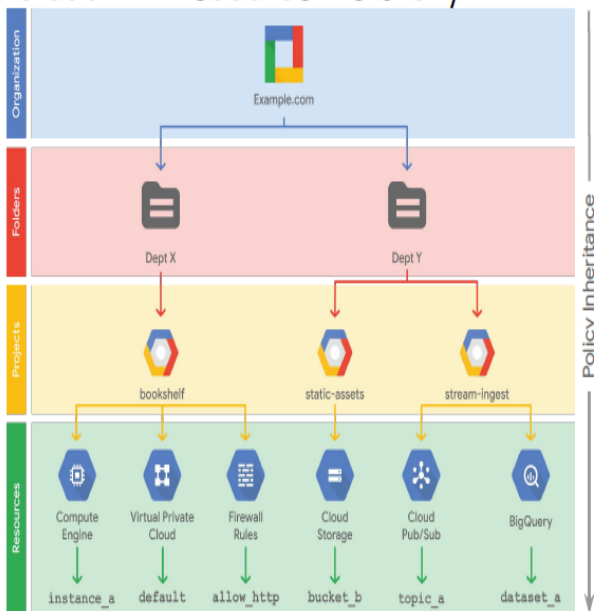


## Cloud IAM objects



## Cloud IAM resource hierarchy

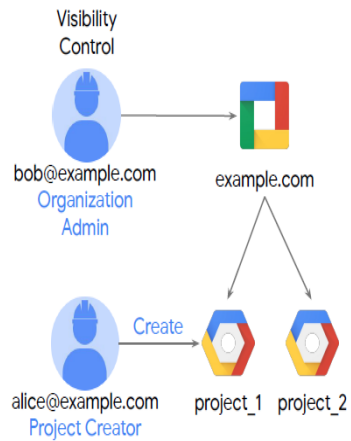


Child policies cannot restrict access granted at the parent level

- \* The Organization node is the root node in this hierarchy, folders are the children of the organization, projects are the children of the folders, and the individual resources are the children of projects.
- \* Each resource has exactly one parent.
- \* Cloud IAM allows you to set policies at all of these levels, where a policy contains a set of roles and role members.
- \* The organization resource represents your company. Cloud IAM roles granted at this level is inherited by all resources under the organization.
- \* The folder resource could represent your department.
- \* Projects represent a trust boundary within your company.
- \* Services within the same project have a default level of trust.
- \* The Cloud IAM policy hierarchy always follows the same path as the GCP resource hierarchy, which means that if you change the resource hierarchy, the policy hierarchy also changes.

## Organization node

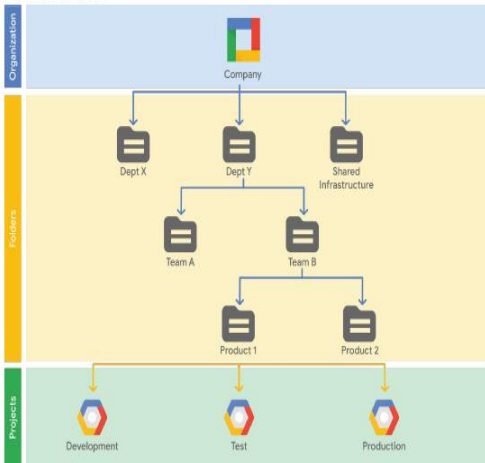
- An organization node is a root node for Google Cloud resources
- Organization roles:
  - Organization Admin: Control over all cloud resources; useful for auditing
  - Project Creator: Controls project creation; control over who can create projects



## Creating and managing organizations

- Created when a **G Suite** or **Cloud Identity** account creates a GCP Project
- **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



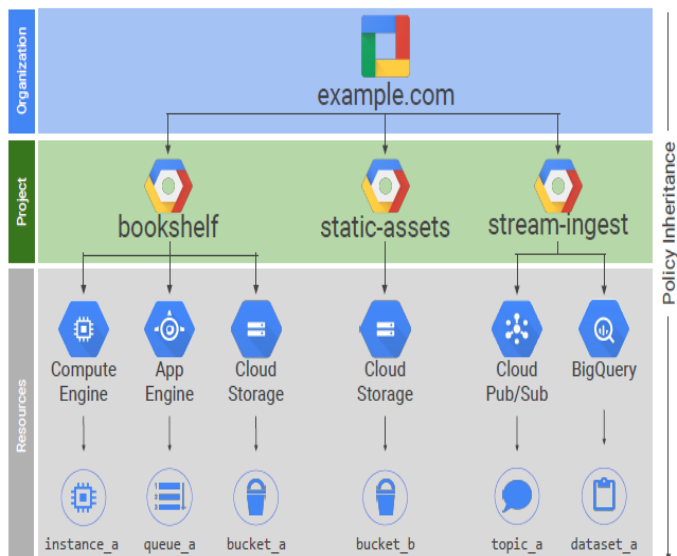
Additional grouping mechanism and isolation boundaries between projects:

- Different legal entities
- Departments
- Teams

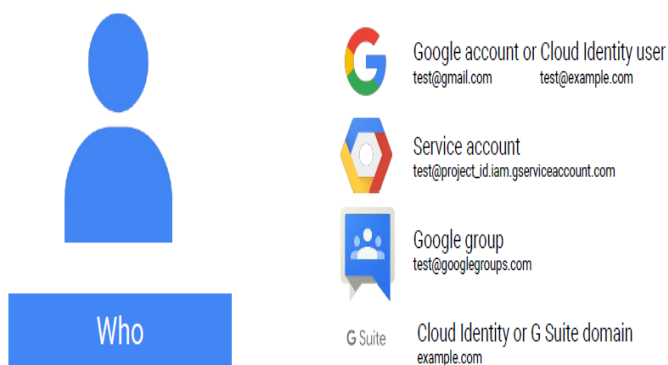
Folders allow delegation of administration rights.

## An example IAM resource hierarchy

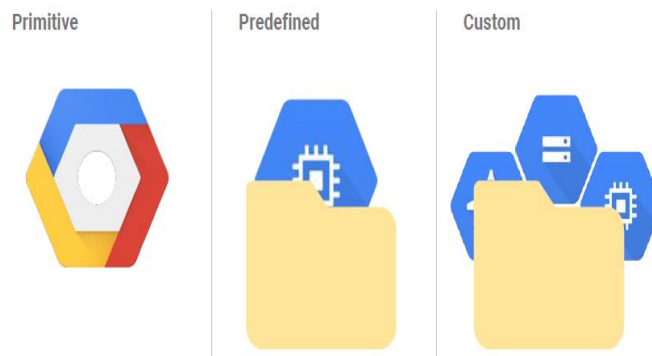
- A policy is set on a resource.
  - Each policy contains a set of roles and role members.
- Resources inherit policies from parent.
  - Resource policies are a union of parent and resource.
- A less restrictive parent policy overrides a more restrictive resource policy.



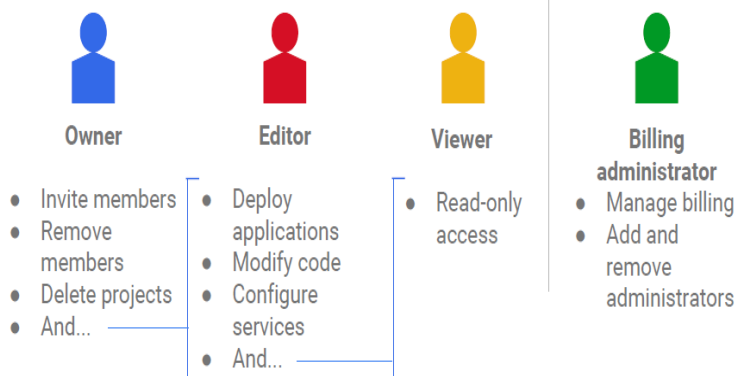
**Who:** IAM policies can apply to any of four types of principals



There are three types of IAM roles



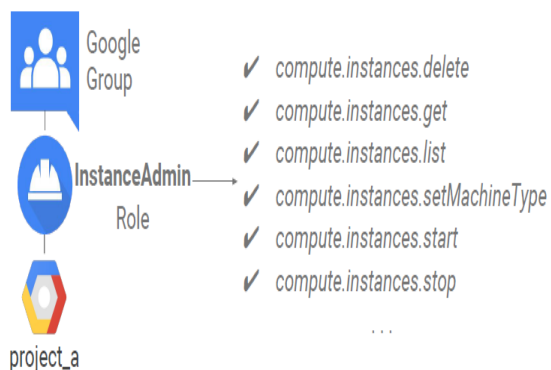
IAM primitive roles offer fixed, coarse-grained levels of access



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

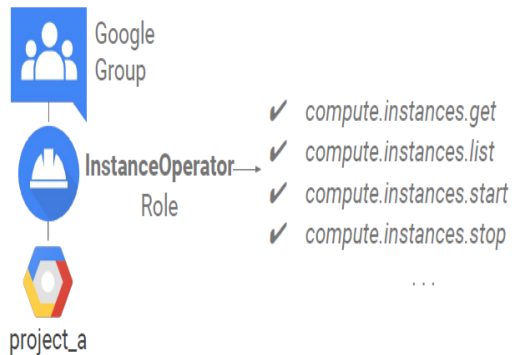
**IAM Primitive Roles** apply across all GCP services in a project. Primitive roles are broad. You apply them to a GCP project, and they affect all resources in that project. IAM primitive roles offer fixed, coarse-grained levels of access. The primitive roles are the Owner, Editor, and Viewer roles.

IAM predefined roles offer more fine-grained permissions on particular services



**IAM Predefined Roles** apply to a particular GCP service in a project. GCP services offers their own sets of predefined roles, and they define where those roles can be applied.

IAM custom roles let you define a precise set of permissions



What if you need something even finer grained?  
That's what **custom roles** permit.

What if you want to give permissions to a Compute Engine virtual machine rather than to a person? That's what service accounts are for. For instance, maybe you have an application running in a virtual machine that needs to store data in Google Cloud Storage.

Service Accounts control server-to-server interactions

- Provide an identity for carrying out **server-to-server** interactions in a project
- Used to **authenticate** from one service to another
- Used to **control privileges** used by resources
  - So that applications can perform actions on behalf of authenticated end users
- Identified with an **email** address:

`PROJECT_NUMBER-compute@developer.gserviceaccount.com`

`PROJECT_ID@appspot.gserviceaccount.com`

**Service Accounts** are named with an email address, but instead of passwords they use cryptographic keys to access resources.

