Error Reporting This aggregates application crash information for display in a centralized interface.

Trace This is a distributed tracing service that captures latency data about an application to help identify performance problem areas.

Debugger This enables developers to inspect the state of executing code, inject commands, and view call stack variables.

Profiler This is used to collect CPU and memory utilization information across the call hierarchy of an application. Profiler uses statistical sampling to minimize the impact of profiling on application performance. The combination of management tools provides insights into applications as they run in production, enabling more effective monitoring and analysis of operational systems.

==========Data Analytics ==========

GCP has a number of services designed for analyzing big data in batch and streaming modes. Some of the most important tools in this set of services include the following:

■ BigQuery, a petabyte-scale analytics database service for data warehousing

■ Cloud Dataflow, a framework for defining batch and stream processing pipelines

■ Cloud Dataproc, a managed Hadoop and Spark service

■ Cloud Dataprep, a service that allows analysts to explore and prepare data for analysis Often, data analytics and data warehousing projects use several of these services together

==========

An organization is the root of the resource hierarchy and typically corresponds to a company or organization. G-suite domains and a Cloud Identity accounts map to GCP organizations. G Suite is Google's office productivity suite, which includes Gmail, Docs, Drive, Calendar, and other services. If your company uses G Suite, you can create an organization in your GCP hierarchy. If your company does not use G Suite, you can use Cloud Identity,

A single cloud identity is associated with at most one organization. Cloud identities have super admins, and those super admins assign the role of Organization Administrator Identity and Access Management (IAM) role to users who manage the organization. In addition, GCP will automatically grant Project Creator and Billing Account Creator IAM roles to all users in the domain. This allows any user to create projects and enable billing for the cost of resources used

Folders are the building blocks of multilayer organizational hierarchies. Organizations contain folders. Folders can contain other folders or projects. A single folder may contain both folders and projects

Project Projects are in some ways the most important part of the hierarchy. It is in projects that we create resources, use GCP services, manage permissions, and manage billing options. The first step in working with a project is to create one. Anyone with the resourcemanager .projects.create IAM permission can create a project. By default, when an organization is created, every user in the domain is granted that permission.

Organization Policies GCP provides an Organization Policy Service. This service controls access to an organization's resources. The Organization Policy Service complements the IAM service. The IAM lets you assign permissions so users or roles can perform specific operations in the cloud.

One way to think of the difference is that IAM specifies who can do things, and the Organization Policy Service specifies what can be done with resources.

All folders and projects below the organization will inherit that policy. Since policies are inherited and cannot be disabled or overridden by objects lower in the hierarchy, this is an effective way to apply a policy across all organizational resources

Policies are managed through the Organization Policies form in the IAM & admin

Sometimes it is helpful to have applications or VMs act on behalf of a user or perform operations that the user does not have permission to perform. you may have an application that needs to access a

database, but you do not want to allow users of the application to access the database directly. Instead, all user requests to the database should go through the application. A service account can be created that has access to the database. That service account can be assigned to the application so the application can execute queries on behalf of users without having to grant database access to those users.

Service accounts are somewhat unusual in that we sometimes treat them as resources and sometime as identities. When we assign a role to a service account, we are treating it as an identity. When we give users permission to access a service account, we are treating it as a resource.

There are two types of service accounts, user-managed service accounts and Googlemanaged service accounts. Users can create up to 100 service accounts per project. When you create a project that has the Compute Engine API enabled, a Compute Engine service account is created automatically. Similarly, if you have an App Engine application in your project, GCP will automatically create an App Engine service account.

Service accounts can be managed as a group of accounts at the project level or at the individual service account level. For example, if you grant iam.serviceAccountUser to a user for a specific project, then that user can manage all service accounts in the project.

Billing accounts store information about how to pay charges for resources used. A billing account is associated with one or more projects. All projects must have a billing account unless they use only free services.

There are two types of billing accounts: self-serve and invoiced. Self-serve accounts are paid by credit card or direct debit from a bank account. The costs are charged automatically. The other type is an invoiced billing account, in which bills or invoices are sent to customers. This type of account is commonly used by enterprises and other large customers

Several roles are associated with billing. It is important to know them for the exam. The billing roles are as follows:

■ Billing Account Creator, which can create new self-service billing accounts

■ Billing Account Administrator, which manages billing accounts but cannot create them

■ Billing Account User, which enables a user to link projects to billing accounts

■ Billing Account Viewer, which enables a user to view billing account cost and transactions

Cloud admins may have Billing Account Administrator to manage the accounts. Any user who can create a project should have Billing Account User so new projects can be linked to the appropriate billing account.

Billing Account Viewer is useful for some, like an auditor who needs to be able to read billing account information but not change it.

Billing Budgets and Alerts

The budget form enables you to have notices sent to you when certain percentages of your budget have been spent in a particular month.

One or more projects can be linked to a billing account, so the budget and alerts you specify should be based on what you expect to spend for all projects linked to the billing account.

By default, three percentages are set: 50 percent, 90 percent, and 100 percent.

Billing data can be exported to either a BigQuery database or a Cloud Storage file. To export billing data to BigQuery, navigate to the Billing section of the console and select Billing export from the menu.

GCP uses APIs to make services programmatically accessible. For example, when you use a form to create a VM or a Cloud Storage bucket, behind the scenes, API functions are executed to create the VM or bucket. however, are not enabled by default in a project.

Stackdriver is a set of services for monitoring, logging, tracing, and debugging applications and resources (see Figure 3.23). For monitoring and logging data to be saved into Stackdriver.

Custom images are especially useful if you have to configure an operating system and install additional software on each instance of a VM that you run.

VM instances have a zone assigned. Zones are data center– like resources, but they may be comprised of one or more closely coupled data centers. They are located within regions.

You specify a region and a zone when you create a VM.

To create Compute Engine resources in a project, users must be team members on the project or a specific resource and have appropriate permissions to perform specific tasks. Users can be associated with projects as follows: ■ Individual users ■ A Google group ■ A G Suite domain ■ A service account

Compute Engine Admin Users with this role have full control over Compute Engine instances.

Compute Engine Network Admin Users with this role can create, modify, and delete most networking resources, and it provides read-only access to firewall rules and SSL certifications. This role does not give the user permission to create or alter instances.

Compute Engine Security Admin Users with this role can create, modify, and delete SSL certificates and firewall rules. Compute Engine Viewer Users with this role can get and list Compute Engine resources but cannot read data from those resources.

An alternative way to grant permissions is to attach IAM policies directly to resources.

Limitations of Preemptible Virtual Machines

■ May terminate at any time. If they terminate within 10 minutes of starting, you will not be charged for that time. ■ Will be terminated within 24 hours. ■ May not always be available. Availability may vary across zones and regions. ■ Cannot migrate to a regular VM. ■ Cannot be set to automatically restart. ■ Are not covered by any service level agreement (SLA)

Compute Engine has more than 25 predefined machine types grouped into standard types, high-memory machines, high-CPU machines, shared core type, and memory-optimized machines. These predefined machine types vary in the number of virtual CPUs (vCPUs) and amount of memory.

