* Prepare for Exam:
  1. Register for the Exam: <https://cloud.google.com/certification/cloud-engineer>
  2. Review each module of GCP (2-3 h)
  3. Take some time on GCP Console (30 m)
  4. <https://cloud.google.com/certification/guides/cloud-engineer> (30 m)
  5. Linux Academy, Flash Cards, <https://kb.epam.com/display/EPMGCP/GCP+Certification> (2 h)
  6. Search YouTube about How to Pass ACE Exam (2 h)
  7. G Cloud commands cheat sheet
  8. <https://cloud.google.com/certification/sample-questions/cloud-engineer> (2 h)

GCloud commands structure:

gcloud + release level (optional) + component + entity + operation + positional args + flags

ex: gcloud + compute + instances + create + example-instance-1 + --zone=us-central1-a

Introduction to Google Cloud Platform

* IaaS Infrastructure as a Service: provides raw compute, storage, and network, organized in ways that are familiar from physical data centers
* PaaS Platform as a Service: lets you bind your application code to libraries that give access to the infrastructure your application needs.
* Cloud computing resources are available on-demand and self-service.
* As part of building a fault-tolerant application, you can spread your resources across multiple zones in a region.
* The cloud provider has a pool of resources and lets you use more or less on demand.
* The zones within a region have fast network connectivity among them.

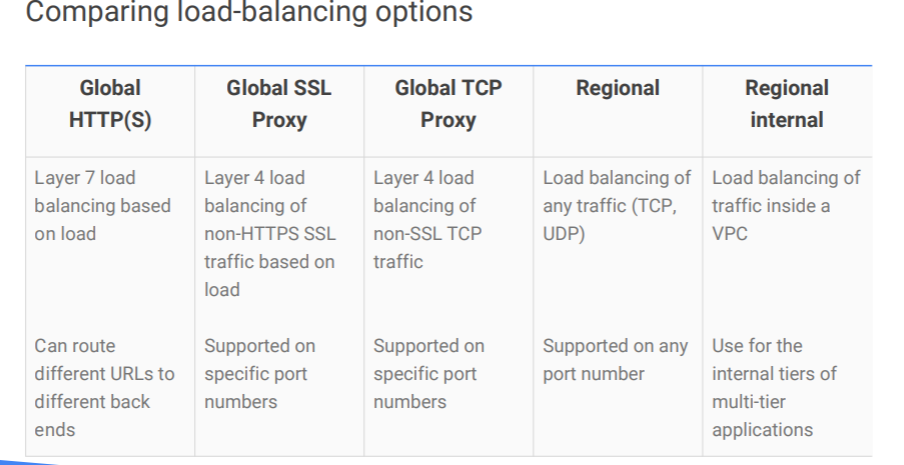
Getting Started with Google Cloud Platform

* You chose the correct response! Google Cloud Platform manages the lower layers of the security stack, such as physical security, and gives customers tools for managing the higher layers.

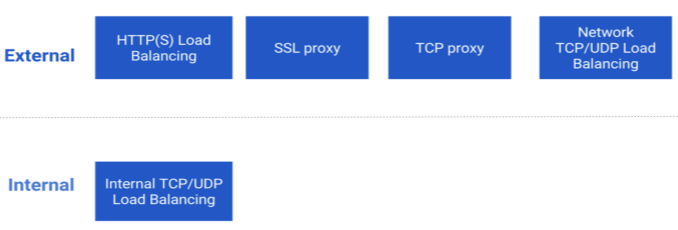
Virtual Machines and Networks in the Cloud

* Each VPC network is contained in a GCP project, but through VPC Peering, you can connect 2 different VPCs each other's, Or you can use the Shared VPC to take advantage of IAM.
* Google Cloud VPC networks are global; subnets are regional
* VPC subnets can span the zones that make up a region. This is beneficial because your solutions can incorporate fault tolerance without complicating your network topology.
* You can dynamically increase the size of a subnet in a custom network by expanding the range of IP addresses allocated to it. Doing that doesn’t affect already configured VMs.
* Compute Engine offers managed virtual machines
* You can get up to a 30% net discount for VMs that run the entire month.
* $$ Preemptible VM: you’ve given Compute Engine permission to terminate it if its resources are needed elsewhere, the per-hour price of preemptible VMs incorporates a substantial discount, so you can save a lot of money with preemptible VMs.
* Compute Engine has a feature called Autoscaling that lets you add and take away VMs from your application based on load metrics.
* Virtual Private Cloud VPC: use route table to forward traffic within the network even across subnets, also use its firewall to control what network traffic is allowed, also use Shared VPC to share a network or individual subnets with other GCP projects.
* Cloud Load Balancing is a fully distributed, software-defined, managed service for all your traffic, It provides cross-region load balancing, If you need cross-regional load balancing for a Web application, use HTTP(S) load balancing. For Secure Sockets Layer traffic that is not HTTP, use the Global SSL Proxy load balancer. If it’s other TCP traffic that does not use Secure Sockets Layer, use the Global TCP Proxy load balancer.
* What if you want to load balance traffic inside your project, say, between the presentation layer and the business layer of your application? For that, use the Internal load balancer. It accepts traffic on a GCP internal IP address and load balances it across Compute Engine VMs.
* Cloud CDN (Content Delivery Network): use Google's globally distributed edge caches to cache content close to your users Or use CDN Interconnect if you’d prefer to use a different CDN.
* Partner Interconnect provides connectivity between your on-premises network and your VPC network through a supported service provider.

.



TCP,'LJDP Load 
Regional 
Network TCP,'uop Load



Storage in the Cloud

* Cloud Storage is binary “immutable" large-object storage, it is not a file system,
* Object Storage means this: you say to your storage, “Here, keep this arbitrary sequence of bytes,,” and the storage lets you address it with a unique key. In Google Cloud Storage and in other systems, these unique keys are in the form of URLs, which means object storage interacts well with web technologies.
* Cloud Storage is object storage rather than file storage. Compute Engine virtual machines use Persistent Disk storage to contain their file systems.
* Data traveling between a customer’s device and Google is encrypted by default using HTTPS/TLS
* Cloud Storage Transfer Service enables you to import large amounts of online data into Google Cloud Storage quickly and cost-effectively.
* Your Cloud Storage files are organized into buckets with: (Globally unique name, Storage class, IAM polices and Access Control List ACL, Location region or multi-region, Object versioning and Object lifecycle)
* ACL Access Control List ACLs consists of: Scope, which defines who can perform the specified actions (for example, a specific user or group of users). And a permission, which defines what actions can be performed.
* Cloud Storage also offers lifecycle management policies. For example, you could tell Cloud Storage to delete objects older than 365 days, or to delete objects created before January 1, 2013; or to keep only the 3 most recent versions of each object in a bucket that has versioning enabled.
* Storage Classes: Regional, Multi-regional, Nearline and Coldline. Multi-regional and Regional are high-performance object storage, whereas Nearline and Coldline are backup and archival storage.
* There are several ways to bring data into Cloud Storage:
  + Online transfer: Self-managed copies using command-line tools or drag-and-drop
  + Storage Transfer Service: Scheduled, managed batch transfers
  + Transfer Appliance: Rackable appliances to securely ship your data
* Data stored in Coldline is billed at a low monthly storage rate, although a fee is assessed on retrievals.
* Cloud Bigtable: is managed NoSQL, accessed using HBase API and it has a native compatibility with big data, Hadoop ecosystems.
* A hashtable is a program structure that stores a flexible variety of data items based on a single lookup key. Persistent hashtables survive each individual runs of the program and so offer long-term data storage.
* NoSQL databases such as Cloud Bigtable are suitable when all items in the database needn't have their integrity checked by a database schema. Why not? Maybe you want your database items to contain variable fields, or maybe because you simply want your application to manage database integrity.
* Cloud Bigtable: is (Replicated storage, Data encryption in-flight and at rest, Role-based ACLs, Drives major applications such as Google Analytics and Gmail.
* Bigtable Access Patterns: Data can be read from and written to Cloud Bigtable through Application API , Streaming or Batch Processing
* Cloud SQL: is a managed RDBMS that Offers MySQL and PostgreSQL databases as a service, Automatic replication, Managed backups, Vertical scaling (read and write), Horizontal scaling (read) and Google security
* Cloud Datastore: is a horizontally scalable NoSQL DB
* Cloud Datastore features:
  + Atomic transactions: execute a set of operations where either all succeed, or none occur
  + High availability of reads and writes
  + Massive scalability with high performance
  + Flexible storage and querying of data
  + Balance of strong and eventual consistency
  + Encryption at rest: automatically encrypts all data before it is written to disk and automatically decrypts the data when read by an authorized user.
  + Fully managed with no planned downtime
* Consider using Cloud Datastore, (is best for semi-structured application data that is used in App Engine applications. ) if you need to store structured objects, or if you require support for transactions and SQL-like queries. This storage services provides terabytes of capacity with a maximum unit size of 1 MB per entity.
* Consider using Cloud Bigtable, (is best for analytical data with heavy read and write events, like AdTech, financial or IoT data) if you need to store a large amount of structured objects. Cloud Bigtable does not support SQL queries, nor does it support multi-row transactions. This storage service provides petabytes of capacity with a maximum unit size of 10 MB per cell and 100 MB per row.
* Consider using Cloud Storage, (is best for structured and unstructured binary or object data, like images, large media files and backups. ) if you need to store immutable blobs larger than 10 MB, such as large images or movies. This storage service provides petabytes of capacity with a maximum unit size of 5 TB per object.
* Consider using Cloud SQL or Cloud Spanner, if you need full SQL support for an online transaction processing system. Cloud SQL provides up to Up to 10,230 GB, depending on machine type, while Cloud Spanner provides petabytes. If Cloud SQL does not fit your requirements because you need horizontal scalability, not just through read replicas, consider using Cloud Spanner.
* Cloud SQL is best for web frameworks and existing applications, like storing user credentials and customer orders.
* Cloud Spanner is best for large-scale database applications that are larger than 2 TB. For example, for financial trading and e-commerce use cases.

Containers in the Cloud

* Why use containers?
  + Consistency Across development, testing, and production environments
  + Loose coupling Between application and operating system layers
  + Workload migration Simplified between on-premises and cloud environments
  + Agility Agile development and operations
* Kubernetes is a container cluster orchestration system, Automates deployment, scaling, and operations for container clusters, its open source built for Public, private, hybrid cloud
* Kubernetes eases application management
  + Workload portability: You can run in many environments, across cloud providers, Implementation is open and modular.
  + Rolling updates: You can upgrade applications without downtime.
  + Persistent storage: Details of how storage is provided are abstracted from how it is consumed.
  + Multi-zone clusters: Run a single cluster in multiple zones.
  + Load balancing: External IP address routes traffic to correct port.
  + Autoscaling: Automatically adapt to changes in workload
* Kubernetes Engine’s complementary services:
  + Google Cloud Container Builder: Create Docker container images from app code in Google Cloud Storage
  + Google Container Registry: Docker image storage that’s private to your GCP project
* Container Builder lets you create Docker container images from application source code located in Cloud Storage. Container images created by Container Builder are automatically stored in Container Registry.
* Container Registry provides private Docker image storage on Google Cloud Platform. Although Docker provides a central registry to store public images, you might not want your images to be accessible to the world. In this case, you must use a private registry.
* The registry can be accessed through an HTTPS endpoint
* Virtual Machines configured on the hardware level but Containers on the software level.

Containers start much faster than virtual machines and use fewer resources, because each container does not have its own instance of the operating system.

* POD: a group of containers that are deployed together with guaranteed network access.
* In Kubernetes, all containers run in pods.
* Kubernetes Engine > PODS > Containers
* A Kubernetes cluster is a group of machines where Kubernetes can schedule containers in pods. The machines in the cluster are called “nodes.”
* Because the resources used to build Kubernetes Engine clusters come from Compute Engine, Kubernetes Engine gets to take advantage of Compute Engine’s and Google VPC’s capabilities.
* The Cluster Nodes are nothing but a virtual machines.