What if you want to give permissions to a Compute Engine virtual machine rather than to a person? That's what service accounts are for. For instance, maybe you have an application running in a virtual machine that needs to store data in Google Cloud Storage.

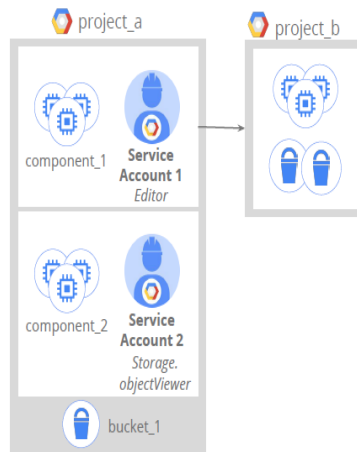
## Service Accounts and IAM

- Service accounts **authenticate using keys**.
  - Google manages keys for Compute Engine and App Engine.
- You can assign a predefined or custom IAM role to the service account.



## Example: Service Accounts and IAM

- VMs running component\_1 are granted **Editor** access to project\_b using *Service Account 1*.
- VMs running component\_2 are granted **objectViewer** access to bucket\_1 using *Service Account 2*.
- Service account permissions can be changed without recreating VMs.



In this simple example, a service account has been granted Compute Engine's Instance Admin role. This would allow an application running in a VM with that service account to create, modify, and delete other VMs.

What can you use to manage your GCP administrative users?

What if you already have a different corporate directory?



There are four ways to interact with GCP

### Cloud Platform Console

Web user interface



### Cloud Shell and Cloud SDK

Command-line interface



### Cloud Console Mobile App

For iOS and Android



### REST-based API

For custom applications



## Quiz Answers

True or False: If a Google Cloud IAM policy gives you Owner permissions at the project level, your access to a resource in the project may be restricted by a more restrictive policy on that resource.

False: Policies are a union of the parent and the resource. If a parent policy is less restrictive, it overrides a more restrictive resource policy.

True or False: All Google Cloud Platform resources are associated with a project.

True: All Google Cloud Platform resources are associated with a project.







## Quiz: Service Accounts

Service accounts are used to provide which of the following?

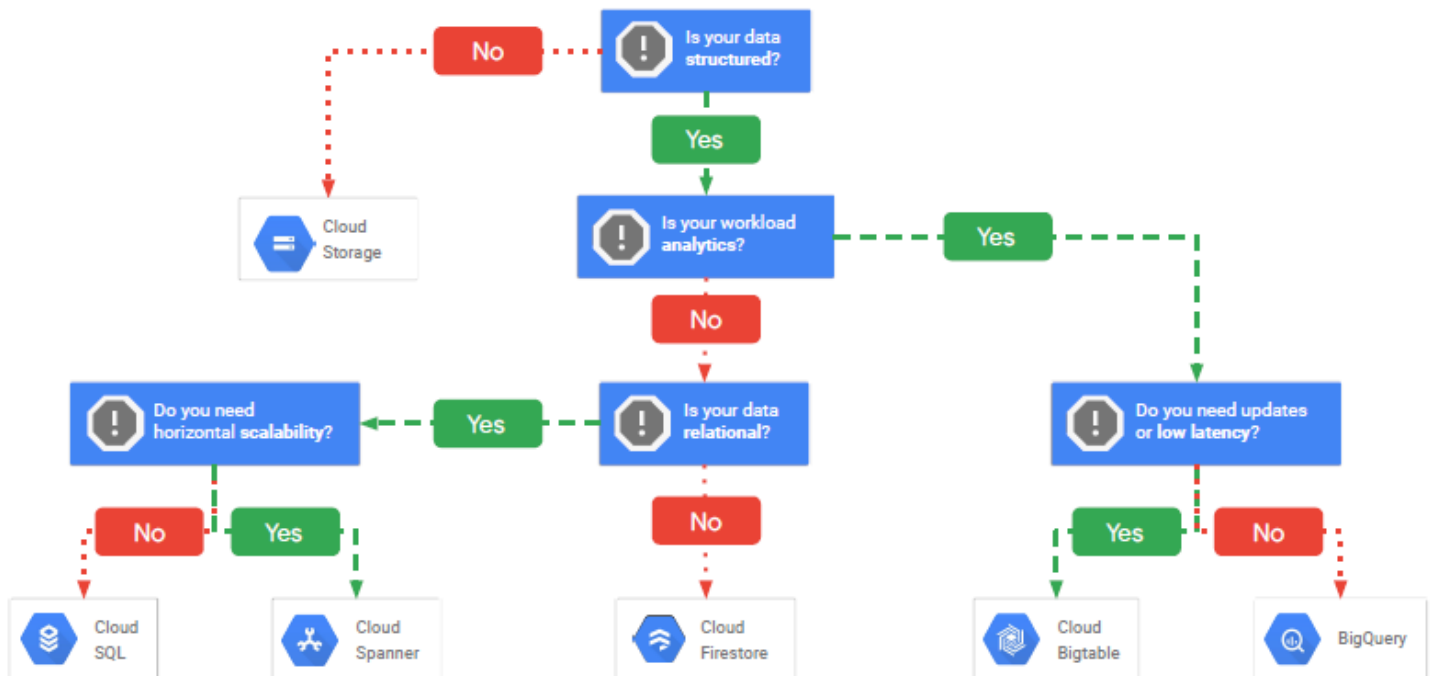
- ☐ Authentication between Google Cloud Platform services
- ☐ Key generation and rotation when used with App Engine and Compute Engine
- ☐ A way to restrict the actions a resource (such as a VM) can perform
- ☐ A way to allow users to act with service account permissions
- ✓ All of the above

## Storage

# Storage and database services

Object	Relational		Non-relational		Warehouse
					
Cloud Storage	Cloud SQL	Cloud Spanner	Cloud Firestore	Cloud Bigtable	BigQuery
<b>Good for:</b> Binary or object data	<b>Good for:</b> Web frameworks	<b>Good for:</b> RDBMS+scale, HA, HTAP	<b>Good for:</b> Hierarchical, mobile, web	<b>Good for:</b> Heavy read + write, events,	<b>Good for:</b> Enterprise data warehouse
<b>Such as:</b> Images, media serving, backups	<b>Such as:</b> CMS, eCommerce	<b>Such as:</b> User metadata, Ad/Fin/MarTech	<b>Such as:</b> User profiles, game state	<b>Such as:</b> AdTech, financial, IoT	<b>Such as:</b> Analytics, dashboards

## Storage and database decision chart





### Cloud Storage

- \*\* Cloud Storage is binary large-object storage
- \*\* High performance, internet-scale
- \*\* Simple administration
  - \*\* Does not require capacity management
- \*\* It's scalable to exabytes of data
- \*\* The time to first byte is in milliseconds
- \*\* It has very high availability across all storage classes
- \*\* And It has a single API across those storage classes
- \*\* Access : **gsutil** command & (RESTful) JSON API or XML API

### Changing default storage classes



- \*\* Default class is applied to new objects
  - \*\* Regional bucket can never be changed to Multi-Regional
  - \*\* Multi-Regional bucket can never be changed to Regional
  - \*\* Objects can be moved from bucket to bucket
  - \*\* Object Lifecycle Management can manage the classes of objects
  - \*\* 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
  - \*\* Data encryption at rest ( Google Cloud Storage always encrypts your data on the server side, before it is written to disk )
  - \*\* Data encryption in transit by default from Google to endpoint ( Data traveling between a customer's device and Google is encrypted by default using HTTPS/TLS (Transport Layer Security)).
  - \*\* The storage objects offered by Google Cloud Storage are "immutable," which means that you do not edit them in place, but instead create a new version.
  - \*\* Online and offline import services are available
  - \*\* Google Cloud Storage's primary use is whenever binary large-object storage is needed: online content, backup and archiving, storage of intermediate results in processing workflows, and more.
  - \*\* Cloud Storage files are organized into **Buckets**
  - \*\* **Bucket attributes** (Globally unique name Storage class, Location (region or multi-region), IAM policies or Access Control Lists, Object versioning setting, Object lifecycle management rules).
  - \*\* **Access Control of Bucket & Objects** : By IAM Role, ACL's, signed cryptographic Keys, Signed Policy Documents
  - \*\* Roles are inherited from project to bucket to object.
  - \*\* You can create access control lists ("ACLs") that offer finer control, ACLs define who has access to your buckets and objects, as well as what level of access they have.
  - \*\* Each ACL consists of two pieces of information: A 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 (for example, read or write).
  - \*\* You create a URL that grants read or write access to a specific Cloud Storage resource and specifies when the access expires.
- gsutil signurl -d 10m path/to/privatekey.p12 gs://bucket/object*
- \*\* Object Change Notification can be used to notify an application when an object is updated or added to a bucket through a watch request.



# Overview of storage classes

	Regional	Multi-Regional	Nearline	Coldline
Design patterns	Data that is <b>used in one region</b> or needs to remain in region	Data that is <b>used globally</b> and has no regional restrictions	Data that is accessed <b>no more than once a month</b>	Data that is accessed <b>no more than once a year</b>
Use case	Data-intensive computations, data governance	Website content, interactive workloads	Backup, long-tail multimedia content	Archiving or disaster recovery
Availability SLA	99.9%	99.95%	99.0%	99.0%
Durability	99.999999999%	99.999999999%	99.999999999%	99.999999999%
Duration	Hot data	Hot data	30-day minimum	90-day minimum
Retrieval cost	none	none	\$	\$\$

Object Import Services :

\*\* **Transfer Appliance** is a hardware appliance you can use to securely migrate large volumes of data (from hundreds of terabytes up to 1 petabyte) to GCP

\*\* **Storage Transfer Service** enables high-performance imports of online data. That data source can be another Cloud Storage bucket, an Amazon S3 bucket, or an HTTP/HTTPS location.