App Engine users have less to manage, but they also have less control over the compute resources that are used to execute the application

App Engine also provides resident instances. These instances run continually. You can add or remove resident instances manually. When the number of deployed instances changes frequently, it can be difficult to estimate the costs of running instances. Fortunately, GCP allows users to set up daily spending limits as well as create budgets and set alarms.

App Engine provides two types of runtime environments: standard and flexible

Kubernet

The master node manages the cluster. Cluster services, such as the Kubernetes API server, resource controllers, and schedulers, run on the master. The Kubernetes API Server is the coordinator for all communications to the cluster. The master determines what containers and workloads are run on each node. When a Kubernetes cluster is created from either Google Cloud Console or a command line, a number of nodes are created as well. These are Compute Engine VMs. The default Kubernetes Engine 83 VM type is n1-standard-1 (1 vCPU and 3.75GB memory), but you can specify a different machine type when creating the cluster.

Kubernetes deploys containers in groups called pods . Containers within a single pod share storage and network resources. Containers within a pod share an IP address and port space. A pod is a logically single unit for providing a service. Containers are deployed and scaled as a unit.

Kubernetes High Availability One way Kubernetes maintains cluster health is by shutting down pods that become starved for resources. Kubernetes supports something called eviction policies that set thresholds for resources. When a resource is consumed beyond the threshold, then Kubernetes will start shutting down pods. Another way Kubernetes provides for high reliability is by running multiple identical pods. A group of running identical pods is called a deployment . The identical pods are referred to as replicas . When deployments are rolled out, they can be in one of three states. ■ Progressing, which means the deployment is in the process of performing a task

■ Completed, which means the rollout of containers is complete and all pods are running the latest version of containers

■ Failed, which indicates the deployment process encountered a problem it could not recover .

When an instance is stopped, it is not consuming compute resources, so you will not be charged.

Working with Snapshots

Snapshots are copies of data on a persistent disk. You use snapshots to save data on a disk so you can restore it. This is a convenient way to make multiple persistent disks with the same data. When you first

create a snapshot, GCP will make a full copy of the data on the persistent disk. The next time you create a snapshot from that disk, GCP will copy only the data that has changed since the last snapshot.

Working with Images

Images are similar to snapshots in that they are copies of disk contents. The difference is that snapshots are used to make data available on a disk, while images are used to create VMs. Images can be created from the following: ■ Disk ■ Snapshot ■ Cloud storage file ■ Another image

Kubernetes Engine is Google Cloud Platform's (GCP's) managed Kubernetes service. With this service, GCP customers can create and maintain their own Kubernetes clusters without having to manage the Kubernetes platform.

This collection of tasks is known as container orchestration.

Containers offer a highly-portable, light-weight means of distributing and scaling your applications or workloads, like VMs, without replicating the guest OS. They can start and stop much faster (usually in seconds) and use fewer resources. You can think of a container as similar to shipping containers for applications and workloads.

Nodes execute the workloads run on the cluster. Nodes are VMs that run containers configured to run an application. Nodes are primarily controlled by the cluster master, but some commands can be run manually. The nodes run an agent called kubelet, which is the service that communicates with the cluster master.

Kubernetes Objects

■ Pods ■ Services ■ Volumes ■ Namespaces

Each of these objects contributes to the logical organization of workloads

Pods Pods are single instances of a running process in a cluster. Pods contain at least one container. They usually run a single container, but can run multiple containers. Each pod gets a unique IP address and a set of ports. Containers connect to a port. Multiple containers in a pod connect to different ports and can talk to each other on localhost. This structure is designed to support running one instance of an application within the cluster as a pod. A pod allows its containers to behave as if they are running on an isolated VM, sharing common storage, one IP address, and a set of ports.

Compute Engine managed instance groups. A key difference is that pods are for executing applications in containers and may be placed on various nodes in the cluster, while managed instance groups all execute the same application code on 148 Chapter 7 ■ Computing with Kubernetes each of the nodes. Also, you typically manage instance groups yourself by executing commands in Cloud Console or through the command line. Pods are usually managed by a controller..

Kubernetes provides a level of indirection between applications running in pods and other applications that call them: it is called a service. A service, in Kubernetes terminology, is an object that provides API endpoints with a stable IP address that allow applications to discover pods running a particular

application. Services update when changes are made to pods, so they maintain an up-to-date list of pods running an application

ReplicaSet

A ReplicaSet is a controller used by a deployment that ensures the correct number identical of pods are running. For example, if a pod is determined to be unhealthy, a controller will terminate that pod. The ReplicaSet will detect that not enough pods for that application or workload are running and will create another. ReplicaSets are also used to update and delete pods.

Deployment Another important concept in Kubernetes is the deployment. Deployments are sets of identical pods. The members of the set may change as some pods are terminated and others are started, but they are all running the same application. The pods all run the same application because they are created using the same pod template. A pod template is a definition of how to run a pod. The description of how to define the pod is called a pod specification. Kubernetes uses this definition to keep a pod in the state specified in the template. That is, if the specification has a minimum number of pods that should be in the deployment and the number falls below that, then additional pods will be added to the deployment by calling on a ReplicaSet.

StatefulSet Deployments are well suited to stateless applications. Those are applications that do not need to keep track of their state. For example, an application that calls an API to perform a calculation on the input values does not need to keep track of previous calls or calculations. An application that calls that API may reach a different pod each time it makes a call. There are times, however, when it is advantageous to have a single pod respond to all calls for a client during a single session. StatefulSets are like deployments, but they assign unique identifiers to pods. This enables Kubernetes to track which pod is used by which client and keep them together. StatefulSets are used when an application needs a unique network identifier or stable persistent storage. Deploying Kubernetes Clusters 149 Job A job is an abstraction about a workload. Jobs create pods and run them until the application completes a workload.

Job specifications are specified in a configuration file and include specifications about the container to use and what command to run. Now that you're familiar with how Kubernetes is organized and how workloads are run, we'll cover how to deploy a Kubernetes cluster using Kubernetes Engine.

App Engine Components

App Engine Standard applications consist of four components: ■ Application ■ Service ■ Version ■ Instance.

An App Engine application is a high-level resource created in a project; that is, each project can have one App Engine application. All resources associated with an App Engine app are created in the region specified when the app is created. Apps have at least one service, which is the code executed in the App Engine environment. Because multiple versions of an application's code base can exist, App Engine supports versioning of apps. A service can have multiple versions, and these are usually slightly different, with newer versions incorporating new features, bug fixes, and other changes relative to earlier versions. When a version executes, it creates an instance of the app. Services are typically structured to perform a single function with complex applications made up of multiple services, known

as microservices. One microservice may handle API requests for data access, while another microservice performs authentication and a third records data for billing purposes

gcloud app deploy app.yml # deploy app in app Engine

gcloud app versions stop v1 v2 # stop deployment app in app Engine

Instances are created to execute an application on an App Engine managed server. App Engine can automatically add or remove instances as needed based on load. When instances are scaled based on load, they are called dynamic instances. These dynamic instances help optimize your costs by shutting down when demand is low.