Applications in the Cloud

* App Engine is a PaaS for building scalable applications: it makes deployment, maintenance, and scalability easy so you can focus on innovation
* You can choose which Infrastructure that should your application run on, even on VM through the Compute Engine or Container through Kubernetes Engine BUT what if you don’t want to focus on the infrastructure at all? Then use the App Engine, just put your code on and the App Engine Service should take care of the rest.
* App Engine will scale your application automatically in response to the amount of traffic it receives so you only pay for the resources you use.
* App Engine features:
  + Persistent storage with queries, sorting, and transactions
  + Automatic scaling and load balancing
  + Asynchronous task queues for performing work outside the scope of a request
  + Scheduled tasks for triggering events at specified times or regular intervals
  + Integration with other Google cloud services and APIs
* You’ll develop your application and run a test version of it locally using the App Engine SDK. Then, when you’re ready, you’ll use the SDK to deploy it.
* the App Engine flexible environment allows you to customize your runtime and even the operating system of your virtual machine using Dockerfiles.
* App Engine Standard Environment SE starts up faster than the Flixable Environment FE, but that you get less access to the infrastructure in which your application runs.
* SE has no SSH access but FE has
* SE not write to local disk but FE do
* SE not support 3rd party binaries but FE do and (also allows stack customization and background processes).
* SE access network via App Engine Services only but FE can
* SE, after free daily use, pay per instance class, with automatic shutdown but FE pay for resource allocation per hour; no automatic shutdown
* App Engine Standard Environment is for people who want the service to take maximum control of their application’s deployment and scaling. Kubernetes Engine gives the application owner the full flexibility of Kubernetes. App Engine Flexible Edition is in between.
* Cloud Endpoints is a distributed API management system. It provides an API console, hosting, logging, monitoring, and other features to help you create, share, maintain, and secure your APIs.
* Cloud Endpoints uses the distributed Extensible Service Proxy to provide low latency and high performance for serving even the most demanding APIs.
* Extensible Service Proxy is a service proxy based on NGINX. It runs in its own Docker container for better isolation and scalability.
* Apigee Edge is also a platform for developing and managing API proxies.
* Cloud Endpoints helps you create and maintain APIs; Apigee Edge helps you secure and monetize APIs.
* Apigee Edge used to gradually decompose a pre-existing monolithic application, not implemented in GCP, into microservices.

Next Activity Developing, Deploying, and Monitoring in the Cloud

* Cloud Source Repositories provides Git version control to support collaborative development of any application or service, including those that run on App Engine and Compute Engine.
* Google Cloud diagnostics tools like the Debugger and Error Reporting can use the code from your Git repositories to let you track down issues to specific errors in your deployed code without slowing down your users.
* Cloud Functions (written in JavaScript) is a lightweight, event-based, asynchronous compute solution that allows you to create small, single-purpose functions that respond to cloud events without the need to manage a server or a runtime environment. (ex: suppose that you want to automatically catch any uploaded file to be renamed then you can create this functionality using Cloud Function)
* Cloud Events are things that happen in your cloud environment. These might be things like changes to data in a database, files added to a storage system, or a new virtual machine instance being created. (you can create Event Trigger to trigger these events to handle something)
* Deployment Manager is an infrastructure management service that automates the creation and management of your Google Cloud Platform resources for you.
* To use Deployment Manager, you create a template file, using either the YAML markup language or Python, that describes what you want the components of your environment to look like. Then you give the template to Deployment Manager, which figures out and does the actions needed to create the environment your template describes.
* Even you can store and version-control your Deployment Manager templates in Cloud Source Repositories.
* **Stackdriver** is GCP’s tool for monitoring, logging, and diagnostics. Stackdriver gives you access to many different kinds of signals from your infrastructure platforms, virtual machines, containers, middleware, and application tier: logs, metrics, traces. It gives you insight into your application’s health, performance, and availability, so if issues occur you can fix them faster.
* With Stackdriver Trace, you can sample the latency of App Engine applications and report per-URL statistics.
* Stackdriver Debugger offers a different way. It connects your application’s production data to your source code, so you can inspect its state of your application at any code location in production.

Big Data and Machine Learning in the Cloud

* Cloud Dataproc is managed Hadoop, fast, easy, managed way to run Hadoop and

Spark/Hive/Pig on GCP, Also you can scale clusters up and down even when jobs are running, can be used when Migrate on-premises Hadoop jobs to the cloud

* Apache Hadoop is an open-source framework for big data. It is based on the MapReduce programming model, which Google invented and published.
* Cloud Dataproc is a fast, easy, managed way to run Hadoop, Spark, Hive, and Pig on GCP.
* Why use Cloud Dataproc?
  + Easily migrate on-premises Hadoop jobs to the cloud.
  + Quickly analyze data (like log data) stored in Cloud Storage; create a cluster in 90 seconds or less on average, and then delete it immediately.
  + Use Spark/Spark SQL to quickly perform data mining and analysis.
  + Use Spark Machine Learning Libraries (MLlib) to run classification algorithms.
* Cloud Dataflow offers managed data pipelines, Processes data using Compute Engine instances.
* Cloud Dataproc is great when you have a dataset of known size, or when you want to manage your cluster size yourself. But Cloud Dataflow is great when your data shows up in Realtime, Or it’s of unpredictable size or rate.
* Dataflow is a unified programming model and a managed service for developing and executing a wide range of data processing patterns including ETL, batch computation, and continuous computation. Cloud Dataflow frees you from operational tasks like resource management and performance optimization.
* Cloud Dataflow features: Automatic Resource Management, All resources are provided on demand, Intelligent Work Scheduling, Auto Scaling, Unified Programming Model, Open Source, Monitoring, Integrates with Cloud Storage, Cloud Pub/Sub, Cloud Datastore, Cloud Bigtable, and BigQuery for seamless data processing, Reliable & Consistent Processing Cloud Dataflow provides built-in support for fault-tolerant execution
* BigQuery is a fully managed data warehouse 10 ● Provides near real-time interactive analysis of massive datasets (hundreds of TBs) ● Query using SQL syntax (SQL 2011) ● No cluster maintenance is required.
* BigQuery’s features: Flexible Data Ingestion Load your data from Cloud Storage or Cloud Datastore, or stream it into BigQuery at 100,000 rows per second to enable real-time analysis of your data. Global Availability You have the option to store your BigQuery data in European locations while continuing to benefit from a fully managed service, now with the option of geographic data control, without low-level cluster maintenance. Security and Permissions You have full control over who has access to the data stored in BigQuery. If you share datasets, doing so will not impact your cost or performance; those you share with pay for their own queries. Cost Controls BigQuery provides cost control mechanisms that enable you to cap your daily costs at an amount that you choose. For more information, see Cost Controls. Highly Available Transparent data replication in multiple geographies means that your data is available and durable even in the case of extreme failure modes. Super Fast Performance Run super-fast SQL queries against multiple terabytes of data in seconds, using the processing power of Google's infrastructure. Fully Integrated In addition to SQL queries, you can easily read and write data in BigQuery via Cloud Dataflow, Spark, and Hadoop. Connect with Google Products You can automatically export your data from Google Analytics Premium into BigQuery and analyze datasets stored in Google Cloud Storage, Google Drive, and Google Sheets. BigQuery can make Create, Replace, Update, and Delete changes to databases, subject to some limitations and with certain known issues.
* It’s easy to get data into BigQuery. You can load from Cloud Storage or Cloud Datastore, or stream it into BigQuery at up to 100,000 rows per second.
* Cloud Pub/Sub is scalable, reliable messaging, a fully managed real-time messaging service that allows you to send and receive messages between independent applications.
* Cloud Pub/Sub features: Highly Scalable Any customer can send up to 10,000 messages per second, by default—and millions per second and beyond, upon request. Push and Pull Delivery Subscribers have flexible delivery options, whether they are accessible from the internet or behind a firewall. Encryption Encryption of all message data on the wire and at rest provides data security and protection. Replicated Storage Designed to provide “at least once” message delivery by storing every message on multiple servers in multiple zones. Message Queue Build a highly scalable queue of messages using a single topic and subscription to support a one-to-one communication pattern. End-to-End Acknowledgement Building reliable applications is easier with explicit application-level acknowledgements. Fan-out Publish messages to a topic once, and multiple subscribers receive copies to support one-to-many or many-to-many communication patterns. REST API Simple, stateless interface using JSON messages with API libraries in many programming languages.
* Why use Cloud **Pub/Sub** (Publisher\Subscribers)? ● Building block for data ingestion in Dataflow, Internet of Things (IoT), Marketing Analytics ● Foundation for Dataflow streaming ● Push notifications for cloud-based applications ● Connect applications across Google Cloud Platform (push/pull between Compute Engine and App Engine)
* Cloud Datalab offers interactive data exploration ● Interactive tool for large-scale data exploration, transformation, analysis, and visualization
* Cloud Datalab features: Integrated Cloud Datalab handles authentication and cloud computation out of the box and is integrated with BigQuery, Compute Engine, and Cloud Storage. Multi-Language Support Cloud Datalab currently supports Python, SQL, and JavaScript (for BigQuery user-defined functions). Notebook Format Cloud Datalab combines code, documentation, results, and visualizations together in an intuitive notebook format. Pay-per-use Pricing Only pay for the cloud resources you use: the App Engine application, BigQuery, and any additional resources you decide to use, such as Cloud Storage. Interactive Data Visualization Use Google Charts or matplotlib for easy visualizations. Collaborative Git-based source control of notebooks with the option to sync with non-Google source code repositories like GitHub and Bitbucket. Open Source Developers who want to extend Cloud Datalab can fork and/or submit pull requests on the GitHub hosted project. Custom Deployment Specify your minimum VM requirements, the network host, and more. IPython Support Cloud Datalab is based on Jupyter (formerly IPython) so you can use a large number of existing packages for statistics, machine learning, etc. Learn from published notebooks and swap tips with a vibrant IPython community.
* Cloud DataLab Analyze data in BigQuery, Compute Engine, and Cloud Storage using Python, SQL, and JavaScript.
* Cloud Machine Learning Platform provides modern machine learning services, with pre-trained models and a platform to generate your own tailored models.
* TensorFlow is an open-source software library that’s exceptionally well suited for machine learning applications like neural networks.
* Why use the Cloud Machine Learning platform?
  + For structured data: Classification and regression, Recommendation and Anomaly detection
  + For unstructured data: Image\video analytics and Text analytics
* Cloud Vision API enables developers to understand the content of an image by encapsulating powerful machine learning models in an easy to use REST API.
* The Cloud Speech API enables developers to convert audio to text.
* The Cloud Natural Language API offers a variety of natural language understanding technologies to developers. It can do syntax analysis, breaking down sentences supplied by your users into tokens, identify the nouns, verbs, adjectives, and other parts of speech, and figure out the relationships among the words.
* Cloud Translation API provides a simple programmatic interface for translating an arbitrary string into any supported language.
* The Google Cloud Video Intelligence API allows developers to use Google video analysis technology as part of their applications.