\*\* **Offline Media Import** is a third party service where physical media (such as storage arrays, hard disk drives (HDDs), tapes

## 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

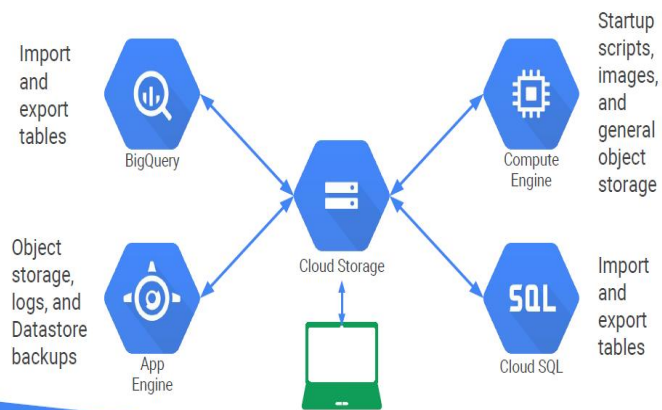
Scheduled, managed batch transfers



### Transfer Appliance <sup>Beta</sup>

Rackable appliances to securely ship your data

## Cloud Storage works with other GCP services

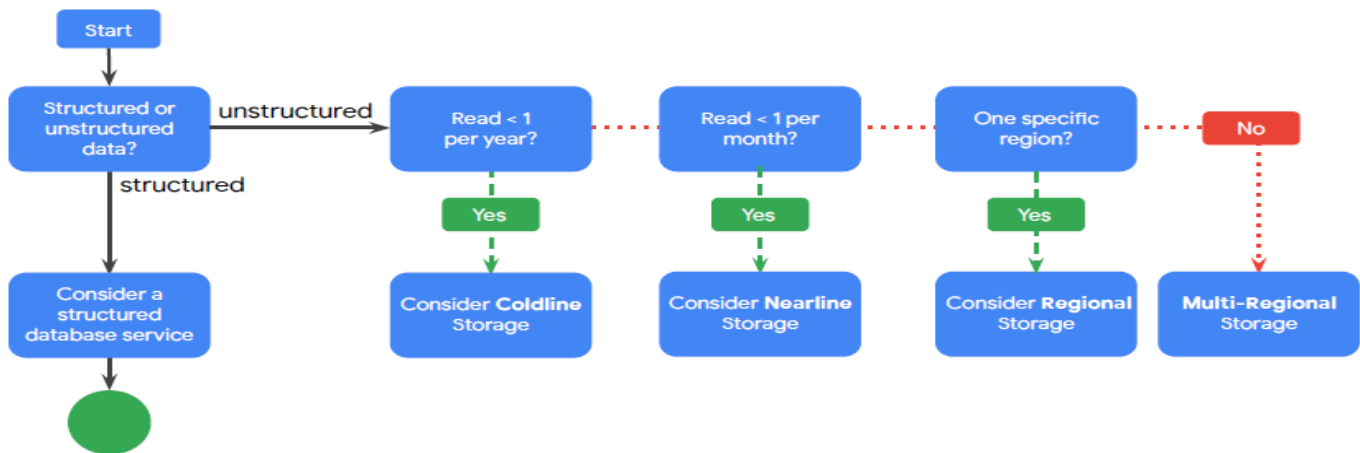


## Cloud Storage provides strong global consistency

- Read-after-write
- Read-after-metadata-update
- Read-after-delete
- Bucket listing
- Object listing
- Granting access to resources



## Choosing a storage class



## Cloud SQL



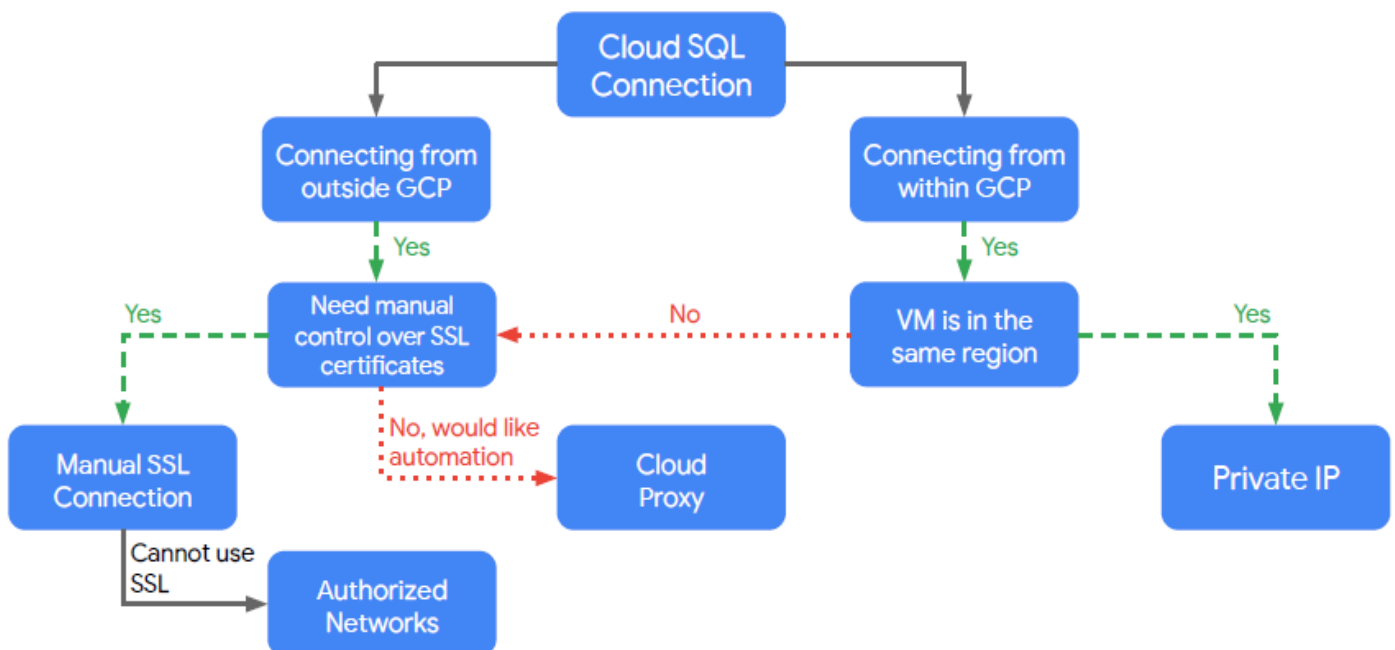
### Cloud SQL

- \*\* Cloud SQL is a fully managed database service (MySQL or PostgreSQL)
- \*\* Patches and updates automatically applied
- \*\* You administer MySQL users
- \*\* Cloud SQL supports many clients
  - \*\* gcloud sql
  - \*\* App Engine, G Suite scripts
  - \*\* Applications and tools
    - \*\* SQL Workbench, Toad
    - \*\* External applications using standard
    - \*\* MySQL drivers
- \*\* **Cloud SQL Performance:** Upto 30 TB of storage, 40,000 IOPS, 416 GB of RAM per instance , scale out with read replicas, scale up to 64 processor cores
- \*\* **Cloud SQL Choice:** MySQL 5.6 or 5.7, PostgreSQL 9.6 or 11.1
- \*\* **Cloud SQL Services:** Replica services, Backup service, Import/export, Scaling (scale up: Machine capacity, scale out: Replicas)
- \*\* There is replica service that can replicate data between multiple zones , 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.

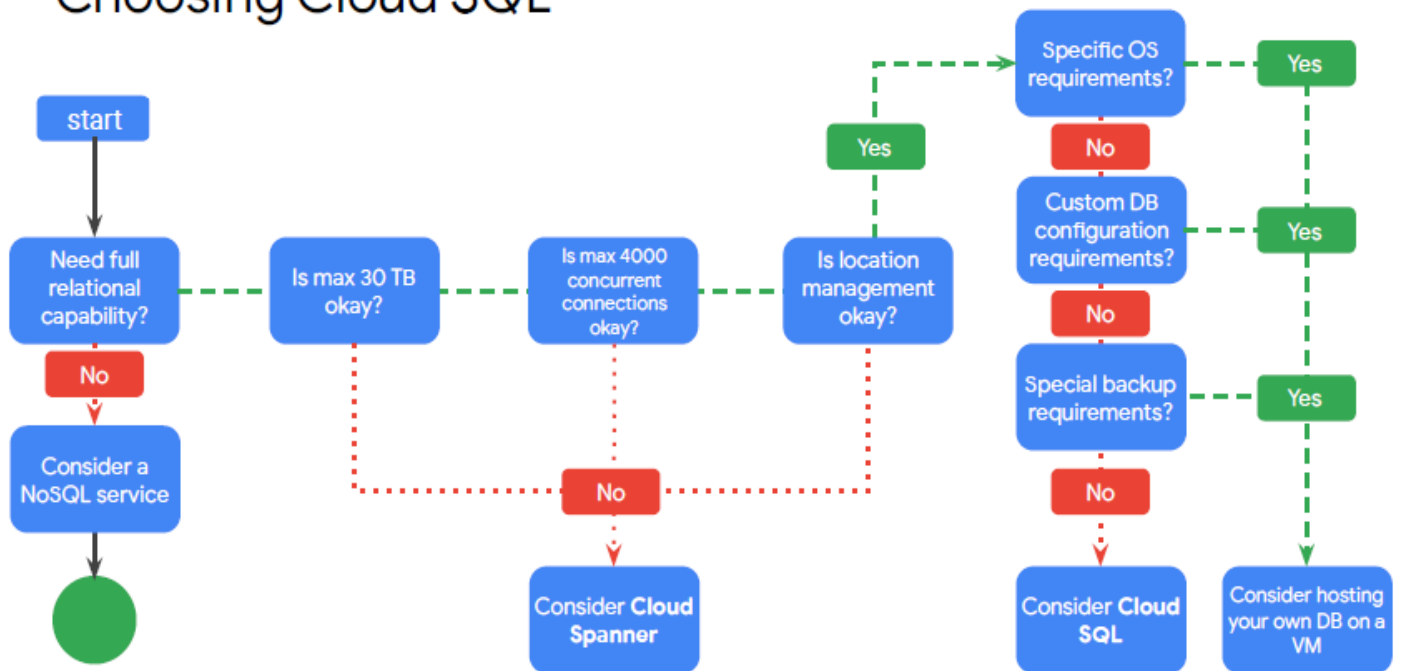
*\*\* If you need more than 30 TB of storage space or over ,4000 concurrent connections to your database, or if you want your application design to be responsible for scaling, availability, and location management when scaling up globally, then consider using Cloud Spanner.*

*\*\* Remember, you can only connect via Private IP if the application and the Cloud SQL server are collocated in the same region and are part of the same VPC network. If your application is hosted in another region, VPC or even project, use a proxy to secure its connection over the external connection.*

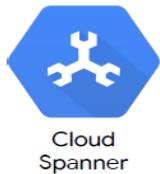
## Connecting to a Cloud SQL instance



# Choosing Cloud SQL



## Cloud Spanner



### Cloud Spanner

- \*\* Cloud Spanner combines the benefits of relational database structure with non-relational horizontal scale
- \*\* Scale to petabytes
- \*\* Strong consistency
- \*\* High availability
- \*\* Used for financial and inventory applications
- \*\* Monthly uptime
  - \*\* Multi-regional: 99.999%
  - \*\* Regional: 99.99%
- \*\* Cloud Spanner provide petabytes of capacity and offers transactional consistency at global scale, schemas, SQL, and automatic, synchronous replication for high availability.
- \*\* Cloud Spanner use cases include financial applications and inventory applications traditionally served by relational database technology.
- \*\* Cloud spanner combine the benefits of relational database structure with non-relational horizontal scale.
- \*\* **Cloud Spanner Architecture:**
  - \*\* 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. This architecture allows for high availability and global placement.
  - \*\* The replication of data will be synchronized across zones using Google's global fiber network. Using atomic clocks ensures atomicity whenever you are updating your data.