These are optimized for performance so users will wait less while an instance is started. Your configuration determines whether an instance is resident or dynamic. If you configure autoscaling or basic scaling, then instances will be dynamic. If you configure manual scaling, then your instances will be resident.

To specify automatic scaling, add a section to app.yaml. Example app.yaml using basic scaling:

To specify manual scaling Example app.yaml using manual scaling

Splitting Traffic between App Engine Versions

If you have more than one version of an application running, you can split traffic between the versions. App Engine provides three ways to split traffic: by IP address, by HTTP cookie, and by random selection. IP address splitting provides some stickiness, so a client is always routed to the same split, at least as long as the IP address does not change. HTTP cookies are useful when you want to assign users to versions. Random selection is useful when you want to evenly distribute workload.

The command to split traffic is gcloud app services set-traffic. Here's an example: gcloud app services set-traffic serv1 --splits v1=.4,v2=.6

Cloud Functions is a serverless compute service provided by Google Cloud Platform (GCP). Cloud Functions is similar to App Engine in that they are both serverless. A primary difference, though, is that App Engine supports multiple services organized into a single application, while Cloud Functions supports individual services that are managed and operate independently of other services.

Cloud Functions allows developers to decouple the initial data quality check from the rest of the extraction, transformation, and load process. There are limits to Cloud Functions. By default, the functions will time out after one minute, although you can set the timeout for as long as nine minutes.

It is important to remember that object storage does not provide a file system. Buckets are analogous to directories in that they help organize objects into groups, but buckets are not true directories that support features such as subdirectories. Google does support an open source project called Cloud

Storage Fuse, which provides a way to mount a bucket as ■ Planning Storage in the Cloud fi le system on Linux and Mac operating systems.

he nearline and coldline storage classes are good options. Nearline storage is designed for use cases in which you expect to access fi les less than once per month. Coldline storage is designed, and priced, for files expected to be accessed once per year or less.

: Regional Multiregional Nearline Coldline Features Object storage replicated across multiple zones Object storage replicated across multiple regions Object storage for access less than once per month Object storage for access less than once per year Storage cost $0.20/GB/month $0.26/GB/month $0.10/GB/month $0.07/GB/month Access cost $0.01/GB $0.05/GB Use case Object storage shared across applications Global access to shared objects Older data in data lakes, backups Document retention, compliance

TABLE 11.1 Storage Services—Summary of Features

: :	**Regional**	**Multiregional**	**Nearline**	**Coldline**
Features	Object storage replicated across multiple zones	Object storage replicated across multiple regions	Object storage for access less than once per month	Object storage for access less than once per year
Storage cost	$0.20/GB/month	$0.26/GB/month	$0.10/GB/month	$0.07/GB/month
Access cost			$0.01/GB	$0.05/GB
Use case	Object storage shared across applications	Global access to shared objects	Older data in data lakes, backups	Document retention, compliance

There are three broad categories of data models available in GCP: object, relational, and NoSQL

Relational: Cloud SQL, Cloud Spanner, and BigQuery

NoSQL: Datastore, Cloud Firestore, and Bigtable ##################chatper 11 page 303

gcloud sql connect ace-exam-mysql –user=root

VPCs associated with projects, you can create a shared VPC within an organization. The shared VPC is hosted in a common project. Users in other projects who have sufficient permissions can create resources in the shared VPC. You can also use VPC peering for interproject connectivity, even if an organization is not defined.

the VPC form, which includes firewall rules, dynamic routing setting, and a DNS server policy. The Firewall Rules section lists rules that can be applied to the VPC.

Regional routing will have Google Cloud Routers learn routes within the region. Global routing will enable Google Cloud Routers to learn routes on all subnetworks in the VPC.

gcloud compute networks create ace-exam-vpc1 --subnet-mode=custom

gcloud beta compute networks subnets create ace-exam-vpc-subnet1 --network=aceexam-vpc1 --region=us-west2 --range=10.10.0.0/16 --enable-private-ip-googleaccess --enable-flow-logs

Priority: Highest-priority rules are applied; any rule with a lower priority that matches are not applied. Priority is specified by an integer from 0 to 65535. 0 is the highest priority, and 65535 is lowest.

Global load balancers are used when an application is globally distributed. Regional load balancers are used when resources providing an application are in a single region. There are three global load balancers:

■ HTTP(S), which balances HTTP and HTTPS load across a set of backend instances

■ SSL Proxy, which terminates SSL/TLS connections, which are secure socket layer connections. This type is used for non-HTTPS traffic.

■ TCP Proxy, which terminates TCP sessions at the load balancer and then forwards traffic to backend servers. The regional load balancers are as follows:

■ Internal TCP/UDP, which balances TCP/UDP traffic on private networks hosting internal VMs

■ Network TCP/UDP, which enables balancing based on IP protocol, address, and port. This load balancer is used for SSL and TCP traffic not supported by the SSL Proxy and TCP Proxy load balancers, respectively. External load balancers distribute traffic from the Internet, while internal load balancers distribute traffic that originates within GCP. The Internal TCP/UDP load balancer is the only internal load balancer. The HTTP(S), SSL Proxy, TCP Proxy, and Network TCP/UDP load balancers are all external.

This assumes the prefi x length was larger than 16 prior to issuing this command. The expand-ip-range command is used only to increase the number of addresses. You cannot decrease them, though. You would have to re-create the subnet with a smaller number of addresses.

gcloud compute networks subnets expand-ip-range ace-exam-subnet1 --prefix-length 16.

65,536, you set the prefi x length to 16:

Cloud Launcher is Google Cloud Platform's (GCP's) marketplace, where you can find preconfigured applications that are ready to deploy into the Google Cloud.Cloud Launcher is a central repository of applications and data sets that can be deployed to your GCP environment.

application is deployed from Cloud Launcher, resources such as VMs, storage buckets, and persistent disks are created automatically without additional human intervention. Deployment Manager gives cloud engineers the ability to define configuration files that describe the resources they would like to deploy.