Virtual Networks: Virtual Private Cloud VPC

* PoPs are where Google’s network is connected to the rest of the internet.
* VPC objects: ● Projects ● Networks (Default, auto mode, custom mode) ● Subnetworks ● Regions ● Zones ● IP addresses (Internal, external, range) ● Virtual machines (VMs) ● Routes ● Firewall rules
* VPC Project:
  + Associates objects and services with billing.
  + Contains networks (up to 5) that can be shared/peered.
* VPC Network:
  + Has no IP address range.
  + Is global and spans all available regions.
  + Contains subnetworks.
  + Is available as default, auto, or custom.
  + Has 3 types:
    - Default Mode: Every project ● One subnet per region ● Default firewall rules
    - Auto Mode: Default network ● One subnet per region ● Regional IP allocation ● Fixed /20 subnetwork per region ● Expandable up to /16
    - Custom Mode: No default subnets created ● Full control of IP ranges ● Regional IP allocation ● Expandable to any RFC 1918 size
* VMs should communicate using the Internal IP if they are in the same network even if they are in a different regions, and using external IP if they are in different networks even if they are in the same region.
* Using a VPN Gateway, you can securely connect your on-premises network to your GCP network
* VMs can be on the same subnet but in different zones.
* Subnetworks work on a regional scale. Because a region contains several zones, subnetworks can cross zones.
* Subnetworks can extend across zones within the same region.
* The subnet is simply an IP address range, and you can use IP addresses within that range, the first and second addresses in the range, .0 and .1, are reserved for the network and the subnet’s gateway, respectively.
* Every subnet has four reserved IP addresses in its primary IP range
* So if we have a 2 or more VMs in different zones but in the same Subnet (same region) then a single firewall rules can be applied on both VMs, also they can communicate to each other using the Subnet IP address.
* Speaking of IP addresses of a subnet, Google Cloud VPCs let you increase the IP address space of any subnets without any workload shutdown or downtime.
* Expand subnets without re-creating instances ● Cannot overlap with other subnets ● Must be inside the RFC 1918 address spaces ● Can expand but not shrink ● Auto mode can be expanded from /20 to /16 ● Avoid large subnets
* VM Internal IP: Allocated from subnet range to VMs by DHCP, VM name + IP is registered with network-scoped DNS
* VM External IP: Assigned from pool(ephemeral) Reserved (static), VM doesn't know external IP; it is mapped to the internal IP
* When you create a VM in GCP, its symbolic name is registered with an internal DNS service that translates the name to the internal IP address. DNS is scoped to the network, so it can translate web URLs and VM names of hosts in the same network, but it can't translate host names from VMs in a different network.
* The external IP address is mapped to the VM's internal address transparently by VPC.
* Each instance has a hostname that can be resolved to an internal IP address: ● The hostname is the same as the instance name. ● FQDN is [hostname].[zone].c.[project-id].internal
* Name resolution is handled by internal DNS resolver: ● Provided as part of Compute Engine (169.254.169.254). ● Configured for use on instance via DHCP. ● Provides answer for internal and external addresses.
* Domain name servers can be hosted on GCP, using Cloud DNS.
* Cloud DNS is a scalable, reliable, and managed authoritative Domain Name System, or DNS, service running on the same infrastructure as Google. Cloud DNS translates requests for domain names like google.com into IP addresses., Low latency, UI, command line, or API
* Alias IP Ranges let you assign a range of internal IP addresses as an alias to a virtual machine's network interface. This is useful if you have multiple services running on a VM, and you want to assign a different IP address to each service.
* A route is a mapping of an IP range to a destination, and let instances in a network send traffic directly to each other
* By default, every network has routes that let instances in a network send traffic directly to each other, even across subnets. In addition, every network has a default route that directs packets to destinations that are outside the network.
* Routes match packets by destination IP address. However, no traffic will flow without also matching a firewall rule.
* route is created when a network or subnet is created.
* Firewall rules protect your VM instances from unapproved connections
* Although firewall rules are applied to the network as a whole, connections are allowed or denied at the instance level. You can think of the firewall as existing not only between your instances and other networks, but between individual instances within the same network.
* firewall rule is composed of the following parameters:
  + The direction of the rule. Inbound connections are matched against ingress rules only, and outbound connections are matched against egress rules only.
  + The source of the connection for ingress packets, or the destination of the connection for egress packets.
  + The protocol and port of the connection, where any rule can be restricted to apply to specific protocols only or specific combinations of protocols and ports only.
  + The action of the rule, which is to allow or deny packets that match the direction, protocol, port, and source or destination of the rule.
  + The priority of the rule, which governs the order in which rules are evaluated. The first matching rule is applied.
  + The rule assignment. By default, all rules are assigned to all instances, but you can assign certain rules to certain instances only.
* Egress firewall rules control outgoing connections originated inside your GCP network. Egress allow rules allow outbound connections that match specific protocol, ports, and IP addresses. Egress deny rules prevent instances from initiating connections that match non-permitted port, protocol, and IP range combinations
* ingress or traffic coming into GCP’s network is not charged, unless there is a resource such as a load balancer that is processing ingress traffic. Responses to requests count as egress and are charged.
* egress or traffic leaving a virtual machine. Egress traffic to the same zone is not charged, as long as that egress is through the internal IP address of an instance. Also, egress traffic to Google products, like YouTube, Maps, Drive, or traffic to a different GCP service within the same region is not charged for. However, there is a charge for egress between zones in the same region, egress within a zone if the traffic is through the external IP address of an instance, and egress between regions.
* Static and ephemeral IP addresses attached to forwarding rules (no charge)
* GCP pricing calculator to estimate the cost of a collection of resources, because each GCP service has its own pricing model. The pricing calculator is a web-based tool that you use to specify the expected consumption of certain services and resources, and it then provides you with an estimated cost.
* VPC networks are by default isolated private networking domains. Therefore, no internal IP address communication is allowed between networks unless you set up mechanisms such as VPC peering or a VPN connection.
* allocating VMs on a single subnet to separate zones, you get improved availability without additional security complexity. A regional managed instance group contains instances from multiple zones across the same region, which provides increased availability.
* we placed resources in different zones in a single region, which provides isolation from many types of infrastructure, hardware, and software failures. Putting resources in different regions as shown on this slide provides an even higher degree of failure independence. This allows you to design robust systems with resources spread across different failure domains. When using a global load balancer, like the HTTP load balancer, you can route traffic to the region that is closest to the user. This can result in better latency for users and lower network traffic costs for your project.
* **Cloud NAT** is Google’s managed network address translation service. It lets you provision your application instances without public IP addresses, while also allowing them to access the internet in a controlled and efficient manner. This means your private instances can access the internet for updates, patching, configuration management, and more.
* Cloud NAT does not implement inbound NAT, In other words, hosts outside your VPC network cannot directly access any of the private instances behind the Cloud NAT gateway. This helps you keep your VPC networks isolated and secure.
* you should enable **Private Google Access** to allow VM instances that only have internal IP addresses to reach the external IP addresses of Google APIs and services. For example, if your private VM instance needs to access a Cloud Storage bucket, you need to enable Private Google Access. OR to create a public IP to access google APIs through it.

When a VM is created the ephemeral external IP address is assigned from a pool. There is no way to predict which address will be assigned, so there is no way to write a rule that will match that VM's IP address before it is assigned. Tags allow a symbolic assignment that does not depend on order in the IP addresses. It makes for simpler, more general, and easier to maintain, firewall rules.

* You can enable Private Google Access on a subnet.
* You can connect to a VM instance using an Identity-Aware Proxy (IAP) tunnel.

Virtual Machines VMs

* VMs created by the Compute Engine, it has some features that can't exist in physical hardware like:
  + a micro VM shares a CPU with other virtual machines, so you can get a VM with less capacity at a lower cost.
  + the virtual CPU will run above its rated capacity for a brief period, using the available shared physical CPU.
* Cloud Functions is very optimum for Microservice architecture but it only support Paython, NodeJS and Go languages.
* the data that you store on local SSDs persists only until you stop or delete the instance.
* How to Connect to Virtual Machine:
  + On a Linux instance (SSH), the creator has SSH capability and can use the GCP Console to grant SSH capability to other users.(Requires firewall rule to allow tcp:22)
  + On a Windows instance (RDP), the creator can use the GCP Console to generate a username and password. After that, anyone who knows the username and password can connect to the instance using a Remote Desktop Protocol, or RDP, client.(Requires firewall rule to allow tcp:3389)

VM lifecycle 
Provisioning 
IP ad&esses 
Running 
SSH RDP 
C on p ute 
Stopping 
Terminated 
Avadability

* The VM instance remains in the running state through the reset process.
* There are different ways you can change a VM state from running. Some methods involve the GCP Console and the gcloud command, while others are performed from the OS, such as for reboot and shutdown.
* Compute Engine can live migrate your virtual machine to another host due to a maintenance event to prevent your applications from experiencing disruptions. A VM’s availability policy determines how the instance behaves in such an event.
* When a VM is terminated, you do not pay for memory and CPU resources. However, you are charged for any attached disks and reserved IP addresses. In the terminated state, you can perform any of the actions listed here, such as changing the machine type, but you cannot change the image of a stopped VM.
* GCP offers several machine types that can be grouped into 2 categories:
  + Predefined machine types: These have a fixed collection of resources, are managed by Compute Engine and are available in multiple different classes. Each class has a predefined ratio of GB of memory per vCPU. These are the:
    - Standard machine types (suitable for tasks that have a balance of CPU and memory needs)
    - High-memory machine types (ideal for tasks that require more memory relative to vCPUs)
    - High-CPU machine types (ideal for tasks that require more vCPUs relative to memory)
    - Memory-optimized machine types (ideal for tasks that require intensive use of memory, with higher memory to vCPU ratios than high-memory machine types. These machines types are perfectly suited for in-memory databases and in-memory analytics, such as SAP HANA and business warehousing workloads, genomics analysis, and SQL analysis services.)
    - Compute-optimized machine types (ideal for compute-intensive workloads. These machine types offer the highest performance per core on Compute Engine. Built on the latest-generation Intel Scalable Processors (the Cascade Lake)
    - Shared-core machine types (provide one vCPU that is allowed to run for a portion of the time on a single hardware hyper-thread on the host CPU running your instance. Shared-core instances can be more cost-effective for running small, non-resource-intensive applications than other machine types. )
  + Custom machine types: These let you specify the number of vCPUs and the amount of memory for your instance.
  + VM Zone: Each zone supports a combination of Ivy Bridge, Sandy Bridge, Haswell, Broadwell, and Skylake platforms. When you create an instance in the zone, your instance will use the default processor supported in that zone. For example, if you create an instance in the us-central1-a zone, your instance will use a Sandy Bridge processor
* Compute Engine uses a resource-based pricing model, where each vCPU and each GB of memory on Compute Engine is billed separately rather than as part of a single machine type
* A preemptible VM is an instance that you can create and run at a much lower price than normal instances. However, Compute Engine might terminate (or preempt) these instances if it requires access to those resources for other tasks. Preemptible instances are excess Compute Engine capacity so their availability varies with usage.
* When you create a new VM instance, recommendations for the new instance will appear 24 hours after the instance has been created.
* To take advantage of the full 30% discount, create your VM instances on the first day of the month, because discounts reset at the beginning of each month. The graph on this slide demonstrates how your effective discount increases with use. For example, if you use a virtual machine for 50% of the month, you get an effective discount of 10%. If you use it for 75% of the month, you get an effective discount of 20%. If you use it for 100% of the month, you get an effective discount of 30%.

Sustained use discounts (example) 
- upgr.& 
f« 4 K pus 
12 KPUs

* a preemptible VM is an instance that you can create and run at a much lower price than normal instances. (up to 80% discount)
* One major use case for preemptible VMs is running a batch processing job. If some of those instances terminate during processing, the job slows but does not completely stop. Therefore, preemptible instances complete your batch processing tasks without placing additional workload on your existing instances, and without requiring you to pay full price for additional normal instances.
* A sole-tenant node is a physical Compute Engine server that is dedicated to hosting VM instances only for your specific project. Use sole-tenant nodes to keep your instances physically separated from instances in other projects, or to group your instances together on the same host hardware, for example if you have a payment processing workload that needs to be isolated to meet compliance requirements.
* Shielded VMs offer verifiable integrity of your VM instances, so you can be confident that your instances haven't been compromised by boot- or kernel-level malware or rootkits. Shielded VM's verifiable integrity is achieved through the use of Secure Boot, virtual trusted platform module or vTPM-enabled Measured Boot, and integrity monitoring
* The Shielded Cloud initiative is meant to provide an even more secure foundation for all of GCP by providing verifiable integrity and offering features, like vTPM shielding or sealing, that help prevent data exfiltration. (In order to use these shielded VM features, you need to select a shielded image. )
* Boot Image: When creating a virtual machine, you can choose the boot disk image. This image includes the boot loader, the operating system, the file system structure, any pre-configured software, and any other customizations.
* Local SSD: Data on these disks will survive a reset but not a VM stop or terminate, because these disks can’t be reattached to a different VM.
* RAM Disk: Faster than local disk, slower than memory
* You can simply use **tmpfs** if you want to store data in memory. This will be the fastest type of performance available if you need small data structures. I recommend a high-memory virtual machine if you need to take advantage of such features, along with a persistent disk to back up the RAM disk data.
* Persistent disks can be rebooted and snapshotted, but local SSDs and RAM disks are ephemeral. I recommend choosing a persistent HDD disk when you don't need performance but just need capacity. If you have high performance needs, start looking at the SSD options. The persistent disks offer data redundancy because the data on each persistent disk is distributed across several physical disks. Local SSDs provide even higher performance, but without the data redundancy. Finally, RAM disks are very volatile but they provide the highest performance.
* there is a limit on how many Local SSDs you can attach to a VM, there is also a limit on how many persistent disks you can attach to a VM.
* if you plan on having a large amount of Disk IO throughput, it will also compete with any network egress or ingress throughput. So remember that, especially if you will be increasing the number of drives attached to a virtual machine.
* With cloud persistent disks, things are very different because all that management is handled for you on the backend. You can simply grow disks and resize the file system because disks are virtual networked devices. Redundancy and snapshot services are built in and disks are automatically encrypted. You can even use your own keys, and that will ensure that no party can get to the data except you.
* The metadata server is particularly useful in combination with startup and shutdown scripts, because you can use the metadata server to programmatically get unique information about an instance, without additional authorization.
* Because the default metadata keys are the same on every instance, you can reuse your script without having to update it for each instance.
* Move an instance to a new zone:
  + Automated process (moving within region):
    - gcloud compute instances move
    - Update references to VM; not automatic
  + Manual process (moving between regions):
    - Snapshot all persistent disks on the source VM.
    - Create new persistent disks in destination zone restored from snapshots.
    - Create new VM in the destination zone and attach new persistent disks.
    - Assign static IP to new VM.
    - Update references to VM.
    - Delete the snapshots, original disks, and original VM.
* Snapshots can also be used to migrate data between zones. I just discussed this when going over the manual process of moving an instance between two regions, but this can also be used to simply transfer data from one zone to another.
* if you want to improve disk performance, you could use a snapshot to transfer data from a standard HDD persistent disk to a SSD persistent disk.
* Snapshot doesn't back up VM metadata, tags, etc.
* You can create an incremental backup to Cloud Storage but note that it will Not visible in your buckets because its managed by the Snapshot Service.
* Snapshots are available only to persistent disks (HDD & SSD) and not to local SSDs.
* Snapshots are useful for periodic backup of the data on your persistent disks.
* Snapshots are incremental and automatically compressed, so you can create regular snapshots on a persistent disk faster and at a much lower cost than if you regularly created a full image of the disk. As we saw with the previous examples, snapshots can be restored to a new persistent disk, allowing for a move to a new zone.
* You can resize disks, but never shrink them!
* Sustained use discounts are automatic discounts that you get for running specific Compute Engine resources (vCPUs, memory, GPU devices) for a significant portion of the billing month. To take advantage of the full 30% discount, create your VM instances on the first day of the month, because discounts reset at the beginning of each month.
* Persistent Disks are not physical disks, they are a virtual-networked service. Each persistent disk remains encrypted either with system-defined keys or with customer-supplied keys.