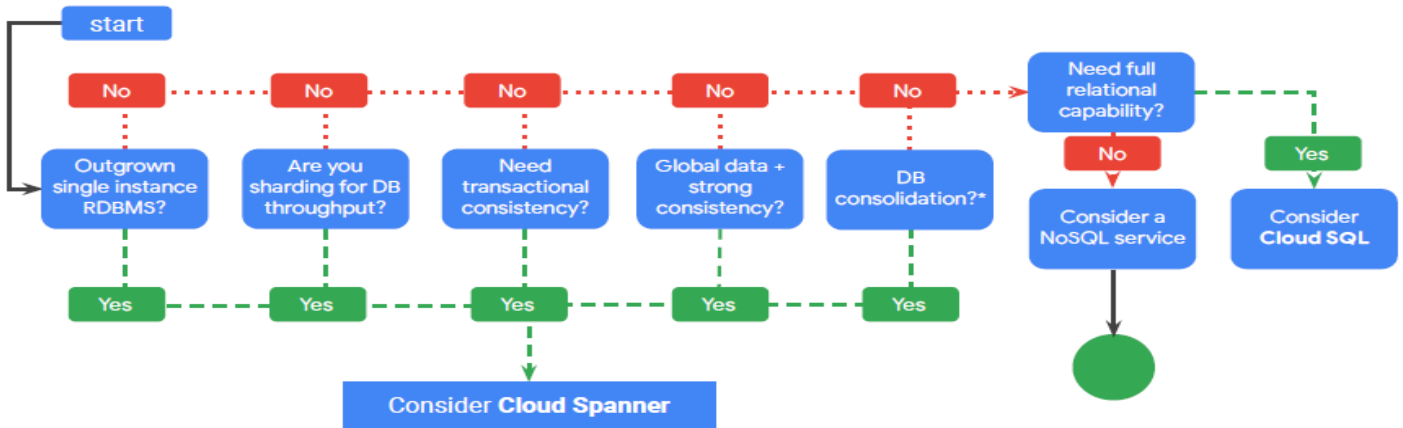
*\*\* If you have outgrown any relational database, are sharding your databases for throughput high performance, need transactional consistency, global data and strong consistency, or just want to consolidate your database, consider using Cloud Spanner.*

*\*\* If you don't need any of these, nor full relational capabilities, consider a NoSQL service such as Cloud Firestore*

## Characteristics

	Cloud Spanner		Relational DB		Non-Relational DB	
Schema	✓	Yes	✓	Yes	✗	No
SQL	✓	Yes	✓	Yes	✗	No
Consistency	✓	Strong	✓	Strong	✗	Eventual
Availability	✓	High	✗	Failover	✓	High
Scalability	✓	Horizontal	✗	Vertical	✓	Horizontal
Replication	✓	Automatic	🔄	Configurable	🔄	Configurable

# Choosing Cloud Spanner



## Cloud FireStore



### Cloud FireStore

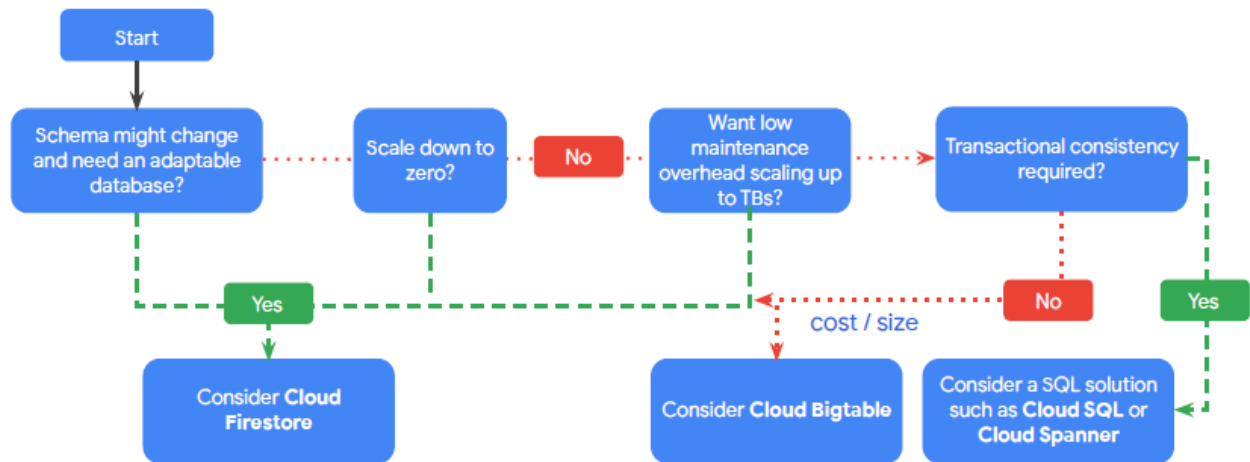
- \*\* Cloud Firestore is a NoSQL document database
- \*\* Simplifies storing, syncing, and querying data
- \*\* Mobile, web, and IoT apps at global scale
- \*\* Live synchronization and offline support
- \*\* Security features
- \*\* ACID transactions
- \*\* Multi-region replication
- \*\* Powerful query engine
- \*\* 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
- \*\* Cloud Firestore's automatic multi-region replication and strong consistency, your data is safe and available, even when disasters strike
- \*\* **Cloud Firestore is the next generation of Cloud Datastore**
  - \*\* Cloud Firestore can operate in Datastore mode
  - \*\* By creating a Cloud Firestore database in Datastore mode, you can access Cloud Firestore's improved storage layer while keeping Cloud Datastore system behavior.
- \*\* **Datastore mode (new server projects):**
  - \*\* Compatible with Datastore applications
  - \*\* Queries are no longer eventually consistent; instead, they are all strongly consistent.
  - \*\* Transactions are no longer limited to 25 entity groups.
  - \*\* Writes to an entity group are no longer limited to 1 per second
- \*\* **Native mode (new mobile and web apps):**
  - \*\* Strongly consistent storage layer
  - \*\* Collection and document data model
  - \*\* Real-time updates
  - \*\* Mobile and Web 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.

\*\* To access all the new Cloud Firestore features, you must use Cloud Firestore in Native mode.

\*\* A general guideline is to use Cloud Firestore in Datastore mode for new server projects, and Native mode for new mobile and web apps.

# Choosing Cloud Firestore



## Cloud BigTable



### Cloud BigTable

\*\* Cloud Bigtable is a NoSQL big data database service

\*\* Petabyte-scale

\*\* Consistent sub-10ms latency

\*\* Seamless scalability for throughput

\*\* Learns and adjusts to access patterns

\*\* Ideal for Ad Tech, Fintech, and IoT

\*\* Storage engine for ML applications

\*\* Easy integration with open source big data tools such as Hadoop , HBase, Cloud Dataflow, and Cloud Dataproc.

\*\* Cloud Bigtable is the same database that powers many of Google ' s core services, including Search, Analytics, Maps, and Gmail.

#### \*\* Cloud Bigtable storage model

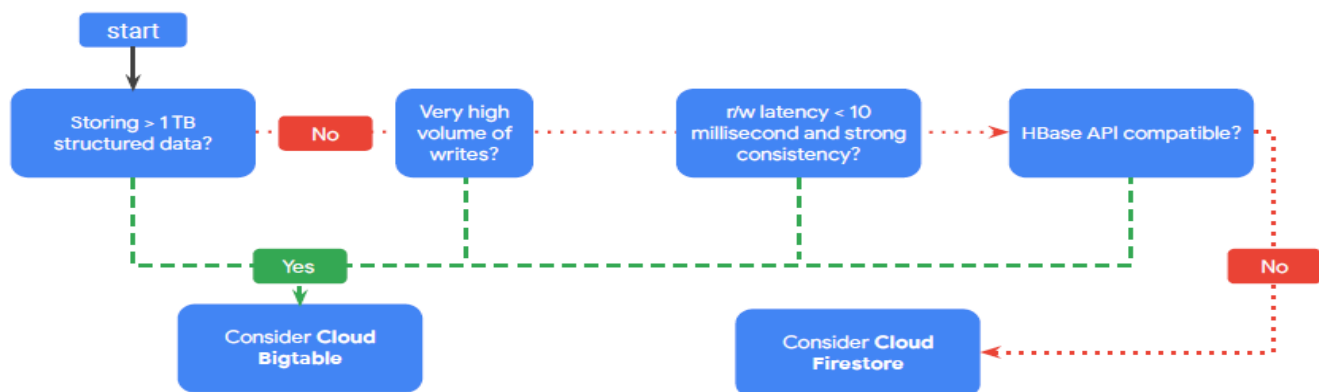
- Cloud Bigtable stores data in massively scalable tables, each of which is a sorted key/value map.
- The table is composed of rows, each of which typically describes a single entity, and columns, which contain individual values for each row.
- Each row is indexed by a single row key, and columns that are related to one another are typically grouped together into a column family.
- Each column is identified by a combination of the column family and a column qualifier, which is a unique name within the column family.
- Cloud Bigtable tables are sparse; if a cell does not contain any data, it does not take up any space.
- A Cloud Bigtable table is sharded into blocks of contiguous rows, called tablets, to help balance the workload of queries. Tablets are similar to HBase regions, for those of you who have used the HBase API.
- Tablets are stored on Colossus, which is Google's file system, in SSTable format. An SSTable provides a persistent, ordered immutable map from keys to values, where both keys and values are arbitrary byte strings.
- Cloud Bigtable learns to adjust to specific access patterns. If a certain Bigtable node is frequently accessing a certain subset of data , Cloud Bigtable will update the indexes so that other nodes can distribute that workload evenly.
- Throughput scales linearly, so for every single node that you do add, you're going to see a linear scale of throughput performance, up to hundreds of nodes.