Basic role

Name	Title	Permissions
roles/viewer	Viewer	Permissions for read-only actions that do not affect state, such as viewing (but not modifying) existing resources or data.
roles/editor	Editor	All viewer permissions, plus permissions for actions that modify state, su as changing existing resources. Note: The Editor role contains permissions to create and delete resources for most Google Cloud services. However, it does not contain permissions to perform all acti for all services. For more information about how to check whether a role has the permissions that you need, see Role types on this page.
roles/owner	Owner	All Editor permissions and permissions for the following actions: • Manage roles and permissions for a project and all resources withi the project. • Set up billing for a project.

gcloud projects add-iam-policy-binding [RESOURCE-NAME] --member user:[USEREMAIL] --role [ROLE-ID] For example, to grant the role App Engine Deployer to a user identified by jane@acexam.com, you could use this: gcloud projects add-iam-policy-binding ace-exam-project --member user:jane@ aceexam.com --role roles/appengine.deployer

IAM roles support least privilege by assigning minimal permissions to predefined roles. It also supports separation of duties by allowing some users to have the ability to change code and others to deploy code. Another common security practice is defense in depth, which applies multiple, overlapping security controls. That is also a practice that should be adopted. IAM can be applied as one of the layers of defense.

Managing Service Accounts

Service accounts are used to provide identities independent of human users. it's give VMs permission to perform tasks. Service accounts are identities that can be granted roles. Service accounts are assigned to VMs, which then use the permissions available to the service accounts to carry out tasks. Three things cloud engineers are expected to know how to do are working with scopes, assigning service accounts to VMs, and granting access to a service account to another project. It service accounts and billing accounts are not part of the resource hierarchy and not involve.

service accounts are used to provide identities independent of human users.

Service accounts are used to enable VMs to perform operations with a set of permissions. The permissions are granted to service accounts through the roles assigned to the service account. You can use the default service account provided by GCP for an instance or you can assign your own.

Scopes are permissions granted to a VM to perform some operation. Scopes authorize the access to API methods. The service account assigned to a VM has roles associated with it. To configure access controls for a VM, you will need to configure both IAM roles and scopes. We have discussed how to manage IAM roles, so now we will turn our attention to scopes.

Stackdriver is a service for collecting performance metrics, logs, and event data from our resources. Metrics include measurements such as the average percent CPU utilization over the past minute and the number of bytes written to a storage device in the last minute.

Stackdriver works in hybrid environments with support for GCP, Amazon Web Services, and on-premise resources.

Metrics are defined measurements on a resource collected at regular intervals. Metrics return aggregate values, such as the maximum, minimum, or average value of the item measured, which could be CPU utilization, amount of memory used, or number of bytes written to a network interface.

The filter criteria include VM features such as zone, region, project ID, instance ID, and labels.

The Group By parameter allows you to group time series, or data that is produced at regular intervals and has a fixed format, for example by zone, and aggregate the values so there are fewer time series to display. This is especially helpful, for example, if you want to have a group of VMs in a cluster appear as a single time series.

Configuring log sinks ■ Viewing and filtering logs ■ Viewing message details

The process of copying data from Logging to a storage system is called exporting, and the location to which you write the log data is called a sink.

The form prompts for three parameters: ■ Sink name ■ Sink service ■ Sink destination You can make up a sink name.

The sink service is one of the following: ■ BigQuery ■ Cloud Storage ■ Cloud Pub/Sub ■ Custom Destination

BigQuery, it is organized into tables based on the log name and timestamps.

When log data is exported to Cloud Storage, Logging writes a set of files to the sink bucket.

Files are organized hierarchically by log type and date. For example, if a syslog is exported on January 2, 2019, to a bucket named ace-examlog-sink1, the path to the file would be ace-exam-log-sink1/syslog/2019/01/02/

When log data is exported to Cloud Pub/Sub, the data is encoded in base64 in an object structure known as a LogEntry. LogEntries contain the logname, timestamp, textPayload, and resource properties, such as type, instance_id, zone, and project_id.

google Cloud Platform provides diagnostic tools that software developers can use to collect information about the performance and functioning of their applications. Specifically, developers can use Cloud Trace and Cloud Debug to collect data as their applications execute

Cloud Trace is a distributed tracing system for collecting latency data from an application. t Cloud Trace is a distributed tracing application that helps developers and DevOps engineers identify sections of code that are performance bottlenecks.

Cloud Debug is an application debugger for inspecting the state of a running program. Cloud Debug allows developers to insert log statements or take snapshots of the state of an application. The service is enabled by default on App Engine and can be enabled for Compute Engine and Kubernetes Engine.

Cloud Debug is used to take snapshots of the status of a program while it executes, and logpoints allow developers to inject log messages on the fly without altering source code

---

Cloud engineers are responsible for monitoring the health and performance of applications and cloud services. GCP provides multiple tools, including monitoring, logging, debugging, and tracing services. Stackdriver Monitoring allows you to define alerts on metrics, such as CPU utilization, so that you can be notified if part of your infrastructure is not performing as expected. Stackdriver Logging collects, stores, and manages log entries. Log data that needs to be stored more than 30 days can be exported to Cloud Storage, BigQuery, or Cloud Pub/Sub. Cloud Trace provides distributed tracing services to identify slow-running parts of code. Cloud Debug provides for creating snapshots of running code and injecting log messages without altering source code.

==========

Google offers several data storage services to choose from. storage and database services and highlights the storage service type, object,file, relational, non-relational, or data warehouse. In this module, we will cover Cloud Storage, Filestore, Cloud SQL, Cloud Spanner,Cloud Firestore, and Cloud Big table.

BigQuery is also listed on the right. I'm mentioning this service because it sits on the edge between data storage and data processing. But the intended use for BigQuery is Big Data Analysis and interactive querying.

you need a shared file system? If you do choose Filestore. If you don't, then choose Cloud Storage.

you will want to choose Cloud Bigtable, or BigQuery depending on your latency and update needs. Otherwise, check whether your data is relational.

If it's not relational, then choose Cloud Firestore. If it is relational, you will want to choose Cloud SQL or Cloud Spanner.

Filestore enables immediate access to data for high-performance, smart analytics without the need to lose valuable time on loading an offloading data to clients drives.

cloud SQL does not fit your requirements because you need horizontal scalability, consider using cloud spanner. Cloud spanner is a service built for the cloud specifically to combine the benefits.

Cloud spanner has schema, SQL and strong consistency. Also like a non relational database, Cloud spanner offers high availability, horizontal scalability and configurable replication, as mentioned,

Cloud spanner offers the best of the relational and non relational worlds. These features allow for mission critical use cases such as building consistent systems for transactions and inventory management in the financial services and

cloud spanner instance replicates data in end cloud zones which can be within

one region or across several regions. The database placement is configurable, meaning you can choose which region to put your database in. This architecture allows for high availability and global placement.

loud SQL, a fully managed MySQL and

PostgreSQL database service; Cloud Spanner, a relational database service with transactional consistency, global scale and high availability; Cloud Fire Store, a fully managed NoSQL document database; Cloud Bigtable, a fully managed NoSQL wide column database and Cloud Memorystore, a fully managed in-memory data store service for Redis

five different ways of connecting your infrastructure to GCP, which are dedicated interconnect, partner interconnect, Cloud VPN, direct pairing, and carrier pairing.