Identity and Access Management IAM

* IAM: It is a way of identifying who can do what on which resource.
* The Organization resource is closely associated with a G Suite or Cloud Identity account.
* When a user with a G Suite or Cloud Identity account creates a GCP Project, an Organization resource is automatically provisioned for them.
* G Suite or Cloud Identity super administrator:
  + Assign the Organization admin role to some users
  + Be the point of contact in case of recovery issues
  + Control the lifecycle of the G Suite or Cloud Identity account and Organization resource
* Organization admin:
  + Define IAM policies
  + Determine the structure of the resource hierarchy
  + Delegate responsibility over critical components such as Networking, Billing, and Resource Hierarchy through IAM roles
* Folders provide an additional grouping mechanism and isolation boundary between projects. Folders can be used to model different legal entities, departments, and teams within a company.
* There are three 3 types of roles in Cloud IAM: primitive roles, predefined roles, and custom roles.
* IAM Primitive roles are the original roles that were available in the GCP Console, but they are broad. You apply them to a GCP project, and they affect all resources in that project

The primitive roles are the Owner, Editor, and Viewer roles.

* The owner has full administrative access. This includes the ability to add and remove members and delete projects.
* The editor role has modify and delete access. This allows a developer to deploy applications and modify or configure its resources.
* The viewer role has read-only access.

Each project can have multiple owners, editors, viewers, and billing administrators.

* IAM predefined roles apply to a particular GCP service in a project, roles are meant to represent abstract functions and are customized to align with real jobs,

GCP services offers their own sets of predefined roles, and they define where those roles can be applied. This provides members with granular access to specific GCP resources and prevents unwanted access to other resources. These roles are collections of permissions because, to do any meaningful operations, you usually need more than one permission.

* Compute Engine has several predefined IAM roles. Let’s look at three of those:
  + The Compute Admin role provides full control of all Compute Engine resources. This includes all permissions that start with compute, which means that every action for any type of Compute Engine resource is permitted.
  + The Network Admin role contains permissions to create, modify, and delete networking resources, except for firewall rules and SSL certificates. In other words, the network admin role allows read-only access to firewall rules, SSL certificates, and instances to view their ephemeral IP addresses.
  + The Storage Admin role contains permissions to create, modify, and delete disks, images, and snapshots.
* IAM custom roles let you define a precise set of permissions, A lot of companies use the “least-privilege” model, in which each person in your organization is given the minimal amount of privilege needed to do their job.
* There are five different types of members: Google Accounts, Service Accounts, Google groups, G Suite domains, and Cloud Identity domains.
* A Google account represents a developer, an administrator, or any other person who interacts with GCP. Any email address that is associated with a Google account can be an identity, including gmail.com or other domains.
* A service account is an account that belongs to your application instead of to an individual end user. When you run code that is hosted on GCP, you specify the account that the code should run as. You can create as many service accounts as needed to represent the different logical components of your application.
* A G Suite domain represents a virtual group of all the Google accounts that have been created in an organization's G Suite account. G Suite domains represent your organization's internet domain name, such as example.com, and when you add a user to your G Suite domain, a new Google account is created for the user inside this virtual group,
* GCP customers who are not G Suite customers can get these same capabilities through Cloud Identity. Cloud Identity lets you manage users and groups using the Google Admin Console, but you do not pay for or receive G Suite’s collaboration products such as Gmail, Docs, Drive, and Calendar. Cloud Identity is available in free and premium editions.
* It’s important to note that you cannot use Cloud IAM to create or manage your users or groups. Instead, you can use Cloud Identity or G Suite to create and manage users.
* Using Google Cloud Directory Sync, your administrators can log in and manage GCP resources using the same usernames and passwords they already use. This tool synchronizes users and groups from your existing Active Directory or LDAP system with the users and groups in your Cloud Identity domain.
* The synchronization is one-way only; which means that no information in your Active Directory or LDAP map is modified. Google Cloud Directory Sync is designed to run scheduled synchronizations without supervision, after its synchronization rules are set up.
* The synchronization is one-way only; which means that no information in your Active Directory or LDAP map is modified. Google Cloud Directory Sync is designed to run scheduled synchronizations without supervision, after its synchronization rules are set up.
* Service accounts provide an identity for carrying out server-to-server interactions
  + Programs running within Compute Engine instances can automatically acquire access tokens with credentials.
  + Tokens are used to access any service API in your project and any other services that granted access to that service account.
  + Service accounts are convenient when you’re not accessing user data.
* There are three types of service accounts: user-created or custom, built-in, and Google APIs service accounts.
* Apart from the default service account, all projects come with a Google Cloud Platform APIs service account, identifiable by the email: project-number@cloudservices.gserviceaccount.com. This is a service account designed specifically to run internal Google processes on your behalf, and it is automatically granted the Editor role on the project.
* Alternatively, you can also start an instance with a custom service account. Custom service accounts provide more flexibility than the default service account, but they require more management from you. You can create as many custom service accounts as you need, assign any arbitrary access scopes or Cloud IAM roles to them, and assign the service accounts to any virtual machine instance.
* the default Compute Engine service account. As I mentioned, this account is automatically created per project. This account is identifiable by the email project-number-compute@developer.gserviceaccount.com, and it is automatically granted the Editor role on the project
* Scopes can be customized when you create an instance using the default service account, but these scopes can be changed after an instance is created by stopping it.
* Another distinction between service accounts is that default service accounts support both primitive and predefined IAM roles, but user-created service accounts only use predefined IAM roles.
* Roles for service accounts can also be assigned to groups or users. So, you can create a service account that has the InstanceAdmin role, which has permissions to create, modify, and delete virtual machine instances and disks. Then you treat this service account as the resource, and decide who can use it by providing users or a group with the Service Account User role. This allows those users to act as that service account to create, modify, and delete virtual machine instances and disks.
* Cloud IAM lets you slice a project into different microservices, each with access to different resources, by creating service accounts to represent each one. You assign the service accounts to the VMs when they are created, and you don’t have to ensure that credentials are being managed correctly because GCP manages security for you.
* Although users require a username and password to authenticate, service accounts use keys. There are two types of service account keys: GCP-managed keys and user-managed keys.
* GCP-managed keys are used by GCP services such as App Engine and Compute Engine. These keys cannot be downloaded and are automatically rotated and used for a maximum of two weeks.
* User-managed keys are created, downloadable, and managed by users. When you create a new key pair, you download the private key, which is not retained by Google. With user-managed keys, you are responsible for security of the private key and other management operations
* Use projects to group resources that share the same trust boundary.
* Cloud IAP lets you establish a central authorization layer for applications accessed by HTTPS, so you can use an application-level access control model instead of relying on network-level firewalls.

Storage and Database Services

* BigQuery sits on the edge between data storage and data processing. You can store data in BigQuery, but the intended use for BigQuery is big data analysis and interactive querying.

Machine generated alternative text:



Machine generated alternative text:



Machine generated alternative text:

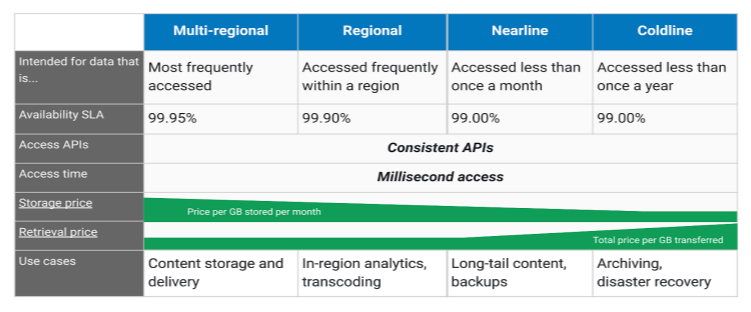


* How to decide:
  + First, ask yourself: Is your data structured? If it’s not, choose Cloud Storage.
  + If your data is structured, does your workload focus on analytics? If it does, you will want to choose Cloud Bigtable or BigQuery, depending on your latency and update needs.
  + Otherwise, check whether your data is relational. If it’s not relational, choose Cloud Firestore.
  + If it is relational, you will want to choose Cloud SQL or Cloud Spanner, depending on your need for horizontal scalability.

Machine generated alternative text:



* Cloud Storage is GCP’s object storage service, and it allows world-wide storage and retrieval of any amount of data at any time. You can use Cloud Storage for a range of scenarios including serving website content, storing data for archival and disaster recovery, or distributing large data objects to users via direct download.
* Cloud Storage is a collection of buckets that you place objects into.
* Cloud Storage has four storage classes: Regional, Multi-regional, Nearline, and Coldline.
  + Regional Storage enables you to store data at lower cost, its recommended when storing frequently accessed data in the same region as your Compute Engine instances. This provides you with better performance for data-intensive computations. You should also choose regional storage for data governance reasons - like if your data needs to remain in a specific region.
  + Multi-Regional Storage, on the other hand, is geo-redundant, which means Cloud Storage stores your data redundantly in at least two geographic locations separated by at least 100 miles within the multi-regional location of the bucket. Multi-Regional Storage can be placed only in multi-regional locations, such as the United States, the European Union, or Asia. Multi-Regional Storage is appropriate for storing data that is frequently accessed, such as serving website content, interactive workloads, or data supporting mobile and gaming applications.
  + Nearline storage is a low-cost, highly durable storage service for storing infrequently accessed data. This storage class is a good choice when you plan to read or modify your data less than once a month because the storage cost is low, but there is an associated retrieval cost. I also recommend using Nearline Storage for backups and serving long-tail multimedia content.
  + Coldline Storage is a very-low-cost, highly durable storage service for data archival, online backup, and disaster recovery. However, unlike other "cold" storage services, your data is available within milliseconds, not hours or days. Coldline Storage is the best choice for data that you plan to access at most once a year, due to its lower storage cost but higher retrieval cost. You might use Coldline Storage if you want to archive data or have access in the event of a disaster recovery event.



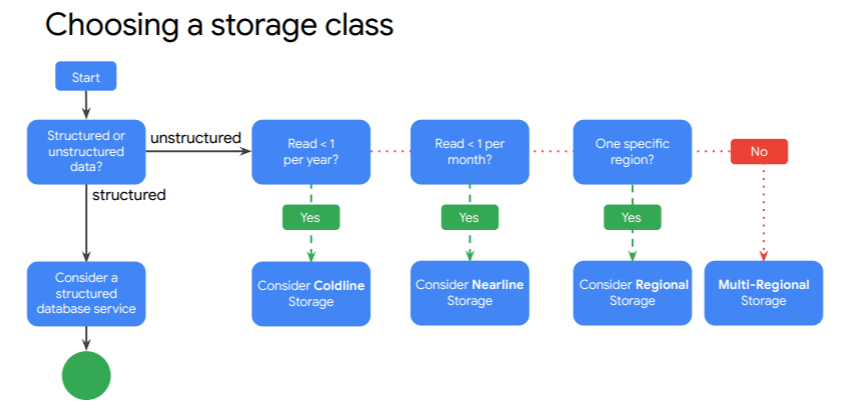
Machine generated alternative text:



* When you upload an object to the bucket, if you don't specify a storage class for the object, the object is assigned the bucket's storage class. You can change the default storage class of a bucket but you can't change a regional bucket to a multi-regional, and vice versa. but both multi-regional buckets and regional buckets can be changed to coldline or nearline.
* When you upload an object, you can specify a storage class for it. You can also change the storage class of an object that already exists in your bucket without moving the object to a different bucket or changing the URL to the object. Setting a per-object storage class is useful, for example, if you have objects in your bucket that you want to keep, but that you don't expect to access frequently. In this case, you can minimize costs by changing the storage class of those specific objects to Nearline Storage or Coldline Storage.

Cloud IAM 
Object 
ACLs 
Signed Policy 
Doc u ment 
Signed URL 
o 
Can be used together

* Let’s look at access control for your objects and buckets that are part of a project.
  + We can use IAM for the project to control which individual user or service account can see the bucket, list the objects in the bucket, view the names of the objects in the bucket, or create new buckets. For most purposes, Cloud IAM is sufficient, and roles are inherited from project to bucket to object.
  + Access control lists or ACLs offer finer control.
  + For even more detailed control, signed URLs provide a cryptographic key that gives time-limited access to a bucket or object.
  + Finally, a signed policy document further refines the control by determining what kind of file can be uploaded by someone with a signed URL. Let’s take a closer look at ACLs and signed URLs.
* Access Control Lists ACL is a mechanism you can use to define who has access to your buckets and objects, as well as what level of access they have. The maximum number of ACL entries you can create for a bucket or object is 100.
* The allUsers identifier listed on this slide represents anyone who is on the internet, with or without a Google account. The allAuthenticatedUsers identifier, in contrast, represents anyone who is authenticated with a Google account.
* Signed URLs allow you to do this for Cloud Storage. You create a URL that grants read or write access to a specific Cloud Storage resource and specifies when the access expires. That URL is signed using a private key associated with a service account.
* Cloud Storage features:
  + Customer-supplied encryption key (CSEK): Use your own key instead of Google-managed keys
  + Object Lifecycle Management: Automatically delete or archive objects
  + Object Versioning: Maintain multiple versions of objects
  + Directory synchronization: Synchronizes a VM directory with a bucket
  + Object change notification
  + Data import
  + Strong consistency
* Objects are immutable: Object Versioning supports the retrieval of objects that are deleted or overwritten
* Object inspection occurs in asynchronous batches, so rules may not be applied immediately. Also, updates to your lifecycle configuration may take up to 24 hours to go into effect. This means that when you change your lifecycle configuration, Object Lifecycle Management may still perform actions based on the old configuration for up to 24 hours. So keep that in mind.
* Object Change Notification (File Listner) can be used to notify an application when an object is updated or added to a bucket through a watch request. Completing a watch request creates a new notification channel. The notification channel is the means by which a notification message is sent to an application watching a bucket. As of this recording, the only type of notification channel supported is a webhook.
* Cloud Pub/Sub notifications are the recommended way to track changes to objects in your Cloud Storage buckets because they're faster, more flexible, easier to set up, and more cost-effective. Cloud Pub/Sub is Google’s distributed real-time messaging service.