-----  
\*\* 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 storage service that is compatible with the HBase API, consider using Cloud Bigtable.

\*\* If you don ' t need any of these and are looking for a storage service that scales down well, consider using Cloud Firestore.

\*\* the smallest Cloud Bigtable cluster you can create has three nodes and can handle 30,000 operations per second.

## Choosing Cloud Bigtable



- Bigtable scales UP well
- Cloud Firestore scales DOWN well



## Cloud MemoryStore



Cloud MemoryStore

### **Cloud MemoryStore**

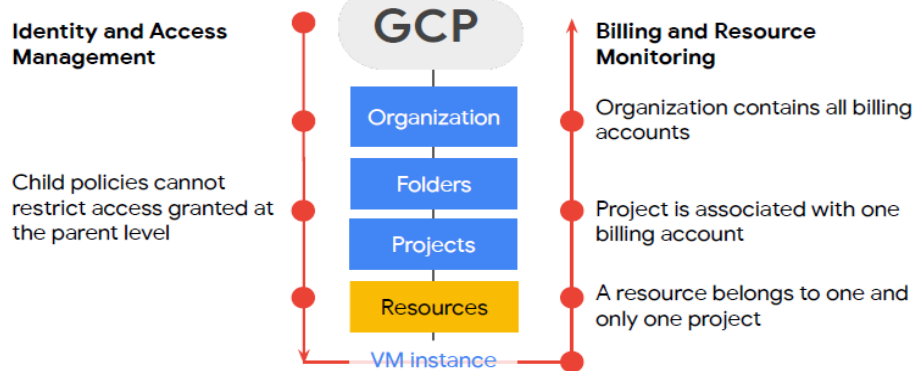
- \*\* Cloud Bigtable is a NoSQL big data database service
  - \*\* In-memory data store service
  - \*\* Focus on building great apps
  - \*\* High availability, failover, patching, and monitoring
  - \*\* Sub-millisecond latency
  - \*\* Instances up to 300 GB
  - \*\* Network throughput of 12 Gbps
  - \*\* Easy Lift-and-Shift
  - \*\* Cloud MemoryStore for Redis provides a fully managed in-memory data store service built on scalable, secure, and highly available infrastructure
  - \*\* Cloud MemoryStore also automates complex tasks like enabling high availability, failover, patching, and monitoring.
- High availability instances are replicated across two zones and provide a 99.9% availability SLA.
- \*\* Cloud MemoryStore easily achieve the sub-millisecond latency and throughput your applications need. Start with the lowest tier and smallest size, and then grow your instance effortlessly with minimal impact to application availability.
  - \*\* Cloud MemoryStore can support instances up to 300 GB and network throughput of 12 Gbps.
  - \*\* Cloud MemoryStore for Redis is fully compatible with the Redis protocol, you can lift and shift your applications from open source Redis to Cloud MemoryStore without any code changes by using the import/export feature.
-

# **GCP Resource Management**

Organization >> Folders >> Projects >> Resources (VM, Storage etc)

- Child policies cannot restrict access granted at the parent level.
- Resource manager lets you hierarchically manage resources by project, folder, and organization
- Policies contain a set of roles and members, and policies are set on resources. These resources inherit policies from their parent. Therefore, resource policies are a union of parent and resource.

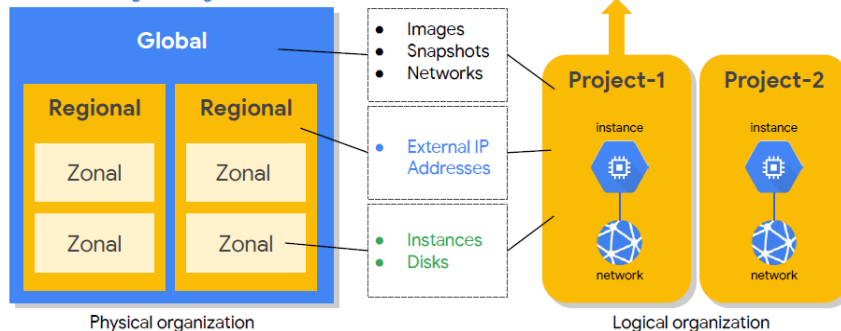
- **Resource Manager** lets you hierarchically manage resources



- IAM policies are inherited top-to-bottom, billing is accumulated from the bottom up.
- Each project is associated with one billing account, which means that an organization contains all billing accounts.
- An organization node is the root node for all Google Cloud Platform resources.
- Project accumulates the consumption of all its resources
  - Track resource and quota usage
    - Enable billing
    - Manage permissions and credentials
    - Enable services and APIs
  - Projects use three identifying attributes:
    - Project Name, which is a human-readable way to identify your projects
    - Project Number, which is automatically generated by the server and assigned to your project.
    - Project ID, also known as Application ID, which is a unique ID that is generated from your project name

## Resource hierarchy

Resources are global, regional, or zonal.



- Physical organization
- Logical organization
- From a physical organization standpoint, resources are categorized as global, regional, or zonal.
  - Images, snapshots, and networks are global resources
  - External IP addresses are regional resources
  - instances and disks are zonal resources

- **Project Quotas**
  - 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? (Ex: you can only have 5 VPC networks / project)
    - How quickly you can make API requests in a project or rate limits. ( Ex: 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.
  - Why use project quotas?
    - Prevent runaway consumption in case of an error or malicious attack
    - Prevent billing spikes or surprises
    - Forces sizing consideration and periodic review
  - *quotas are the maximum amount of resources you can create for that resource type if those resources are available. Quotas do not guarantee that resources will always be available .*
- 

- **Labels & Names**
  - 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.
  - Use labels for ...
    - Team or Cost Center ( team: marketing, team: research )
    - Components (component: redis, component: frontend )
    - Environment or stage (environment: prod, environment: test)
    - Owner or contact (owner: Gaurav, contact:opm)
    - State (state:inuse, state:readyfordeletion )
  - **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, primarily used for networking (applying firewall rules)
- 

- **Billing & E-Mail Alerts**
- For project planning and controlling costs, you can set a budget.
  - you set a budget name and specify which project this budget applies to.
  - you can set the budget at a specific amount or match it to the previous month's spend.
  - you determine your budget amount; you can set the budget alerts.

### Visualize GCP spend with Data Studio

### Budgets and email alerts

Programmatic Budgets: Cloud Pub/Sub → Cloud Functions

- *In addition to receiving an email, you can use Cloud Pub/Sub notifications to programmatically receive spend updates about this budget. You could even create a Cloud Function that listens to the Pub/Sub topic to automate cost management.*



## *Resource Monitoring*



Stackdriver

### **StackDriver**

- Stackdriver has services for monitoring, logging, error reporting, fault tracing, and debugging.
- 

-----



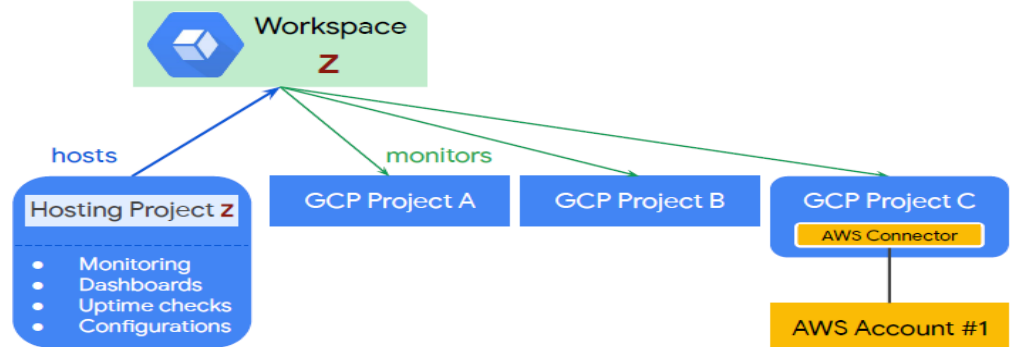
## Monitoring

### StackDriver Monitoring

- Stackdriver dynamically configures monitoring after resources are deployed.
- Stackdriver allows you to monitor your platform, system, and application metrics by ingesting data, such as metrics, events, and metadata.
- You can then generate insights from this data through dashboards, charts, and alerts & and measure uptime and health checks that send alerts via emails.



- 
- 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.
- A Workspace contains the custom dashboards, alerting policies, uptime checks, notification channels, and group definitions that you use with your monitored projects.
- A Workspace can access metric data from its monitored projects, but the metric data and log entries remain in the individual projects.
- The first monitored GCP project in a Workspace is called the *hosting project*, and it must be specified when you create the Workspace.
- Stackdriver role assigned to one person on one project applies equally to all projects monitored by that Workspace.
- Stackdriver Monitoring allows you to create *custom dashboards* that contain charts of the metrics that you want to monitor.
- Stackdriver Monitoring allows you to create *alerting policies* that notify you when specific conditions are met, you or someone else can be automatically get *notification* through email, SMS, or other channels in order to troubleshoot.
- Stackdriver Monitoring allows you to create an alerting policy that monitors your Stackdriver usage and alerts you when you approach the threshold for billing.
- Uptime checks can be configured to test the availability of your public services from locations around the world
- The type of uptime check can be set to *HTTP*, *HTTPS*, or *TCP*.
- 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.
- 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.
- 
- Install Monitoring Agent : `curl -O https://repo.stackdriver.com/stack-install.sh`  
`sudo bash stack-install.sh --write-gcm`

- 
- *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*
  - *If the standard metrics provided by Stackdriver monitoring do not fit your needs, you can create custom metrics.*

# Creating an alert

### Create new alerting policy

#### 1 Conditions

Basic Conditions

HTTP check on instance summer01  
Violates when: Uptime Check Health on Instance (GCE) summer01 fails  
[Edit](#) [Delete](#)

+ Add Another Condition

#### 2 Notifications (optional)

When alerting policy violations occur, you will be notified via these channels. [Learn more](#)

email

+ Add Another Notification

#### 3 Documentation (optional)

When email notifications are sent, they'll include any text entered here. This can convey useful information about the problem and ways to approach fixing it.