Cloud Load Balancing gives you the ability to distribute load-balanced compute resources in single or multiple regions to meet your high availability requirements,

GCP offers different types of load balancers that can be divided into two categories, global and regional. The global load balancers are the HTTP, HTTPS, SSL proxy and TCP proxy load balancers.

Applicable autoscaling policies include scaling based on CPU utilization, load-balancing capacity or monitoring metrics

HTTPS load balancing which acts at layer seven of the OSI model.

This is the application layer which deals with the actual content of each message

allowing for routing decisions based on the URL. GCPs HTTPS load balancing provides global load balancing for HTTPS requests destined for your instances. HTTPS load balancing balances HTTP and

HTTPS traffic across multiple back-end instances and across multiple regions. HTTP requests are load balanced on port 80 or 8080

Traces are used for performance monitoring and analysis. IAM policies are used to control access to resources, not to track which team created them.

Cloud Audit Logs maintain three audit logs: Admin Activity logs, Data Access logs, and System Event logs. There is no such thing as a Policy Access log, a User Login log, or a Performance Metric log in GCP Audit Logs.

Workload type					
General-purpose workloads			Optimized workloads		
Cost-optimized	Balanced	Scale-out optimized	Memory-optimized	Compute-optimized	Accelerator-optimized
E2	N2, N2D, N1	Tau T2D, Tau T2A	M3, M2, M1	C2, C2D	A2
Day-to-day computing at a lower cost	Balanced price/performance across a wide range of machine types	Best performance/cost for scale-out workloads	Ultra high-memory workloads	Ultra high performance for compute-intensive workloads	Optimized for high performance computing workloads
• Web serving • App serving • Back office apps • Small-medium databases • Microservices • Virtual desktops • Development environments	• Web serving • App serving • Back office apps • Medium-large databases • Cache • Media/streaming	• Scale-out workloads • Web serving • Containerized microservices • Media transcoding • Large-scale Java applications	• Medium-large OLAP and in-memory databases such as SAP HANA • In-memory databases and in-memory analytics • Microsoft SQL Server and similar databases • Genomic modeling • Electronic design automation	• Compute-bound workloads • High-performance web serving • Gaming (AAA game servers) • Ad serving • High-performance computing (HPC) • Media transcoding • AI/ML	• CUDA-enabled ML training and inference • HPC • Massive parallelized computation

Shielded VMs helps protect enterprise workloads from threats like remote attacks, privilege escalation, and malicious insiders. Shielded VMs leverage advanced platform security capabilities such as secure and measured boot, a virtual trusted platform module (vTPM), UEFI firmware, and integrity monitoring.

Spot VMs (preemptible VMs)

An instance group is a collection of virtual machine (VM) instances that you can manage as a single entity.

Compute Engine offers two kinds of VM instance groups, managed and unmanaged:

- **Managed instance groups** (MIGs) let you operate apps on multiple identical VMs. You can make your workloads scalable and highly available by taking advantage of automated MIG services, including: autoscaling, autohealing, regional (multiple zone) deployment, and automatic updating.
- **Unmanaged instance groups** let you load balance across a fleet of VMs that you manage yourself. Heterogeneous clusters can be run on unmanaged instance groups.

==========

Service accounts

A service account is a special kind of account used by an application to communication represent a non-human user that needs to authenticate and be authorized to access data in Google APIs. Applications use service accounts to make authorized API calls, authorized as either the service account itself. They are the most common way applications authenticate with Google Cloud Storage. Every project has service accounts associated with it, which may be used for different authentication scenarios, as well as to enable advanced features such as Signed URLs and browser uploads using POST.

Service accounts differ from user accounts in a few key ways:

Service accounts do not have passwords, and cannot log in via browsers or cookies.

Service accounts are associated with public/private RSA key pairs that are used for authentication to Google, and for signing data.

Service accounts do **not** belong to your Google Workspace domain, unlike user accounts. If you share Google Workspace assets, like docs or events, with your entire Google Workspace domain, they are not shared with service accounts

Preventing creation of service accounts

enable this feature, you can create service accounts in a centralized project, then attach the service accounts to resources in other projects. you can create multiple public/private RSA key pairs, known as *user-managed key pairs*, and use the private key to authenticate with Google APIs. This private key is known as a *service account key*.

Google-managed key pairs are automatically rotated and used for signing for a maximum of two weeks. The rotation process is probabilistic; usage of the new key will gradually ramp up and down over the key's lifetime.

Each service account can have up to 10 service account keys. Google stores only the public portion of a user-managed key pair.

You can add service accounts to a Google group, then grant roles to the group.

create user-managed service accounts in your project using the IAM API, the Google Cloud console, or the Google Cloud CLI. You are responsible for managing and securing these accounts. By default, you can create up to 100 user-managed service accounts in a project.

Google-managed service accounts are not listed in the **Service accounts** page in the Google Cloud console.

Default service accounts

User-managed service accounts

Firestore is the next generation of Datastore. Learn more about upgrading to Firestore.

Datastore is a highly scalable NoSQL database for your applications. Datastore automatically handles sharding and replication, providing you with a highly available and durable database that scales automatically to handle your applications' load. Datastore provides a myriad of capabilities such as ACID transactions, SQL-like queries, indexes, and much more.

Cloud Dataprep is **an intelligent data preparation service for visually exploring, cleaning, and transforming structured and unstructured data for analytics, reporting, and machine learning.**

GCP assigns regional internal IP addresses for VM instances, including GKE pods, nodes, and services. They are also used for Internal TCP/UDP Load Balancing and Internal HTTP(S) Load Balancing

---

BigQuery is a managed, petabyte scale data warehouse,(use the --dry-run option with the bq select command to save cost) which uses SQL. Bigtable does not support SQL accept billion of row sensors with IoT and time series data for petabytes. . Cloud SQL and Cloud Spanner support SQL and ANSI 2021 standard but are designed for transaction processing, not analytical applications like data warehouses.