* Cloud SQL is a fully managed service of either MySQL or PostgreSQL databases.
* Cloud SQL supports many clients, such as Cloud Shell, App Engine and G Suite scripts. It also supports other applications and tools that you might be used to like SQL Workbench, Toad and other external applications using standard MySQL drivers
* Services provided by Cloud SQL:
  + There is replica service that can replicate data between multiple zones as shown on the right. This is useful for automatic failover if an outage occurs.
  + Cloud SQL also provides automated and on-demand backups with point-in-time recovery.
  + You can import and export databases using mysqldump, or import and export CSV files.
  + Cloud SQL can also scale up, which does require a machine restart or scale out using read replicas.
* If you are connecting an application that is hosted within the same GCP project as your Cloud SQL instance, and it is collocated in the same region, choosing the Private IP connection will provide you with the most performant and secure connection using private connectivity. In other words, traffic is never exposed to the public internet.
* If the application is hosted in another region or project, or if you are trying to connect to your Cloud SQL instance from outside of GCP, you have 3 options. In this case, I recommend using Cloud Proxy, which handles authentication, encryption, and key rotation for you
* If you need more than 30 TB of storage space or over ,4000 concurrent connections to your database, or if you want your application design to be responsible for scaling, availability, and location management when scaling up globally, then consider using Cloud Spanner,
* Cloud Spanner is a service built for the cloud specifically to combine the benefits of relational database structure with non-relational horizontal scale
* Like a relational database, Cloud Spanner has schema, SQL, and strong consistency. Also, like a non-relational database, Cloud Spanner offers high availability, horizontal scalability, and configurable replication.
* Cloud Spanner is highly recommended when building consistent systems for transactions and inventory management in the financial services and retail industries
* A Cloud Spanner instance replicates data in N 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.
* The Cloud Spanner replication of data will be synchronized across zones using Google’s global fiber network
* Cloud Spanner is highly recommended when you have outgrown any relational database, are sharding your databases for throughput high performance, need transactional consistency, global data and strong consistency, or just want to consolidate your database.
* Cloud Firestore is a fast, fully managed, serverless, cloud-native NoSQL document database that simplifies storing, syncing, and querying data for your mobile, web, and IoT apps at global scale. Its client libraries provide live synchronization and offline support, and its security features and integrations with Firebase and GCP accelerate building truly serverless apps.
* Cloud Firestore also supports ACID transactions, so if any of the operations in the transaction fail and cannot be retried, the whole transaction will fail.
* Cloud Firestore: with automatic multi-region replication and strong consistency, your data is safe and available
* Cloud Firestore is actually the next generation of Cloud Datastore. Cloud Firestore can operate in Datastore mode, making it backwards- compatible with Cloud Datastore. By creating a Cloud Firestore database in Datastore mode, you can access Cloud Firestore's improved storage layer while keeping Cloud Datastore system behavior.
* Could Firestore has 2 modes: Datastore mode for new server projects, and Native mode for new mobile and web apps.
* Cloud Firestore is compatible with all Cloud Datastore APIs and client libraries.
* If your schema might change and you need an adaptable database, you need to scale to zero, or you want low maintenance overhead scaling up to terabytes, consider using Cloud Firestore.
* Cloud Bigtable is a fully managed NoSQL database with petabyte-scale and very low latency. It seamlessly scales for throughput and it learns to adjust to specific access patterns. Cloud Bigtable is actually the same database that powers many of Google’s core services, including Search, Analytics, Maps, and Gmail.
* Cloud Bigtable is a great choice for both operational and analytical applications, including IoT, user analytics, and financial data analysis, because it supports high read and write throughput at low latency. It’s also a great storage engine for machine learning applications.
* Cloud Bigtable integrates easily with popular big data tools like Hadoop, Cloud Dataflow, and Cloud Dataproc. Plus, Cloud Bigtable supports the open source industry standard HBase API
* In Cloud Bigtable, Processing is separated from storage
* if you need to store more than 1 TB of structured data, have very high volume of writes, need read/write latency of less than 10 milliseconds along with strong consistency, or need a storag
* the smallest Cloud Bigtable cluster you can create has three nodes and can handle 30,000 operations per second. Remember that you pay for those nodes while they are operational, whether your application is using them or not.
* Cloud Memorystore for Redis provides a fully managed in-memory data store service built on scalable, secure, and highly available infrastructure managed by Google. Applications running on GCP can achieve extreme performance by leveraging the highly scalable, available, secure Redis service without the burden of managing complex Redis deployments.

BigQuery is a data warehousing service that allows the storage of huge data sets while making them immediately processable without having to extract or run the processing in a separate service.

Resource Management

* The resource manager lets you hierarchically manage resources by project, folder, and organization.
* Although IAM policies are inherited top-to-bottom, billing is accumulated from the bottom up, Resource consumption is measured in quantities, like rate of use or time, number of items, or feature use. Because a resource belongs to only one project, a project accumulates the consumption of all its resources.
* Each project is associated with one billing account, which means that an organization contains all billing accounts.
* Because a project accumulates the consumption of all its resources, it can be used to

track resources and quota usage. Specifically, projects let you enable billing, manage

permissions and credentials, and enable service and APIs.

* All resources in GCP are subject to project quotas or limits. These typically fall into one of the three categories shown here:
  + How many resources you can create per project. For example, you can only have 5 VPC networks per project.
  + How quickly you can make API requests in a project or rate limits. For example, by default, you can only make 5 administrative actions per second per project when using the Cloud Spanner API.
  + There also regional quotas. For example, by default, you can only have 24 CPUs per region.
* Quotas are the maximum amount of resources you can create for that resource type as long as those resources are available.
* Why use project quotas:
  + Project quotas prevent runaway consumption in case of an error or malicious attack. For example, imagine you accidentally create 100 instead of 10 Compute Engine instances using the gcloud command line.
  + Quotas also prevent billing spikes or surprises. Quotas are related to billing, but we will go through how to set up budgets and alerts later, which will really help you manage billing.
  + Finally, quotas force sizing consideration and periodic review. For example, do you really need a 96-core instance, or can you go with a smaller and cheaper alternative?
* Labels are a utility for organizing GCP resources. Labels are key-value pairs that you can attach to your resources, like VMs, disks, snapshots and images. You can create and manage labels using the GCP console, gcloud, or the Resource Manager API, and each resource can have up to 64 labels.

For example, you could create a label to define the environment of your virtual machines. Then you define the label for each of your instances as either production or test.

* Let’s go over some examples of what to use labels for:
  + I recommend adding labels based on team or cost center to distinguish instances owned by different teams. You can use this type of label for cost accounting or budgeting. For example, team:marketing and team:research.
  + You can also use labels to distinguish components. For example, component:redis, component:frontend.
  + Again, you can label based on environment or stage.
  + You should also consider using labels to define an owner or a primary contact for a resource. For example, owner:gaurav, contact:opm.
  + Or add labels to your resources to define their state. For example, state:inuse, state:readyfordeletion
* Difference between Labels and Tags: Labels, we just learned, are user-defined strings in key-value format that are used to organize resources, and they can propagate through billing. Tags, on the other hand, are user-defined strings that are applied to instances only and are mainly used for networking, such as applying firewall rules.
* Budget: Setting a budget lets you track how your spend is growing toward that amount.
* I recommend labeling all your resources and exporting your billing data to BigQuery to analyze your spend. BigQuery is Google’s scalable, fully managed Enterprise Data Warehouse with SQL and fast response times.
* You can even visualize spend over time with Data Studio. Data Studio turns your data into informative dashboards and reports that are easy to read, easy to share, and fully customizable.

Resource Monitoring

* Stackdriver, a service that provides monitoring, logging, and diagnostics for your applications.
* Stackdriver dynamically discovers cloud resources and application services based on deep integration with Google Cloud Platform and Amazon Web Services. Because of its smart defaults, you can have core visibility into your cloud platform in minutes.
* Stackdriver has services for monitoring, logging, error reporting, fault tracing, and debugging. You only pay for what you use,
* Site Reliability Engineering SRE is a discipline that applies aspects of software engineering to operations whose goals are to create ultra-scalable and highly reliable software systems. This discipline has enabled Google to build, deploy, monitor, and maintain some of the largest software systems in the world.
* Stackdriver dynamically configures monitoring after resources are deployed and has intelligent defaults that allow you to easily create charts for basic monitoring activities.
* A Workspace is the root entity that holds monitoring and configuration information in Stackdriver Monitoring. Each Workspace can have between 1 and 100 monitored projects, including one or more GCP projects and any number of AWS accounts. You can have as many Workspaces as you want, but GCP projects and AWS accounts can't be monitored by more than one Workspace.
* A Workspace contains the custom dashboards, alerting policies, uptime checks, notification channels, and group definitions that you use with your monitored projects.
* The first monitored GCP project in a Workspace is called the hosting project, and it must be specified when you create the Workspace. The name of that project becomes the name of your Workspace.
* To access an AWS account, you must configure a project in GCP to hold the AWS Connector
* This means that a Stackdriver role assigned to one person on one project applies equally to all projects monitored by that Workspace, So In order to give people different roles per-project and to control visibility to data, consider placing the monitoring of those projects in separate Workspaces.
* Best practices when creating alerts:
  + I recommend alerting on symptoms, and not necessarily causes. For example, you want to monitor failing queries of a database and then identify whether the database is down.
  + Next, make sure that you are using multiple notification channels, like email and SMS. This helps avoid a single point of failure in your alerting strategy.
  + I also recommend customizing your alerts to the audience’s need by describing what actions need to be taken or what resources need to be examined.
  + Finally, avoid noise, because this will cause alerts to be dismissed over time. Specifically, adjust monitoring alerts so that they are actionable and don’t just set up alerts on everything possible.
* Uptime Check: The type of uptime check can be set to HTTP, HTTPS, or TCP. The resource to be checked can be an App Engine application, a Compute Engine instance, a URL of a host, or an AWS instance or load balancer, For each uptime check, you can create an alerting policy and view the latency of each global location.
* Stackdriver Monitoring can access some metrics without the Monitoring agent, including CPU utilization, some disk traffic metrics, network traffic, and uptime information. However, to access additional system resources and application services, you should install the Monitoring agent. The Monitoring agent is supported for Compute Engine and EC2 instances.
* The Monitoring agent can be installed with these two simple commands, which you could include in your startup script.
* If the standard metrics provided by Stackdriver monitoring do not fit your needs, you can create custom metrics.
* Stackdriver Logging allows you to store, search, analyze, monitor, and alert on log data and events from GCP and AWS. It is a fully managed service that performs at scale and can ingest application and system log data from thousands of VMs.
* Logging includes storage for logs, a user interface called the Logs Viewer, and an API to manage logs programmatically. The service lets you read and write log entries, search and filter your logs, and create log-based metrics.
* Logs are only retained for 30 days, but you can export your logs to Cloud Storage buckets, BigQuery datasets, and Cloud Pub/Sub topics.
* Exporting logs to BigQuery allows you to analyze logs and even visualize them in Data Studio, BigQuery runs extremely fast SQL queries on gigabytes to petabytes of data. This allows you to analyze logs,
* For example, I queried my logs to identify the top IP addresses that have exchanged traffic with my web server. Depending on where these IP addresses are and who they belong to, I could relocate part of my infrastructure to save on networking costs or deny some of these IP addresses if I don’t want them to access my web server.
* If you want to visualize your logs, I recommend connecting your BigQuery tables to Data Studio. Data Studio transforms your raw data into the metrics and dimensions that you can use to create easy-to-understand reports and dashboards.
* I mentioned that you can also export logs to Cloud Pub/Sub. This enables you to stream logs to applications or endpoints.
* Similar to Stackdriver’s Monitoring agent, it’s a best practice to install the Logging agent on all your VM instances. The Logging agent can be installed with these two simple commands, which you could include in your startup script. This agent is supported for Compute Engine and EC2 instances
* Stackdriver Error Reporting counts, analyzes, and aggregates the errors in your running cloud services. A centralized error management interface displays the results with sorting and filtering capabilities, and you can even set up real-time notifications when new errors are detected.
* Stackdriver Trace is a distributed tracing system that collects latency data from your applications and displays it in the GCP Console.
* Stackdriver Trace automatically analyzes all of your application's traces to generate in-depth latency reports that surface performance degradations and can capture traces from App Engine, HTTP(S) load balancers, and applications instrumented with the Stackdriver Trace API.
* Managing the amount of time it takes for your application to handle incoming requests and perform operations is an important part of managing overall application performance. Stackdriver Trace is actually based on the tools used at Google to keep our services running at extreme scale.
* Stackdriver Debugger is a feature of GCP that lets you inspect the state of a running application, in real time, without stopping or slowing it. Specifically, the debugger adds less than 10ms to the request latency when the application state is captured. In most cases, this is not noticeable by users.
* Monitoring is the foundational process at the base of Google's Site Reliability Engineering (SRE)
* Services that are currently supported by Cloud Error Reporting (App Engine Standard, App Engine Flexible, Compute Engine and Kubernates)