[Edit](#) [Preview](#) [Markdown Formatting Help](#)

<b> Main Server health check failed </b>  
+ Server named summer01 failed a Stackdriver uptime check  
+ IP Address of the server is: 104.197.58.79

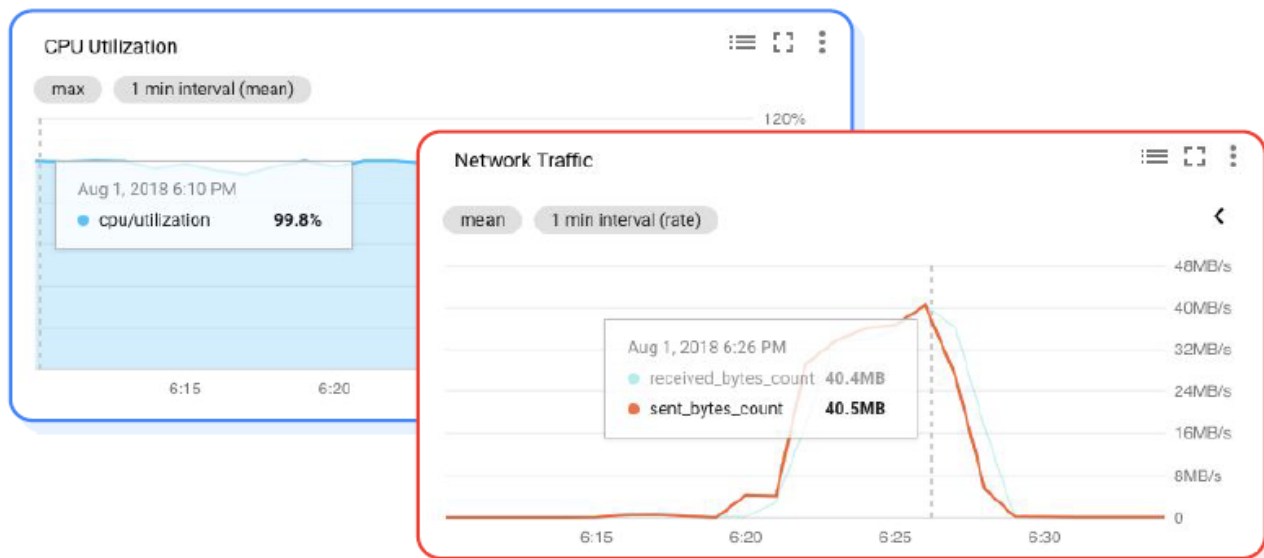
#### 4 Name this policy

A policy's name is used in identifying which policies were triggered, as well as managing configurations of different policies.

Uptime Check Policy

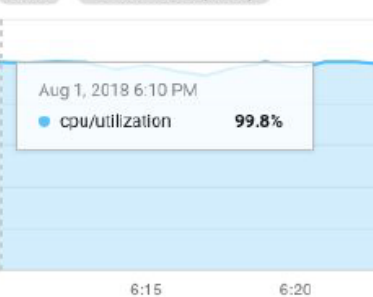
[Save Policy](#) [Cancel](#)

## Dashboards visualize utilization and network traffic



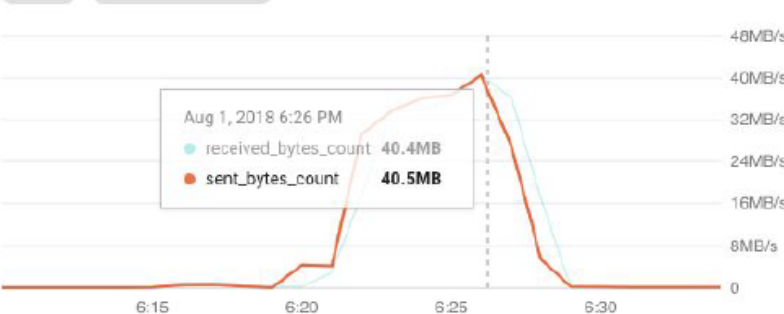
CPU Utilization

max 1 min interval (mean)



Network Traffic

mean 1 min interval (rate)



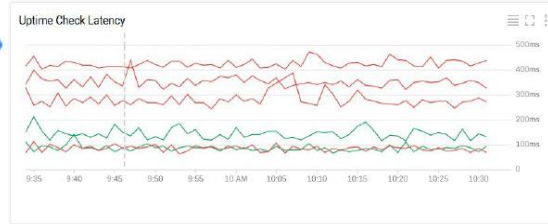
## Monitoring Agent



VM



## Uptime check example



Uptime	100.000%	Outages	0 minutes
Location Results		All locations passed	
Check config			
Check Type	HTTP		
Resource	summer01		
Path	/		
Check Every	1 minute		
Port	80		
Locations	Global		
Timeout	10 seconds		



Logging

### Stackdriver Logging

- Stackdriver has services for monitoring, logging, error reporting, fault tracing, and debugging.
- Stackdriver Logging allows you to store, search, analyze, monitor, and alert on log data and events from GCP and AWS.
- Stackdriver Logging is a fully managed service that performs at scale and can ingest application and system log data from thousands of VMs.
- Stackdriver Logging includes following :

*Storage for logs*

*A user interface called the Logs Viewer*

*An API to manage logs programmatically*

*Service to read and write log entries*

*Log search/view/filter*

*Log-based metrics*

*Monitoring alerts can be set on log events*

*Data can be exported to Cloud Storage, BigQuery, and Cloud Pub/Sub*

*(default log retention 30 days)*

## Analyze logs in BigQuery and visualize in Data Studio

Results							
Row	vpc_name	bytes	subnetwork_name	dest_ip	src_ip	dest_port	protocol
1	vpc-demo	23529368	vpc-demo-web	74.125.28.95	10.1.1.2	443.0	6.0
2	vpc-demo	15237089	vpc-demo-web	74.125.197.95	10.1.1.2	443.0	6.0
3	vpc-demo	4390076	vpc-demo-web	74.125.135.95	10.1.1.2	443.0	6.0
4	vpc-demo	1606002	vpc-demo-web	74.125.199.95	10.1.1.2	443.0	6.0
5	vpc-demo	1479280	vpc-demo-web	108.177.98.95	10.1.1.2	443.0	6.0
6	vpc-demo	828169	vpc-demo-web	173.194.202.95	10.1.1.2	443.0	6.0
7	null	150991	null	10.1.1.2	151.101.52.204	48868.0	6.0
8	null	18024	null	10.1.1.2	74.125.199.95	37910.0	6.0
9	null	17573	null	10.1.1.2	74.125.199.139	58010.0	6.0
10	null	16687	null	10.1.1.2	74.125.28.95	46118.0	6.0

- 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, such as your network traffic
- You can also export logs to Cloud Pub/Sub. This enables you to stream logs to applications or endpoints.
- Install Logging agent :

```
curl -sSO https://dl.google.com/cloudagents/install-logging-agent.sh
```

```
sudo bash install-logging-agent.sh
```

Stackdriver's Monitoring agent, it's a best practice to install the Logging agent on all your VM instances. This agent is supported for Compute Engine and EC2 instances.



BigQuery



Data Studio

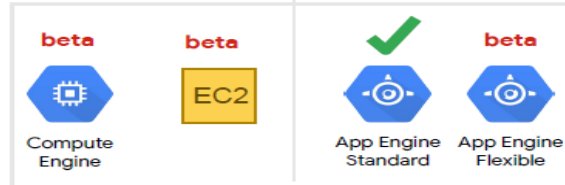




## Error Reporting

### StackDriver Error Reporting

- Stackdriver has services for monitoring, logging, error reporting, fault tracing, and debugging.
- Stackdriver Error Reporting counts, analyzes, and aggregates the errors in your running cloud services.
- Stackdriver Error Reporting a centralized error management interface displays the results with sorting and filtering capabilities, and you can even set up real-time dashboards & notifications when new errors are detected.
- Stackdriver Error Reporting is Generally Available for the App Engine standard environment and is a Beta feature for App Engine flexible environment, Compute Engine, and AWS EC2.
- The exception stack trace parser can process Go, Java, .NET, Node.js, PHP, Python, and Ruby.



•



## Trace

### StackDriver Trace

- Stackdriver has services for monitoring, logging, error reporting, fault tracing, and debugging.
- Stackdriver Trace is a distributed tracing system that collects latency data from your applications and displays it in the GCP Console.
- You can track how requests propagate through your application and receive detailed near real-time performance insights.
- 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.



## Debugger

### StackDriver Debugger

- Stackdriver has services for monitoring, logging, error reporting, fault tracing, and debugging.
- 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.
- Stackdriver debugger adds less than 10ms to the request latency when the application state is captured.
- Stackdriver debugger can take debugger snapshots to capture call stack and local variables of a running application.
- Stackdriver debugger can inject debug logpoints can inject logging into a service without stopping it.
- Stackdriver Debugger supports multiple languages, including Java, Python, Go, Node.js and Ruby.

## **GCP : Interconnecting Networks**

Compute Engine offers managed virtual machines

- High CPU, high memory, standard and shared-core machine types
- Persistent disks
  - Standard, SSD, local SSD
  - Snapshots
- Resize disks with no downtime
- Instance metadata and startup scripts



- Per-second billing, sustained use discounts, committed use discounts
- Preemptible instances
- High throughput to storage at no extra cost
- Custom machine types: Only pay for the hardware you need

Compute Engine bills by the second for use of virtual machines, with a one-minute minimum. For each VM that you run for more than 25% of a month, Compute Engine automatically gives you a discount for every incremental minute. You can get up to a 30% net discount for VMs that run the entire month.

**Committed use discounts** : If your workload is stable and predictable, you can purchase a specific number of vCPUs and memory for up to a 57% discount off of normal prices in return for committing to a usage term of 1 year or 3 years.

**Preemptible VM** is different from an ordinary Compute Engine VM in only one respect: you 've given Compute Engine permission to terminate it if its resources are needed elsewhere. You can save a lot of money with preemptible VMs, although be sure to make your job able to be stopped and restarted.