Cloud SQL is a managed relational database service suitable for regionally used applications. Cloud Spanner is also a managed relational database but it is designed for multi-region and global applications. (transaction processing)

The main difference is that **spanner is not made for generic SQL needs**; it is for massive scale opportunities such as 1000s of writes per second globally. At the same time, Cloud SQL is a generic database used for normal storage and query purposes that do not require any software installation and maintenance

Cloud Dataproc is a managed Spark/Hadoop service, not a relational database and support both batch and streaming mode.
Cloud Dataflow is a stream analytics and batch processing service
Cloud Pub/Sub is a queuing service that is used to ingest data and store it until it can be processed and used for processing service.(for small events)

Cloud Build is a service for creating container images.
Cloud Data Fusion is a managed service that is designed for building data transformation pipelines
Cloud Dataprep is used to prepare data for analytics and machine learning
Firestore is the next generation of mobile service that can synchronize data between mobile devices and centralized storge Datastore. Learn more about upgrading to Firestore.
Firebase is **Google's mobile development platform that empowers you to quickly build and grow your app**.cloud source repositories is the service user for version control.
Cloud memorystore is the only GCP designed to cache data in memmory.
cloud source repositories is the service user for version control.
cloud natural language processing provides functionality for analyzing text cloud version is for image processing

Datastore is a highly scalable NoSQL database for your applications. It is used for document database **gcloud datastore export gs://my-datastore-backup –async**

massaging account are a fictitious option

cloud data store
VPNs are used to connect GCP networks with on premises networks.
the connected networks are in different organizations, they must use VPC Network Peering.
VPC sharing is only available within a single organization

Cloud Datastore is a schemaless NoSQL datastore in Google's cloud. Applications can use Datastore **to query your data with SQL-like queries that support filtering and sorting**. Datastore replicates data across multiple datacenters, which provides a high level of read/write availability.

Google Cloud Functions gen-2 is **a serverless execution environment for building and connecting cloud services 60 minutes for HTTP functions**. 10 minutes for event-driven functions.

Bigtable	**Bigquery**
Online Transaction Processing system	Online Analytical Processing system
NoSQL wide-column DB	Relational Structured data
Mutable & an IoT application that analyzes data	Immutable
For high-volume read/write operations	For analysis and reporting

GCP +++++++++_____Permission ++

Access is granted to Cloud Storage objects using IAM or access control lists (ACLs). When uniform bucket-level access is applied, users only have access through IAM roles and permissions. A users that could access objects before uniform bucket-level access is applied but not after must have had access through ACLs.

Cloud Storage is **a service for storing your objects in Google Cloud**. An object is an immutable piece of data consisting of a file of any format. You store objects in containers called buckets

Standard Storage: Good for “hot” data that’s accessed frequently, including websites, streaming videos, and mobile apps.

Nearline Storage: Low cost. Good for data that can be stored for at least 30 days, including data backup and long-tail multimedia content.

Coldline Storage: Very low cost. Good for data that can be stored for at least 90 days, including disaster recovery.

Archive Storage: Lowest cost. Good for data that can be stored for at least 365 days, including regulatory archives.

Create custom image from **instance, persistent disk, a snapshot, another image and in a file of google storge.**

Depricate image

the sustained used discounts are automatically applied. Remember that this does not apply on certain machine types. For example, E2 and A2.The sustained used discounts do not apply for them. And also remember that if you're using App Engine flexible or Dataflow to create the VMs, sustained used discounts do not apply

virtual machine of this type for one year or three years, and depending on the machine type and the GPUs you are asking for, you can get up to 70% discount. The discount that you would get with committed use discount is higher than the discount that you would get with sustained use discounts.

==========

https://jayendrapatil.com/tag/gcp-iam-policy-inheritance/

https://www.coursehero.com/file/60589685/Gettingstartedpdf/

https://k21academy.com/google-cloud/identity-and-access-management/

++

==========GCP-command==========

##https://cloud.google.com/sdk/docs/install-sdk#linux

https://cloud.google.com/sdk/docs/cheatsheet
https://tutorialsdojo.com/google-cloud-functions-vs-app-engine-vs-cloud-run-vs-gke/

==========CGP CLI=======gcloud, gsutil, `kubectl`
`bq`, the BigQuery command-line tool

The gsutil is a Python application that lets you access Google Cloud Storage from the command line. **The gsutil command is used only for Cloud Storage.**

The gcloud command-line interface is the primary CLI tool to create and manage Google Cloud resources. Google Compute Engine virtual machine instances and other resources, Google Cloud SQL instances, Google Kubernetes Engine clusters, Google Cloud Dataproc clusters and jobs, Google Cloud DNS managed zones and record sets, Google Cloud Deployment manager deployments.

curl -O https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-cli-398.0.0-linux-x86_64.tar.gz

tar -xf google-cloud-cli-398.0.0-linux-x86.tar.gz

./google-cloud-sdk/install.sh

./google-cloud-sdk/bin/gcloud init

gcloud init #Initialize, authorize, and configure gcloud

gcloud version #Display version and installed components

gcloud components install #Install specific components *

gcloud components update #Update the gcloud CLI to the latest version

gcloud info # Display current gcloud environment details

gcloud config set #Define a property (like compute/zone) for current configuration

gcloud config get value #Fetch value of a gcloud CLI property

gcloud config list #Display all the properties for the current configuration

gcloud config configurations create #Create a new named configuration

gcloud config configurations list #Display a list of all available configurations

gcloud config configurations activate #Switch to an existing named configuration

#List accounts whose credentials are stored on the local system:

gcloud auth list

#view the help for **gcloud compute instances create**:

gcloud help compute instances create

#List the properties in your active gcloud CLI configuration:
gcloud config list

#List instances created in zone *us-central1-a*:
gcloud compute instances list –filter="zone:us-central1-a"
gcloud compute project-info describe --project *PROJECT_ID*

view your default region and zone, run the following commands:
gcloud config get-value compute/region

gcloud compute instances list #show instance list
gcloud sql instances list #show sql instance list

gcloud compute ssh learning-cloud-demo #CONNECTING TO YOUR INSTANCE

gcloud compute instances list # Check list of instance

#bukets: Google storage is a file storage service available from Google Cloud. Quite similar to Amazon S3 it offers interesting functionalities such as signed-urls, bucket synchronization, collaboration bucket settings, parallel uploads and is S3 compatible.

#start using gsutil, you first need to authenticate it with your Google Cloud Platform account. Issue the command gcloud auth login

gsutil iam ch allUsers:objectViewer gs://free-photos-on-gcp # **allow all users to read these files**

gsutil rewrite -s nearline gs://free-photos-gcp # change the storage class to nearline

gsutil -v

gsutil mb gs://mybuketname # create buket

gsutil cp * gs://mybuketname   # copy all file from current directory to buket

#flag -m, to perform a parallel (multi-threaded/multi-processing) copy
nclude local subfolders in your upload, -r

gsutil ls gs://mybuketname # list buket conatin
gsutil rm * gs://mybuketname/* # remove the objects from the storage bucket
gsutil rb gs://mybuketname/ #deleting the Cloud Storage bucket

gsutil cp gs://<bucket_A>/<remote_file> gs://<bucket_B>/ # transfer a file between buckets

sutil cp <new_folder> gs://<bucketname>/ #Create a folder in a bucket with cp

gsutil du -h gs://<bucketname>/   # Check storage space with du

gsutil -m rsync -r -d ./myfolder gs://<bucketname> #synchronizes the content of the local folder with the storage bucket
gsutil ls -L -b gs://bucket #listing-bucket-details

gsutil ls ## list of buket
#Bucket and Bucket versioning
gsutil versioning set on gs://<bucketname>
gsutil versioning set off gs://<bucketname>

###############Lifecycle##############################
Lifecycle configurations allows you to automatically delete or change the storage class of objects when some criterion
To enable lifecycle for a bucket with settings defined in the config_file.json file, run:

$ gsutil lifecycle set <config_file.json> gs://<bucket_name> is met.