Interconnecting Networks

* Direct Peering requires you to meet Google’s peering requirements.
* Shared VPC is a centralized approach to multi-project networking, because security and network policy occurs in a single designated VPC network.
* VPNs use IPSec tunnels to provide an encapsulated and encrypted path through a hostile or untrusted environment.
* Dedicated Interconnect requires a connection in a GCP colocation facility and provides 10 Gbps per link.
* Cloud VPN securely connects your on-premises network to your GCP VPC network through an IPsec VPN tunnel. Traffic traveling between the two networks is encrypted by one VPN gateway, then decrypted by the other VPN gateway.
* In order to create a connection between two VPN gateways, you must establish two VPN tunnels. Each tunnel defines the connection from the perspective of its gateway, and traffic can only pass when the pair of tunnels is established.
* Cloud VPN supports both static and dynamic routes. In order to use dynamic routes, you need to configure Cloud Routers. Cloud Router can manage routes for a Cloud VPN tunnel using Border Gateway Protocol, or BGP. This routing method allows for routes to be updated and exchanged without changing the tunnel configuration.
* To automatically propagate network configuration changes, the VPN tunnel uses Cloud Router to establish a BGP session between the VPC and the on-premises VPN gateway, which must support BGP.
* Dedicated connections provide a direct connection to Google’s network, but shared connections provide a connection to Google’s network through a partner. Layer 2 connections (Dedicated and Partner Interconnect) use a VLAN that pipes directly into your GCP environment, providing connectivity to internal IP addresses in the RFC 1918 address space. Layer 3 connections (Direct and Carrier Peering) provide access to G Suite services, YouTube, and Google Cloud APIs using public IP addresses.
* Cloud VPN. This service uses the public internet, but traffic is encrypted and provides access to internal IP addresses. That’s why Cloud VPN is a useful addition to Direct Peering and Carrier Peering.
* Dedicated Interconnect provides direct physical connections between your on-premises network and Google’s network. This enables you to transfer large amounts of data between networks, which can be more cost-effective than purchasing additional bandwidth over the public internet.
* Partner Interconnect provides connectivity between your on-premises network and your VPC network through a supported service provider. This is useful if your data center is in a physical location that cannot reach a Dedicated Interconnect colocation facility or if your data needs don't warrant a Dedicated Interconnect.

Ipsec VPN 
Dedicated 
Interconnect 
Encrypted tunnel to VPC 
networks through the 
internet 
Dedicated. direct 
connection to vpC 
bandwidth. 
connection to VPC 
network through a service 
tunnel 
10 Gbps per ink 
100 Gbps 
50 Mbps - 
10 G-*2S per 
connect ion 
Req s 
(h premises 
VPN gateway 
in 
colocation facility 
Access 
Internal IP 
addresses

* Cloud Peering services, which are Direct Peering and Carrier Peering. These services are useful when you require access to Google and Google Cloud properties.
* Direct Peering with Google is done by exchanging BGP routes between Google and the peering entity. After a Direct Peering connection is in place, you can use it to reach all of Google’s services, including the full suite of Google Cloud Platform products. Unlike Dedicated Interconnect, Direct Peering does not have an SLA.
* GCP’s Edge Points of Presence, or PoPs, are where Google's network connects to the rest of the internet via peering.
* If you require access to Google public infrastructure and cannot satisfy Google’s peering requirements, you can connect via a Carrier Peering partner.

Conn t ion 
Direct 
Carrier 
peering 
provides 
Dedicated. direct 
connection to Google's 
Peering through servZe 
provider to Google's 
put*c network 
10 Gbps 
per link 
Varies based on 
Requirements 
Connection in 
c,cp pops 
Service provider 
Access Type 
Public IP

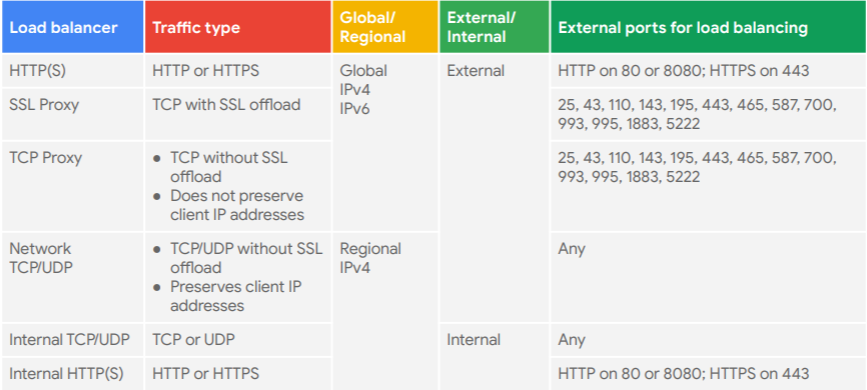
* Interconnect services provide direct access to RFC1918 IP addresses in your VPC, with an SLA. Peering services, in contrast, offer access to Google public IP addresses only, without an SLA.
* Shared VPC allows an organization to connect resources from multiple projects to a common VPC network. This allows the resources to communicate with each other securely and efficiently using internal IPs from that network.
* VPC Network Peering, in contrast, allows private RFC 1918 connectivity across two VPC networks, regardless of whether they belong to the same project or the same organization.

Consideration 
Across organizations 
Within project 
Network administration 
Shared VPC 
Centralized 
VPC Network Peering 
Yes 
Yes 
Decentralized

* If you want to configure private communication between VPC networks in different organizations, you have to use VPC Network Peering. Shared VPC only works within the same organization.
* if you want to configure private communication between VPC networks in the same project, you have to use VPC Network Peering. This doesn’t mean that the networks need to be in the same project, but they can be. Shared VPC only works across projects.
* Shared VPC is a centralized approach to multi-project networking, because security and network policy occurs in a single designated VPC network. In contrast, VPC Network Peering is a decentralized approach, because each VPC network can remain under the control of separate administrator groups and maintains its own global firewall and routing tables.

Load Balancing and Autoscaling

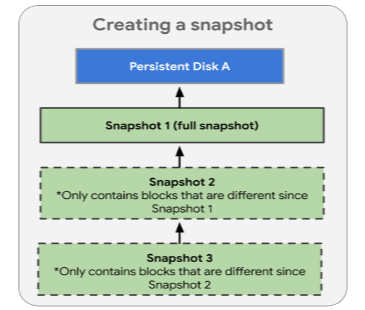
* A **managed instance group** is a collection of identical VM instances that you control as a single entity, using an instance template. You can easily update all the instances in the group by specifying a new template in a rolling update. Also, when your applications require additional compute resources, managed instance groups can automatically scale the number of instances in the group.
* Managed instance groups can work with load balancing services to distribute network traffic to all of the instances in the group. If an instance in the group stops, crashes, or is deleted, the managed instance group automatically recreates the instance so it can resume its processing tasks. The recreated instance uses the same name and the same instance template as the previous instance.
* Regional managed instance groups are generally recommended over zonal managed instance groups because they allow you to spread the application load across multiple zones instead of confining your application to a single zone or you having to manage multiple instance groups across different zones.
* In order to create a managed instance group, you first need to create an instance template.
* managed instance groups offer autoscaling capabilities that allow you to automatically add or remove instances from a managed instance group based on increases or decreases in load.
* A health check is very similar to an uptime check in Stackdriver. You just define a protocol, port, and health criteria, Based on this configuration, GCP computes a health state for each instance.
* GCP’s HTTP(S) load balancing provides global load balancing for HTTP(S) requests destined for your instances.
* This load balancer supports both IPv4 and IPv6 clients, is scalable, requires no pre-warming, and enables content-based and cross-region load balancing.
* The backend services contain a health check, session affinity, a timeout setting, and one or more backends.
* HTTP(S) load balancing uses a round-robin algorithm to distribute requests among available instances.
* The Backend Service contain a balancing mode tells the load balancing system how to determine when the backend is at full usage. If all the backends for the backend service in a region are at full usage, new requests are automatically routed to the nearest region that can still handle requests. The balancing mode can be based on CPU utilization or requests per second (RPS).
* Content-based load balancing: The traffic is split by the load balancer based on the URL header as specified in the URL map.
* An HTTP(S) load balancer has the same basic structure as an HTTP load balancer, but differs in the following ways:
  + An HTTP(S) load balancer uses a target HTTPS proxy instead of a target HTTP proxy.
  + An HTTP(S) load balancer requires at least one signed SSL certificate installed on the target HTTPS proxy for the load balancer.
  + The client SSL session terminates at the load balancer.
  + HTTP(S) load balancers support the QUIC transport layer protocol.
* QUIC is a transport layer protocol that allows faster client connection initiation, eliminates head-of-line blocking in multiplexed streams, and supports connection migration when a client's IP address changes.
* To use HTTPS, you must create at least one SSL certificate that can be used by the target proxy for the load balancer. You can configure the target proxy with up to 10 SSL certificates.
* SSL proxy is a global load balancing service for encrypted, non-HTTP traffic. This load balancer terminates user SSL connections at the load balancing layer, then balances the connections across your instances using the SSL or TCP protocols. These instances can be in multiple regions, and the load balancer automatically directs traffic to the closest region that has capacity.
* SSL proxy load balancing supports both IPv4 and IPv6 addresses for client traffic and provides Intelligent routing, Certificate management, Security patching and SSL policies.
  + Intelligent routing means that this load balances can route requests to backend locations where there is capacity.
  + From a certificate management perspective, you only need to update your customer-facing certificate in one place when you need to switch certificates. Also, you can reduce the management overhead for your virtual machine instances by using self-signed certificates on your instances.
  + In addition, if vulnerabilities arise in the SSL or TCP stack, GCP will apply patches at the load balancer automatically in order to keep your instances safe
* TCP proxy is a global load balancing service for unencrypted, non-HTTP traffic. This load balancer terminates your customers’ TCP sessions at the load balancing layer, then forwards the traffic to your virtual machine instances using TCP or SSL. These instances can be in multiple regions, and the load balancer automatically directs traffic to the closest region that has capacity.
* TCP proxy load balancing supports both IPv4 and IPv6 addresses for client traffic. Similar to the SSL proxy load balancer, the TCP proxy load balancer provides Intelligent routing and Security patching.
* Network load balancing is a regional, non-proxied load balancing service. In other words, all traffic is passed through the load balancer, instead of being proxied, and traffic can only be balanced between VM instances that are in the same region, unlike a global load balancer.
* Network load balancing service uses forwarding rules to balance the load on your systems based on incoming IP protocol data, such as address, port, and protocol type. You can use it to load balance UDP traffic and to load balance TCP and SSL traffic on ports that are not supported by the TCP proxy and SSL proxy load balancers.
* A target pool resource defines a group of instances that receive incoming traffic from forwarding rules. When a forwarding rule directs traffic to a target pool, the load balancer picks an instance from these target pools based on a hash of the source IP and port and the destination IP and port. These target pools can only be used with forwarding rules that handle TCP and UDP traffic.
* Internal load balancing is a regional, private load balancing service for TCP- and UDP-based traffic. In other words, this load balancer enables you to run and scale your services behind a private load balancing IP address This means that it is only accessible through the internal IP addresses of virtual machine instances that are in the same region.
* Therefore, use internal load balancing to configure an internal load balancing IP address to act as the frontend to your private backend instances. Because you don’t need a public IP address for your load-balanced service, your internal client requests stay internal to your VPC network and region.
* GCP internal load balancing distributes client instance requests to the backends using a different approach, It uses lightweight load balancing built on top of Andromeda (Google’s network virtualization stack) to provide software-defined load balancing
* Internal load balancing enables you to support use cases such as the traditional 3-tier web services.
* One differentiator between the different GCP load balancers is the support for IPv6 clients. Only the HTTP(S), SSL proxy, and TCP proxy load balancing services support IPv6 clients. IPv6 termination for these load balancers enables you to handle IPv6 requests from your users and proxy them over IPv4 to your backends.