Scale up or scale out with Compute Engine



*Use big VMs for memory- and compute-intensive applications*



*Use Autoscaling for resilient, scalable applications*

- maximum number of virtual CPUs in a VM was 96, and the maximum memory size was in beta at 624.
- big VMs are great for workloads like in-memory databases and CPU-intensive analytics
- Google VPC supports several different kinds of load balancing



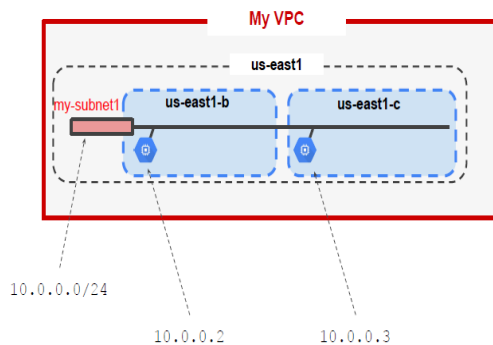
### Virtual Private Cloud Networking

Each VPC network is contained in a GCP project.

VPC networks connect your GCP resources to each other and to the internet.

You can provision GCP resources, connect them to each other, and isolate them from one another.

Google Cloud VPC networks are global; subnets are regional



- Google Virtual Private Cloud networks that you define have global scope.
- They can have subnets in any GCP region worldwide.
- Subnets can span the zones that make up a region.
- You can also have resources in different zones on the same subnet.
- You can dynamically increase the size of a subnet in a custom network by expanding the range of IP addresses allocated to it.
- **VPC Network Topology :**
- **Route table** to forward traffic within the network, even across subnets.
- **Firewall** to control what network traffic is allowed.
- **Shared VPC** to share a network, or individual subnets, with other GCP projects.
- **VPC Peering** to interconnect networks in GCP projects.



### Cloud DNS

\*\* DNS is what translates Internet hostnames to addresses, and as you would imagine.

\*\* Most famous Google services that people don't pay for is 8.8.8.8, which provides a public DNS to world.

\*\* Cloud DNS is highly available and scalable.

\*\* It's managed DNS service running on the same infrastructure as Google.

\*\* Cloud DNS is also programmable. You can publish and manage millions of DNS zones and records using the GCP Console, the command-line interface, or the API.

\*\* Create managed zones, then add, edit, delete DNS records.

Programmatically manage zones and records using RESTful API or command-line interface



### Cloud CDN (Content Delivery Network)

\*\* Google has a global system of edge caches. You can use this system to accelerate content delivery in your application using Google Cloud CDN.

\*\* Applications will experience lower network latency, the origins of your content will experience reduced load, and you can save money too. Once you've set up HTTP(S) Load Balancing, simply enable Cloud CDN with a single checkbox.

\*\* GCP's CDN Interconnect partner program

\*\*

## Google Cloud Platform offers many interconnect options



### VPN

Secure multi-Gbps connection over VPN tunnels



### Direct Peering

Private connection between you and Google for your hybrid cloud workloads



### Dedicated Interconnect

Connect N X 10G transport circuits for private cloud traffic to Google Cloud at Google POPs  
*SLAs available*



### Carrier Peering

Connection through the largest partner network of service providers



### Partner Interconnect

Connectivity between your on-premises network and your VPC network through a supported service provider  
*SLAs available*



### Cloud CDN (Content Delivery Network)

\*\* some customers don't want to use the Internet, either because of security reasons or they need more reliable bandwidth.

\*\* **Direct Peering** means putting a router in the same public datacenter as a Google point of presence and exchanging traffic. Google has more than 100 points of presence around the world.

\*\* **Carrier Peering** means Customers who aren't already in a point of presence can contract with a partner in the Carrier Peering program to get connected.

\*\* **Dedicated Interconnect** means customers will get one or more direct, private connections to Google. If these connections have topologies that meet Google's specifications, they can be covered by up to a 99.99% SLA. These connections can be backed up by a VPN for even greater reliability.

\*\* **Partner Interconnect** provides connectivity between your on-premises network and your VPC network through a supported service provider. A Partner Interconnect connection is useful if your data center is in a physical location that can't reach a Dedicated Interconnect colocation facility or if your data needs don't warrant an entire 10 Gbps connection. 99.99% SLA

\*\*

**Q : Name 3 networking services available to your applications on Google Cloud Platform.**

A : Cloud Virtual Network, Cloud Interconnect, Cloud DNS, Cloud Load Balancing, and Cloud CDN.

**Q : Name 3 Compute Engine pricing features.**

A: Per-second billing, custom machine types, preemptible instances.

**Q : True or False: Google Cloud Load Balancing lets you balance HTTP traffic across multiple Compute Engine regions.**

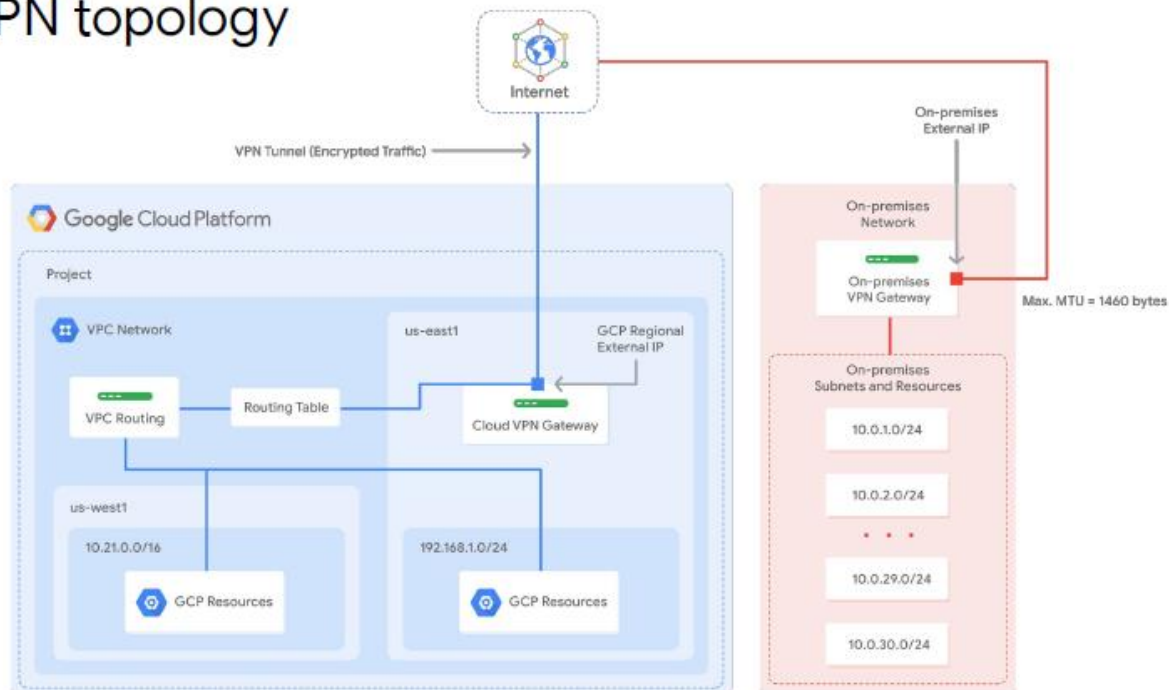
A : True.



## Cloud VPN

- 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. This protects your data as it travels over the public internet.
- Useful for low-volume data connections.
- 99.9% SLA
- Cloud VPN supports:
  - Site-to-site VPN
  - Static routes & Dynamic routes (Cloud Router)
  - IKEv1 and IKEv2 ciphers

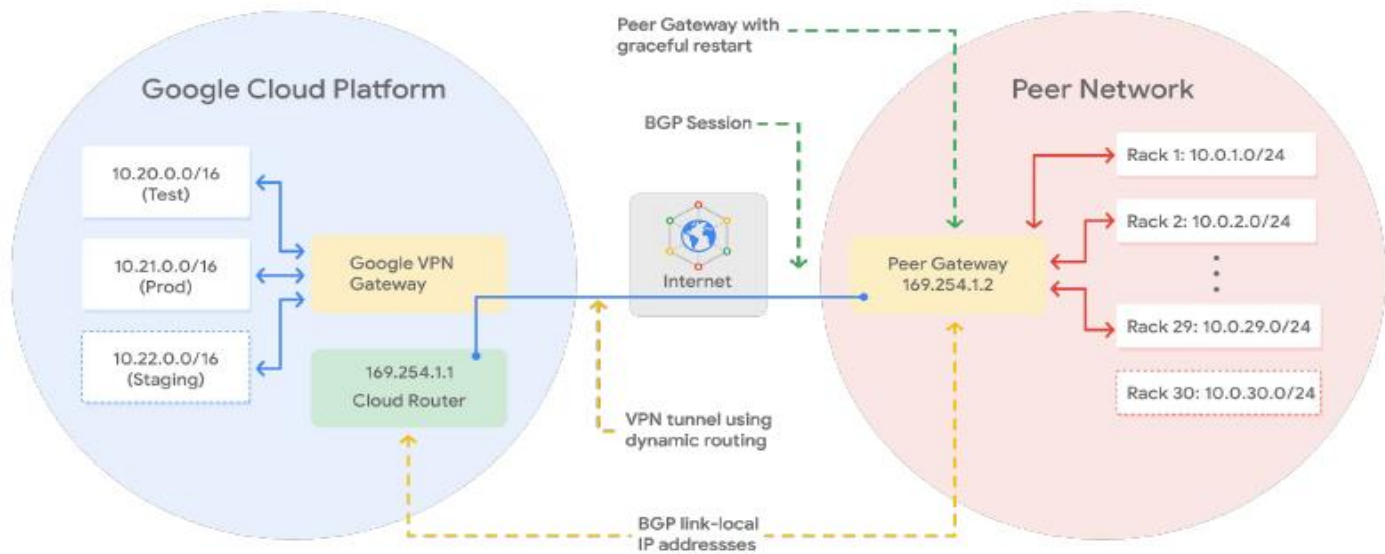
## VPN topology



- In order to connect to your on-premises network and its resources, you need to configure your Cloud VPN gateway, on-premises VPN gateway, and two VPN tunnels.
- The Cloud VPN gateway is a regional resource that uses a regional external IP address.
- Your on-premises VPN gateway can be a physical device in your data center or a physical or software-based VPN offering in another cloud provider's network. This VPN gateway also has an external IP address.
- A VPN tunnel then connects your VPN gateways and serves as the virtual medium through which encrypted traffic is passed.
- 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.
- *When using Cloud VPN is that the maximum transmission unit, or MTU, for your on-premises VPN gateway cannot be greater than 1460 bytes. This is because of the encryption and encapsulation of packets.*