Make all files readable gsutil -m acl set -R -a public-read gs://<bucket-name>/
Config auth gsutil config -a
Grant bucket access gsutil iam ch user:denny@gmail.com:objectCreator,objectViewer gs://<bucket-name>
Remove bucket access gsutil iam ch -d user:denny@gmail.com:objectCreator,objectViewer gs://<bucket-name>

################################## Network#####################################
List all networks gcloud compute networks list #Detail of one network gcloud compute networks describe <network-name> --format json

Create network gcloud compute networks create <network-name> #Create subnet gcloud compute networks subnets create subnet1 --network net1 --range 10.5.4.0/24
Get a static ip gcloud compute addresses create --region us-west2-a vpn-1-static-ip #List all ip addresses gcloud compute addresses list
Describe ip address gcloud compute addresses describe <ip-name> --region us-central1 #List all routes gcloud compute routes list
gcloud compute zones list #List Compute Engine zones

gcloud compute instances describe # Display a virtual machine (VM) instance’s details

gcloud compute instances stop INSTANCE-NAME --async # stop compute instance The --async parameter displays information about the start operation.

gcloud compute instances list #List all VM instances in a project

gcloud compute disks snapshot #Create snapshot of persistent disks

gcloud compute snapshots describe #Display a specified snapshot’s details

gcloud compute snapshots delete #Delete a snapshot

gcloud compute ssh mv-server #Connect to a VM instance by using SSH

List all instances gcloud compute instances list,

Show instance info gcloud compute instances describe "<instance-name>" --project "<project-name>" --zone "us-Stop an instance gcloud compute instances stop instance-2

Start an instance gcloud compute instances start instance-2

Create an instance gcloud compute instances create vm1 --image image-1 --tags test --zone "<zone>" --machine-SSH to instance gcloud compute ssh --project "<project-name>" --zone "<zone-name>" "<instance-name>"

Download files gcloud compute copy-files example-instance:~/REMOTE-DIR ~/LOCAL-DIR --zone us-central1-a

Upload files gcloud compute copy-files ~/LOCAL-FILE-1 example-instance:~/REMOTE-DIR --zone us-centra

kubectl autoscale rc my-app-rc --min=2 --max=6 --cpu-percent=80 # What command would you use to implement autoscaling with these parameters?
gcloud container images list # view metadata about existing container images. What command would you use

1. gcloud compute instances create my-test-vm --source-instance-template=my-instance-template-with-custom-image
2. gcloud compute instance-groups managed list
3. gcloud compute instance-groups managed delete my-managed-instance-group
4. gcloud compute instance-groups managed create my-mig --zone us-central1-a --template my-instance-template-with-custom-image --size 1
5. gcloud compute instance-groups managed set-autoscaling my-mig --max-num-replicas=2 --zone us-central1-a
6. gcloud compute instance-groups managed stop-autoscaling my-mig --zone us-central1-a
7. gcloud compute instance-groups managed resize my-mig --size=1 --zone=us-central1-a
8. gcloud compute instance-groups managed recreate-instances my-mig --instances=my-mig-85fb --zone us-central1-a
9. gcloud compute instance-groups managed delete my-managed-instance-group --region=us-central1
10. gcloud compute instances create my-test-vm --source-instance-template=my-instance-template-with-custom-image
11. gcloud compute instance-groups managed list
12. gcloud compute instance-groups managed delete my-managed-instance-group
13. gcloud compute instance-groups managed create my-mig --zone us-central1-a --template my-instance-template-with-custom-image --size 1
14. gcloud compute instance-groups managed set-autoscaling my-mig --max-num-replicas=2 --zone us-central1-a
15. gcloud compute instance-groups managed stop-autoscaling my-mig --zone us-central1-a
16. gcloud compute instance-groups managed resize my-mig --size=1 --zone=us-central1-a
17. gcloud compute instance-groups managed recreate-instances my-mig --instances=my-mig-85fb --zone us-central1-a
18. gcloud compute instance-groups managed delete my-managed-instance-group --region=us-central1

19. kubectl expose deployment hello-server –type LoadBalancer –port 80 –target-port 8080
20. gcloud compute networks subnets describe [SUBNET_NAME] –region us-central1
21. gsutil versioning set on gs://whizlabs-bucket # enable version

New service account is created they will automatically have privilege to mange that service account iam.serviceProjectAccountUser is functional role

resourcemanager.projects.create is the tole that allow users to create project

Application Performance Management
Cloud Debugger : Investigate your code's behavior in production.
Cloud Trace :Find performance bottlenecks in production.
Cloud Profiler: Identify patterns of CPU, time, and memory consumption in production

Cloud Logging and Error Reporting
Cloud Logging : Real-time log management and analysis.
Error Reporting:Identify and understand your application errors.

https://versprite.com/blog/security-operations/google-stackdriver/#:~:text=Stackdriver%20Logging%20allows%20you%20to,API%20to%20manage%20logs%20programmatically.

http://kushagrarakesh.blogspot.com/2019/03/stackdriver-multiple-choice-questions.html?_sm_au_=iVVSWlkks2t7jt47M7BKNK07qH22M

Stackdriver integrates several technologies, including monitoring, logging, error reporting, and debugging that are commonly implemented in other environments as separate solutions using separate products. What are key benefits of integration of these services?
Answer: Reduces overhead, reduces noise, streamline use, and fixes problems faster
Stackdriver integration streamlines and unifies these traditionally independent services, making it much easier to establish procedures around them and to use them in continuous ways.

- Stackdriver
- Cloud Armor

- Cloud Security Scanner

Google Cloud Platform	Amazon Web Services[38]	Microsoft Azure[39]	Oracle Cloud[40]
Google Compute Engine	Amazon EC2	Azure Virtual Machines	Oracle Cloud Infra OCI
Google App Engine	AWS Elastic Beanstalk	Azure App Services	Oracle Application Container
Google Kubernetes Engine	Amazon Elastic Kubernetes Service	Azure Kubernetes Service	Oracle Kubernetes Service
Google Cloud Bigtable	Amazon DynamoDB	Azure Cosmos DB	Oracle NoSQL Database
Google BigQuery	Amazon Redshift	Azure Synapse Analytics	Oracle Autonomous Data Warehouse
Google Cloud Functions	AWS Lambda	Azure Functions	Oracle Cloud Fn
Google Cloud Datastore	Amazon DynamoDB	Azure Cosmos DB	Oracle NoSQL Database

Google Cloud Storage	Amazon S3	Azure Blob Storage	Oracle Cloud Storage OCI

Compute

App Engine - Platform as a Service to deploy Java, PHP, Node.js, Python, C#, .Net, Ruby and Go applications.