Infrastructure Automation

* Deployment Manager uses a system of highly structured templates and configuration files to document the infrastructure in an easily readable and understandable format. Deployment Manager conceals the actual Cloud API calls, so you don't need to write code and can focus on the definition of the infrastructure.
* Deployment Manager is an infrastructure deployment service that automates the creation and management of GCP resources for you.
* Deployment Manager will deploy resources in parallel. You can even abstract parts of your configuration into individual building blocks or templates that can be used for other configurations.
* In deployment manager, each resource must contain a name, type, and properties:
  + For the name, I am using an environment variable to get the name from the top-level configuration, which makes this template more flexible.
  + For the type, I am defining the API for a VPC network, which is compute.v1.network. You can find all supported types in the documentation or query them within Cloud Shell, as you will explore in the upcoming lab.
  + By definition, an auto mode network automatically creates a subnetwork in each region. Therefore, I am setting the autoCreateSubnetworks property to true.
* Using the selfLink reference for the network name ensures that the VPC network is created before the firewall rule. This is very important because Deployment Manager creates all the resources in parallel, unless you use references. You would get an error without the reference because you cannot create a firewall rule for a non-existing network.
* GCP Marketplace lets you quickly deploy functional software packages that run on GCP. Essentially, GCP Marketplace offers production-grade solutions from third-party vendors who have already created their own deployment configurations based on Deployment Manager. These solutions are billed together with all of your project’s GCP services.
* Google Cloud Marketplace offers production-grade solutions from third-party vendors who have already created their own deployment configurations based on Deployment Manager.
* Terraform enables you to safely and predictably create, change, and improve infrastructure. It is an open-source tool that codifies APIs into declarative configuration files that can be shared among team members, treated as code, edited, reviewed, and versioned.

Managed Services

* Managed services are partial or complete solutions offered as a service. They exist on a continuum between platform as a service and software as a service, depending on how much of the internal methods and controls are exposed. Using a managed service allows you to outsource a lot of the administrative and maintenance overhead to Google, if your application requirements fit within the service offering.
* BigQuery is GCP’s serverless, highly scalable, and cost-effective cloud data warehouse. It is a petabyte-scale data warehouse that allows for super-fast queries using the processing power of Google's infrastructure. Because there is no infrastructure for you to manage, you can focus on uncovering meaningful insights using familiar SQL without the need for a database administrator.
* Cloud Dataflow is a managed service for executing a wide variety of data processing patterns. It’s essentially a fully managed service for transforming and enriching data in stream and batch modes with equal reliability and expressiveness. With Cloud Dataflow, a lot of the complexity of infrastructure setup and maintenance is handled for you. It’s built on Google Cloud infrastructure and autoscales to meet the demands of your data pipelines, allowing it to intelligently scale to millions of queries per second.
* Cloud Dataflow supports fast, simplified pipeline development via expressive SQL, Java, and Python APIs in the Apache Beam SDK, which provides a rich set of windowing and session analysis primitives as well as an ecosystem of source and sink connectors. Cloud Dataflow is also tightly coupled with other GCP services like Stackdriver, so you can set up priority alerts and notifications to monitor your pipeline and the quality of data coming in and out.
* Cloud Dataflow processes stream and batch data. This data could come from other GCP services like Cloud Datastore or Cloud Pub/Sub, which is Google’s messaging and publishing service. The data could also be ingested from third-party services like Apache Avro and Apache Kafka. After you transform the data with Cloud Dataflow, you can analyze it in BigQuery, AI Platform, or even Cloud Bigtable. Using Data Studio, you can even build real-time dashboards for IoT devices.
* Cloud Dataprep is an intelligent data service for visually exploring, cleaning, and preparing structured and unstructured data for analysis, reporting, and machine learning.
* Cloud Dataprep is an integrated partner service operated by Trifacta
* Cloud Dataprep can be leveraged to prepare raw data from BigQuery, Cloud Storage, or a file upload before ingesting it onto a transformation pipeline like Cloud Dataflow. The refined data can then be exported to BigQuery or Cloud Storage for analysis and machine learning.
* Cloud Dataproc is a fast, easy-to-use, fully managed cloud service for running Apache Spark and Apache Hadoop clusters in a simpler way. You only pay for the resources you use with per-second billing. If you leverage preemptible instances in your cluster, you can reduce your costs even further.
* Cloud Dataproc has built-in integration with other GCP services, such as BigQuery, Cloud Storage, Cloud Bigtable, Stackdriver Logging, and Stackdriver Monitoring. This provides you with a complete data platform rather than just a Spark or Hadoop cluster.
* Cloud Dataproc and Cloud Dataflow can both be used for data processing,
* Cloud Dataproc takes less than 90 seconds to start a cluster.
* There are two kinds of VM instance groups: un-managed and managed. Managed instance groups (MIGs) allow you to operate applications on multiple identical VMs. They offer high availability, scalability (using Autoscaling) and automated updates.
* A managed instance group contains identical instances that you can manage as a single entity in a single zone. Managed instance groups maintain high availability of your apps by proactively keeping your instances available, which means in RUNNING state. Managed instance groups support autoscaling, load balancing, rolling updates, autohealing, and more.
* You can also create regional managed instance groups, which contain instances across multiple zones within the same region.
* Kubernetes is a set of APIs that you can use to deploy containers on a set of nodes called a cluster.
* A Pod is the smallest unit in Kubernetes that you create or deploy. A Pod represents a running process on your cluster as either a component of your application or an entire app.
* Generally, you only have one container per pod, but if you have multiple containers with a hard dependency, you can package them into a single pod and share networking and storage. The Pod provides a unique network IP and set of ports for your containers, and options that govern how containers should run.
* Cloud Events are things that happen in your cloud environment. These might be things like changes to data in a database, files added to a storage system, or a new virtual machine instance being created.
* Cloud Functions are written in Javascript, Python or Go and execute in a managed Node.js environment on Google Cloud Platform. Events from Cloud Storage and Cloud Pub/Sub can trigger Cloud Functions asynchronously, or you can use HTTP invocation for synchronous execution.
* Snapshot types:
* 
* Compute Engine stores multiple copies of each snapshot across multiple locations with automatic checksums to ensure the integrity of your data. Use IAM roles to share snapshots across projects.
* By default, Pods in a Deployment are only accessible inside your GKE cluster. To make them publicly available, you can connect a load balancer to your Deployment by running the kubectl expose command
* In GKE, the load balancer is created as a Network Load Balancer.
* Instances are the basic building blocks of App Engine, providing all the resources needed to successfully host your application. This includes the language runtime, the App Engine APIs, and your application's code and memory. Each instance includes a security layer to ensure that instances cannot inadvertently affect each other.
* Instances are resident or dynamic. A dynamic instance starts up and shuts down automatically based on the current needs. A resident instance runs all the time, which can improve your application's performance. Both dynamic and resident instances instantiate the code included in an App Engine service version.
* If you use manual scaling for an app, the instances it runs on are resident instances. If you use either basic or automatic scaling, your app runs on dynamic instances.
* Basic scaling (dynamic instances) is ideal for work that is intermittent or driven by user activity.
* Each subnet must have a primary range, and, optionally, up to five secondary range for alias IP. Primary and secondary IP ranges must be RFC 1918 addresses. Within a VPC network, all primary and secondary IP ranges must be unique, but they do not need to be contiguous. For example, the primary range of a subnet can be 10.0.0.0/24 while the primary range of another subnet in the same network can be 192.168.0.0/16. The primary IP range for the subnet can be expanded, but not replaced or shrunk, after the subnet has been created. You can remove and replace a subnet's secondary IP address range only if no instances are using that range.
* The minimum primary or secondary range size is 8 IP addresses. In other words, the longest subnet mask you can use is /29.
* You can expand the primary IP range of an existing subnet by modifying its subnet mask,
* Several services, including App Engine flexible, App Engine standard, and Kubernetes engine have stackdriver monitoring built in. For other services without stackdriver monitoring built in, such as compute engine, there are monitoring agents that can be installed.
* [Google Kubernetes Engine](https://cloud.google.com/kubernetes-engine/) (GKE) provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure. The GKE environment consists of multiple machines (specifically [Google Compute Engine](https://cloud.google.com/compute) instances) grouped together to form a [container cluster](https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-architecture)
* When you run a GKE cluster, you also gain the benefit of advanced cluster management features that Google Cloud provides. These include:
  + [Load-balancing](https://cloud.google.com/compute/docs/load-balancing-and-autoscaling) for Compute Engine instances.
  + [Node Pools](https://cloud.google.com/kubernetes-engine/docs/node-pools) to designate subsets of nodes within a cluster for additional flexibility.
  + [Automatic scaling](https://cloud.google.com/kubernetes-engine/docs/cluster-autoscaler) of your cluster's node instance count.
  + [Automatic upgrades](https://cloud.google.com/kubernetes-engine/docs/node-auto-upgrade) for your cluster's node software.
  + [Node auto-repair](https://cloud.google.com/kubernetes-engine/docs/node-auto-repair) to maintain node health and availability.
  + [Cloud Logging and Monitoring](https://cloud.google.com/kubernetes-engine/docs/how-to/logging) for visibility into your cluster.
* Your [compute zone](https://cloud.google.com/compute/docs/regions-zones/#available) is an approximate regional location in which your clusters and their resources live.
* A [cluster](https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-architecture) consists of at least one cluster master machine and multiple worker machines called nodes. Nodes are [Compute Engine virtual machine (VM) instances](https://cloud.google.com/compute/docs/instances/) that run the Kubernetes processes necessary to make them part of the cluster.
* GKE uses Kubernetes objects to create and manage your cluster's resources. Kubernetes provides the Deployment object for deploying stateless applications like web servers. Service objects define rules and load balancing for accessing your application from the Internet.
* A service account is an account that belongs to your application instead of an individual end user.
* every project must have at least one individual as a member.
* You can also create a custom role using a “curated role” as its base. This means you take a role that is similar to the one you need to create, and then add or remove permissions from a copy of that role until it meets your needs exactly.
* Access scopes are the legacy method of specifying permissions for your VM. Before the existence of IAM roles, access scopes were the only mechanism for granting permissions to service accounts, Although they are not the primary way of granting permissions now, you must still set access scopes when configuring an instance to run as a service account., If you need to change access scopes on an instance, you will need to stop that instance first, and then restart it for the changes to take effect.
* You can grant different groups of VMs in your project different identities. This makes it easier to manage different permissions for each group. You also can change the permissions of the service accounts without having to recreate the VMs.
* Cloud Audit Logging maintains three audit logs for each project, folder, and organization: Admin Activity, System Event and Data Access.
* service level indicators (SLIs), objectives (SLOs) and agreements (SLAs). These measurements describe basic properties of metrics that matter, what values we want those metrics to have and how we'll react if we can't provide the expected service
* An SLI is a service level indicator: A carefully defined quantitative measure of some aspect of the level of service that is provided.
* request latency: how long it takes to return a response to a request as a key SLI. Other common SLIs include the error rate, often expressed as a fraction of all requests received and system throughput, typically measured in requests per second. Another kind of SLI important to SREs is availability or the fraction of the time that a service is usable. It is often defined in terms of the fraction of well-formed requests that succeed.
* Durability: the likelihood that data will be retained over a long period of time is equally important for data storage systems. The measurements are often aggregated: i.e., raw data is collected over a measurement window and then turned into a rate, average, or percentile.

|  |  |  |  |
| --- | --- | --- | --- |
| **SLI** | **Metric** | **Description** | **SLO** |
| Request latency | Front end latency | Measures how long a user is waiting for the page to load. A high latency typically correlates to a negative user experience | 99% of requests from the previous 60 minute period are services in under 3 seconds |
| Error rate | Front end error rate | Measures the error rate experienced by users. A high error rate likely indicates an issue. | 0 Errors in the previous 60 minute period |
| Error rate | Checkout error rate | Measures the error rate experienced by other services calling the checkout service. A high error rate likely indicates an issue. | 0 Errors in the previous 60 minute period |
| Error rate | Currency Service error rate | Measures the error rate experienced by other services calling the currency service. A high error rate likely indicates an issue. | 0 Errors in the previous 60 minute period |
| Availability | Front end success rate | Measures the rate of successful requests as a way to determine the availability of the service. A low success rate likely indicates that users are having a poor experience. | 99% of requests are successful over the previous 60 minute period |

* Indentation is important in YAML
* This base configuration is a great starting point for any Google Cloud resource. The **name** field allows you to name the resource, and the **type** field allows you to specify the Google Cloud resource that you want to create. You can also define properties, but these are optional for some resources.