Compute Engine - Infrastructure as a Service to run Microsoft Windows and Linux virtual machines.

Google Kubernetes Engine (GKE) or GKE on-prem offered as part of Anthos platform[6][7] - Containers as a Service based on Kubernetes.

Cloud Functions - Functions as a Service to run event-driven code written in Node.js, Java, Python, or Go.

Cloud Run - Compute execution environment based on Knative.[8] Offered as Cloud Run (fully managed)[9] or as Cloud Run for Anthos.[9] Currently supports GCP, AWS and VMware management.[10]

Storage & Databases

Cloud Storage - Object storage with integrated edge caching to store unstructured data.

Cloud SQL - Database as a Service based on MySQL, PostgreSQL and Microsoft SQL Server.

Cloud Bigtable - Managed NoSQL database service.[11]

Cloud Spanner - Horizontally scalable, strongly consistent, relational database service.[12]

Cloud Datastore - NoSQL database for web and mobile applications.[13]

Persistent Disk - Block storage for Compute Engine virtual machines.[14]

Cloud Memorystore - Managed in-memory data store based on Redis and Memcached.[15]

Local SSD: High-performance, transient, local block storage.

Filestore: High-performance file storage for Google Cloud users.[16]

AlloyDB: Fully managed PostgreSQL database service.[17]

Networking

Google Cloud Network Topology - External Load Balancer Architecture.png

VPC - Virtual Private Cloud for managing the software defined network of cloud resources.

Cloud Load Balancing - Software-defined, managed service for load balancing the traffic.

Cloud Armor - Web application firewall to protect workloads from DDoS attacks.
Cloud CDN - Content Delivery Network based on Google's globally distributed edge points of presence.

Cloud Interconnect - Service to connect a data center with Google Cloud Platform

Cloud DNS - Managed, authoritative DNS hosting service running on the same infrastructure as Google.

Network Service Tiers - Option to choose Premium vs Standard network tier for higher-performing network.

Big Data

BigQuery - Scalable, managed enterprise data warehouse for analytics.[18]

Cloud Dataflow - Managed service based on Apache Beam for stream and batch data processing.[19]

Dataproc - Big data platform for running Apache Hadoop and Apache Spark jobs.[20]

Cloud Composer - Managed workflow orchestration service built on Apache Airflow.[21]

Cloud Datalab - Tool for data exploration, analysis, visualization and machine learning. This is a fully managed Jupyter Notebook service.[22]

Cloud Dataprep - Data service based on Trifacta to visually explore, clean, and prepare data for analysis.[23]

Cloud Pub/Sub - Scalable event ingestion service based on message queues.[24]

Cloud Data Studio - Business intelligence tool to visualize data through dashboards and reports.[25]

Cloud AI

Cloud AutoML - Service to train and deploy custom machine learning models. As of September 2018, the service is in Beta.[26]

Cloud TPU - Accelerators used by Google to train machine learning models.[27]

Cloud Machine Learning Engine - Managed service for training and building machine learning models based on mainstream frameworks.[28]

Cloud Talent Solution (formerly Cloud Job Discovery) - Service based on Google's search and machine learning capabilities for the recruiting ecosystem.[29]

Dialogflow Enterprise - Development environment based on Google's machine learning for building conversational interfaces.[30]

Cloud Natural Language - Text analysis service based on Google Deep Learning models.[31]

Cloud Speech-to-Text - Speech to text conversion service based on machine learning.[32]

Cloud Text-to-Speech - Text to speech conversion service based on machine learning.[33]

Cloud Translation API - Service to dynamically translate between thousands of available language pairs.

Cloud Vision API - Image analysis service based on machine learning.[34]

Cloud Video Intelligence - Video analysis service based on machine learning.[35]

Management Tools

Operations suite (formerly Stackdriver) - Monitoring, logging, and diagnostics for applications on Google Cloud Platform and AWS.[36]

Cloud Deployment Manager - Tool to deploy Google Cloud Platform resources defined in templates created in YAML, Python or Jinja2.[37]

Cloud Console - Web interface to manage Google Cloud Platform resources.

Cloud Shell - Browser-based shell command-line access to manage Google Cloud Platform resources.

Cloud Console Mobile App - Android and iOS application to manage Google Cloud Platform resources.

Cloud APIs - APIs to programmatically access Google Cloud Platform resources

Identity & Security

Cloud Identity - Single sign-on (SSO) service based on SAML 2.0 and OpenID.

Cloud IAM - Identity & Access Management (IAM) service for defining policies based on role-based access control.

Cloud Identity-Aware Proxy - Service to control access to cloud applications running on Google Cloud Platform without using a VPN.

Cloud Data Loss Prevention API - Service to automatically discover, classify, and redact sensitive data.

Security Key Enforcement - Two-step verification service based on a security key.

Cloud Key Management Service - Cloud-hosted key management service integrated with IAM and audit logging.

Cloud Resource Manager - Service to manage resources by project, folder, and organization based on the hierarchy.

Cloud Security Command Center - Security and data risk platform for data and services running in Google Cloud Platform.

Cloud Security Scanner - Automated vulnerability scanning service for applications deployed in App Engine.

Access Transparency - Near real-time audit logs providing visibility to Google Cloud Platform administrators.

VPC Service Controls - Service to manage security perimeters for sensitive data in Google Cloud Platform services.

IoT

Cloud IoT Core - Secure device connection and management service for Internet of Things.

Edge TPU - Purpose-built ASIC designed to run inference at the edge. As of September 2018, this product is in private beta.

Cloud IoT Edge - Brings AI to the edge computing layer.

API Platform

Maps Platform - APIs for maps, routes, and places based on Google Maps.

Apigee API Platform - Lifecycle management platform to design, secure, deploy, monitor, and scale APIs.

API Monetization - Tool for API providers to create revenue models, reports, payment gateways, and developer portal integrations.

Developer Portal - Self-service platform for developers to publish and manage APIs.

API Analytics - Service to analyse API-driven programs through monitoring, measuring, and managing APIs.

Apigee Sense - Enables API security by identifying and alerting administrators to suspicious API behaviours.

Cloud Endpoints - An NGINX-based proxy to deploy and manage APIs.

Service Infrastructure - A set of foundational services for building Google Cloud products.
Regions and zones

VPC Flow Logs **records a sample of network flows sent from and received by VM instances, including instances used as Google Kubernetes Engine nodes**. These logs can be used for network monitoring, forensics, real-time security analysis, and expense optimization.

. Google Cloud always reserves 4 IP addresses for every subnet you create. Reason for this is:

- First IP is a network address
- Second is reserved for the default gateway
- Second-to-last is reserved for future use
- Last address is the broadcast address