------------------------------------------------Revision---------------------------------------------------

* Google Cloud VPC networks are global; subnets are regional
* Organization may have more than 1 billing account, every billing account contains 1 or more projects.
* Organization can have 1 or more Stackdriver monitoring Workspace and the workspace can have 1 or more projects to monitor. But the project can't be monitored by more than 1 workspace.
* The first monitored GCP project in a Workspace is called the hosting project, and it must be specified when you create the Workspace. The name of that project becomes the name of your Workspace.
* Preemptible VM: you’ve given Compute Engine permission to terminate it if its resources are needed elsewhere, the per-hour price of preemptible VMs incorporates a substantial discount, so you can save a lot of money with preemptible VMs.
* VPC: use route table to forward traffic within the network even across subnets, also use its firewall to control what network traffic is allowed, also use Shared VPC to share a network or individual subnets with other GCP projects
* Cloud CDN (Content Delivery Network): use Google's globally distributed edge caches to cache content close to your users Or use CDN Interconnect if you’d prefer to use a different CDN
* Load Balancers:
  + Global\External > HTTP(s), SSL Proxy, TCP Proxy and (Network TCP\UDP but this is Regional not global)
  + Internal\Regional > Internal TCP\UDP

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Compute Engine virtual machines use Persistent Disk storage to contain their file systems.
* Your Cloud Storage files are organized into buckets with: (Globally unique name, Storage class, IAM polices and Access Control List ACL, Location region or multi-region, Object versioning and Object lifecycle)
* ACL Access Control List ACLs consists of: Scope, which defines who can perform the specified actions (for example, a specific user or group of users). And a permission, which defines what actions can be performed.
* Regional Storage: enables you to store data at lower cost, its recommended when storing frequently accessed data in the same region as your Compute Engine instances. (better performance)
* Multi-Regional Storage: is geo-redundant (at least two geographic locations separated by at least 100 miles), is appropriate for storing data that is frequently accessed, such as serving website content, interactive workloads, or data supporting mobile and gaming applications.
* Nearline storage is a low-cost, highly durable storage service for storing infrequently accessed data. This storage class is a good choice when you plan to read or modify your data less than once a month because the storage cost is low, but there is an associated retrieval cost. I also recommend using Nearline Storage for backups and serving long-tail multimedia content.
* Coldline Storage is a very-low-cost, highly durable storage service for data archival, online backup, and disaster recovery. However, unlike other "cold" storage services, your data is available within milliseconds, not hours or days. Coldline Storage is the best choice for data that you plan to access at most once a year, due to its lower storage cost but higher retrieval cost. You might use Coldline Storage if you want to archive data or have access in the event of a disaster recovery event.
* you can't change a regional bucket to a multi-regional, and vice versa. but both multi-regional buckets and regional buckets can be changed to coldline or nearline.
* You can Synchronizes a VM directory with a bucket.
* GCP bucket has Object Change Notification like File Listner
* You can control the access to GCP bucket using (IAM, ACLs, Signed URL or Signed policy document)
* Cloud Databases:
  + Cloud Bigtable: is managed NoSQL, accessed using HBase API and it has a native compatibility with big data, Hadoop ecosystems.
  + Cloud SQL: is a managed RDBMS that Offers MySQL and PostgreSQL databases as a service, Automatic replication, Managed backups, Vertical scaling (read and write), Horizontal scaling (read) and Google security
  + Cloud Datastore: is a horizontally scalable NoSQL DB (is best for semi-structured application data that is used in App Engine applications. )
  + Cloud SQL is best for web frameworks and existing applications, like storing user credentials and customer orders.
  + Cloud Spanner is best for large-scale database applications that are larger than 2 TB. ex, financial trading and e-commerce.(relational database structure with non-relational horizontal scale)
  + **Cloud Dataproc** is a fast, easy, managed way to run **Hadoop**, **Spark**, Hive, and Pig on GCP. Easily migrate on-premises Hadoop jobs to the cloud., it's great when you have a dataset **of known size**, or when you want to manage your cluster size yourself
  + Cloud Dataflow offers managed data pipelines, Processes data ETL using Compute Engine instances, it's great when your data shows up in Realtime, Or it’s of unpredictable size or rate, Cloud Dataflow frees you from operational tasks like resource management and performance optimization.
  + BigQuery is a fully managed data warehouse, SQL syntax, enable real-time analysis of your data.
  + Cloud Pub/Sub (Publisher\Subscribers) is scalable, reliable messaging, a fully managed real-time messaging service that allows you to send and receive messages between independent applications, up to 10,000 messages per second,
  + Cloud Datalab offers interactive data exploration, it can analyze data in BigQuery, Compute Engine, and Cloud Storage using Python, SQL, and JavaScript.
  + Cloud Vision API enables developers to understand the content of an image.
  + The Cloud Speech API enables developers to convert audio to text.
  + The Cloud Natural Language API offers a variety of natural language understanding technologies to developers. It can do syntax analysis, breaking down sentences supplied by your users into tokens, identify the nouns, verbs, adjectives, and other parts of speech, and figure out the relationships among the words.
  + The Google Cloud Video Intelligence API allows developers to use Google video analysis technology as part of their applications.
  + Cloud Memorystore for Redis provides a fully managed in-memory data store service built on scalable, secure, and highly available infrastructure managed by Google. Applications running on GCP can achieve extreme performance by leveraging the highly scalable, available, secure Redis service without the burden of managing complex Redis deployments.
  + Cloud Dataprep is an intelligent data service for visually exploring, cleaning, and preparing structured and unstructured data for analysis, reporting, and machine learning.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Kubernetes Multi-zone clusters: Run a single cluster in multiple zones.
* Kubernetes Container Builder lets you create Docker container images from application source code located in Cloud Storage. Container images created by Container Builder are automatically stored in Container Registry.
* Kubernetes > POD: a group of containers that are deployed together with guaranteed network access.
* Kubernetes cluster is a group of machines where Kubernetes can schedule containers in pods. The machines in the cluster are called “nodes.”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* App Engine will scale your application automatically in response to the amount of traffic it receives so you only pay for the resources you use.
* the App Engine flexible environment allows you to customize your runtime and even the operating system of your virtual machine using Dockerfiles.
* App Engine Standard Environment SE starts up faster than the Flixable Environment FE, but that you get less access to the infrastructure in which your application runs.
  + SE has no SSH access but FE has
  + SE not write to local disk but FE do
  + SE access network via App Engine Services only but FE can
  + SE, after free daily use, pay per instance class, with automatic shutdown but FE pay for resource allocation per hour; no automatic shutdown
* Cloud Functions (written in JavaScript) is a lightweight, event-based, asynchronous compute solution that allows you to create small, single-purpose functions that respond to cloud events without the need to manage a server or a runtime environment. (ex: suppose that you want to automatically catch any uploaded file to be renamed then you can create this functionality using Cloud Function)
* Cloud Events are things that happen in your cloud environment. These might be things like changes to data in a database, files added to a storage system, or a new virtual machine instance being created. (you can create Event Trigger to trigger these events to handle something)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* PoPs are where Google’s network is connected to the rest of the internet
* VPC objects include: Projects, Networks (Default, auto mode, custom mode), Subnetworks, Regions, Zones, IP addresses (Internal, external, range), Virtual machines (VMs), Routes, Firewall rules
* VPC Project Contains networks (up to 5) that can be shared/peered.
* VMs should communicate using the Internal IP if they are in the same network even if they are in a different regions, and using external IP if they are in different networks even if they are in the same region.
* Subnetworks can extend across zones within the same region.
* The subnet is simply an IP address range, and you can use IP addresses within that range, the first and second addresses in the range, .0 and .1, are reserved for the network and the subnet’s gateway, respectively.
* FQDN is [hostname].[zone].c.[project-id].internal
* The external IP address is mapped to the VM's internal address transparently by VPC.
* Alias IP Ranges let you assign a range of internal IP addresses as an alias to a virtual machine's network interface. This is useful if you have multiple services running on a VM, and you want to assign a different IP address to each service.
* firewall rule is composed of the following parameters:
  + The direction of the rule. Inbound connections (ingress) & outbound connections(egress).
  + The source of the connection for ingress packets, or the destination of the connection for egress.
  + The protocol and port of the connection.
  + The action of the rule, which is to allow or deny packets that match the direction, protocol, port, and source or destination of the rule.
  + The priority of the rule, which governs the order in which rules are evaluated. The first matching rule is applied.
  + The rule assignment. By default, all rules are assigned to all instances, but you can assign certain rules to certain instances only.
* allocating VMs on a single subnet to separate zones, you get improved availability without additional security complexity. A regional managed instance group contains instances from multiple zones across the same region, which provides increased availability.
* **Cloud NAT** is Google’s managed network address translation service. It lets you provision your application instances without public IP addresses (egress only)
* if your private VM instance needs to access Google APIs and services like Cloud Storage bucket, you need to enable Private Google Access. OR to create a public IP to access google APIs through it.
* When a VM is created the ephemeral external IP address is assigned from a pool. There is no way to predict which address will be assigned, so there is no way to write a rule that will match that VM's IP address before it is assigned. Tags allow a symbolic assignment that does not depend on order in the IP addresses:
  + You can enable Private Google Access on a subnet.
  + You can connect to a VM instance using an Identity-Aware Proxy (IAP) tunnel

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Cloud Functions is very optimum for Microservice architecture but it only support Paython, NodeJS and Go languages.
* the data that you store on local SSDs persists only until you stop or delete the instance.
* How to Connect to Virtual Machine? Firewall rule tcp:22 for SSH Linux & tcp:3389 for RDP Windows
* One major use case for preemptible VMs is running a batch processing job. If some of those instances terminate during processing, the job slows but does not completely stop. Therefore, preemptible instances complete your batch processing tasks without placing additional workload on your existing instances, and without requiring you to pay full price for additional normal instances.
* Local SSD: Data on these disks will survive a reset but not a VM stop or terminate, because these disks can’t be reattached to a different VM
* I recommend choosing a persistent HDD disk when you don't need performance but just need capacity. If you have high performance needs, start looking at the SSD options
* The metadata server is particularly useful in combination with startup and shutdown scripts, because you can use the metadata server to programmatically get unique information about an instance, without additional authorization.
* Move an instance to a new zone (gcloud compute instances move)
* Snapshot doesn't back up VM metadata, tags, etc.
* Snapshots are available only to persistent disks (HDD & SSD) and not to local SSDs.
* There are three 3 types of roles in Cloud IAM: primitive roles, predefined roles, and custom roles.
* There are three types of service accounts: user-created or custom, built-in, and Google APIs service accounts.
* default service accounts support both primitive and predefined IAM roles, but user-created service accounts only use predefined IAM roles.
* Cloud IAP lets you establish a central authorization layer for applications accessed by HTTPS, so you can use an application-level access control model instead of relying on network-level firewalls.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Difference between Labels and Tags: Labels, we just learned, are user-defined strings in key-value format that are used to organize resources, and they can propagate through billing. Tags, on the other hand, are user-defined strings that are applied to instances only and are mainly used for networking, such as applying firewall rules.
* Stackdriver role assigned to one person on one project applies equally to all projects monitored by that Workspace, So In order to give people different roles per-project and to control visibility to data, consider placing the monitoring of those projects in separate Workspaces.
* Stackdriver Logs are only retained for 30 days, but you can export your logs to Cloud Storage buckets, BigQuery datasets, and Cloud Pub/Sub topics.
* If you want to visualize your logs, I recommend connecting your BigQuery tables to Data Studio. Data Studio transforms your raw data into the metrics and dimensions that you can use to create easy-to-understand reports and dashboards.
* Stackdriver Trace is a distributed tracing system that collects latency data from your applications and displays it in the GCP Console.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Shared VPC is a centralized approach to multi-project networking, because security and network policy occurs in a single designated VPC network.
* Cloud VPN supports both static and dynamic routes. In order to use dynamic routes, you need to configure Cloud Routers.
* Dedicated connections provide a direct connection to Google’s network, but shared connections provide a connection to Google’s network through a partner.
* Layer 2 connections (Dedicated and Partner Interconnect) use a VLAN that pipes directly into your GCP environment
* Layer 3 connections (Direct and Carrier Peering) provide access to G Suite services, YouTube, and Google Cloud APIs using public IP addresses
* Dedicated Interconnect provides direct physical connections between your on-premises network and Google’s network. This enables you to transfer large amounts of data between networks
* Partner Interconnect provides connectivity between your on-premises network and your VPC network through a supported service provider. This is useful if your data center is in a physical location that cannot reach a Dedicated Interconnect colocation facility or if your data needs don't warrant a Dedicated Interconnect.
* GCP’s Edge Points of Presence, or PoPs, are where Google's network connects to the rest of the internet via peering.
* If you require access to Google public infrastructure and cannot satisfy Google’s peering requirements, you can connect via a Carrier Peering partner.
* Shared VPC allows an organization to connect resources from multiple projects to a common VPC network.
* VPC Network Peering, in contrast, allows private RFC 1918 connectivity across two VPC networks, regardless of whether they belong to the same project or the same organization.
* If you want to configure private communication between VPC networks in different organizations, you must use VPC Network Peering. Shared VPC only works within the same organization.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Regional managed instance groups are generally recommended over zonal managed instance groups because they allow you to spread the application load across multiple zones instead of confining your application to a single zone or you having to manage multiple instance groups across different zones.
* HTTP(S) load balancing uses a round-robin algorithm to distribute requests among available instances.
* The Backend Service contain a balancing mode that can be based on CPU utilization or requests per second (RPS).
* Using the selfLink reference for the network name ensures that the VPC network is created before the firewall rule. This is very important because Deployment Manager creates all the resources in parallel, unless you use references. You would get an error without the reference because you cannot create a firewall rule for a non-existing network.