# Dynamic routing with Cloud Router



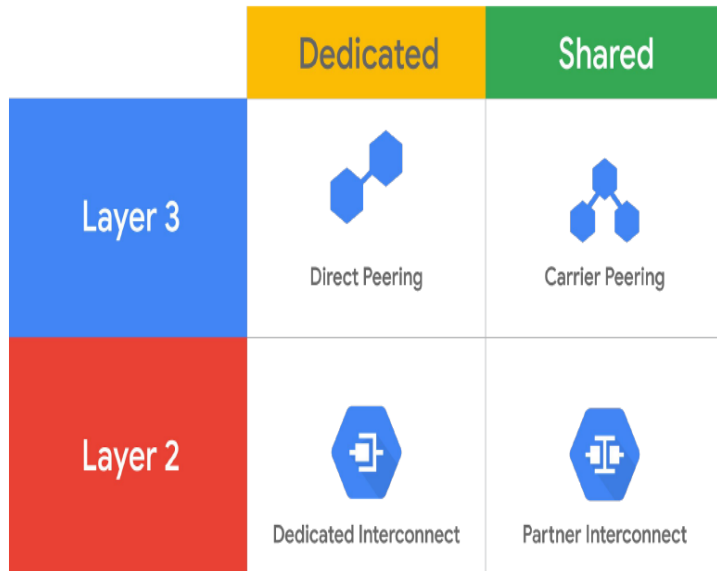
- Cloud VPN supports both static and dynamic routes. In order to use dynamic routes, you need to configure Cloud Routers.
- Cloud Router can manage routes for a Cloud VPN tunnel using Border Gateway Protocol, or BGP. This routing method allows for routes to be updated and exchanged without changing the tunnel configuration.

Above diagram shows two different regional subnets in a VPC network, namely Test and Prod. The on-premises network has 29 subnets, and the two networks are connected through Cloud VPN tunnels. Now, how would you handle adding new subnets? For example, how would you add a new “Staging” subnet in the GCP network and a new on-premises 10.0.30.0/24 subnet to handle growing traffic in your data center?

- 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.
- The new subnets are then seamlessly advertised between networks, means that instances in the new subnets can start sending and receiving traffic immediately.
- To set up BGP, an additional IP address has to be assigned to each end of the VPN tunnel.
- These two IP addresses must be link-local IP addresses, belonging to the IP address range 169.254.0.0/16.
- These addresses are not part of IP address space of either network and are used exclusively for establishing a BGP session.

## Cloud Interconnect and Peering services

- Cloud Interconnect and Peering services available to connect your infrastructure to Google's network.
- These services can be split into *Dedicated Vs Shared connections & Layer 2 Vs Layer 3 connections*.
- The services are *Direct Peering, Carrier Peering, Dedicated Interconnect, and Partner Interconnect*.

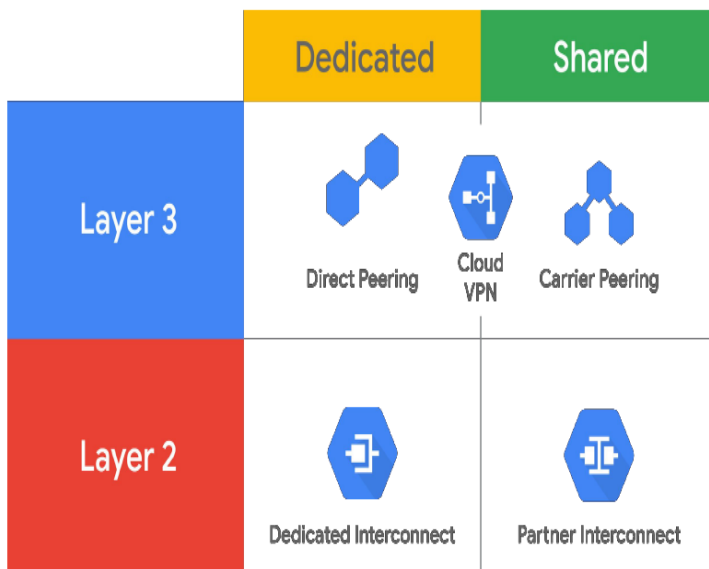


Dedicated connections provide a direct connection to Google's network.

Shared connections provide a connection to Google's network through a partner.

Layer 2 connections 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 provide access to G Suite services, YouTube, and Google Cloud APIs using public IP addresses.

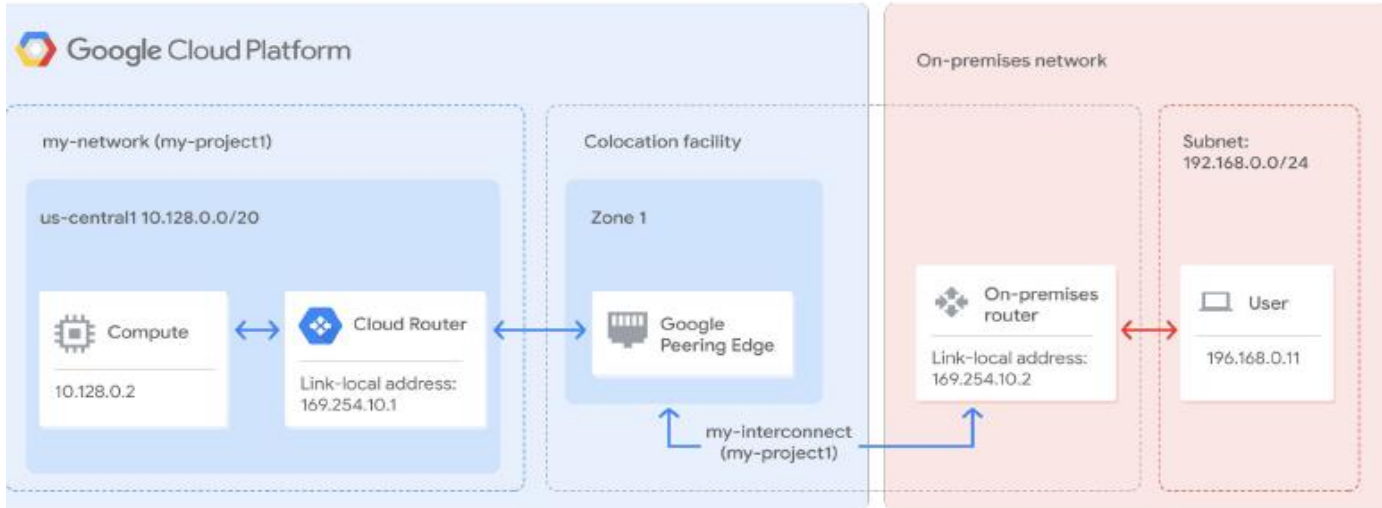


Google also offers its own Virtual Private Network service, called Cloud VPN.

Cloud VPN 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.

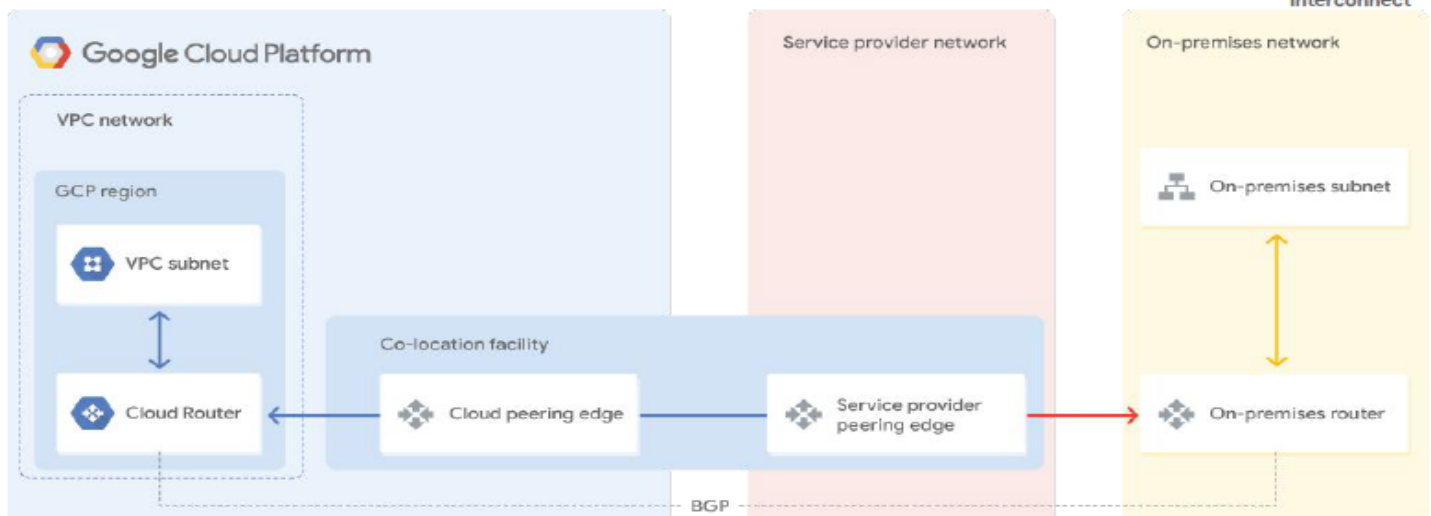
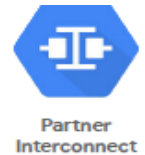
# Dedicated Interconnect provides direct physical connections



- **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.
- In order to use Dedicated Interconnect, you need to provision a cross connect between the Google network and your own router in a common colocation facility, as shown in this diagram.
- To exchange routes between the networks, you configure a BGP session over the interconnect between the Cloud Router and the on-premises router. This will allow user traffic from the on-premises network to reach GCP resources on the VPC network, and vice versa.
- *Dedicated Interconnect can be configured to offer a 99.9% or a 99.99% uptime SLA.*
- In order to use Dedicated Interconnect, your network must physically meet Google's network in a supported **colocation facility**. (like Mumbai, Chennai in India)



# Partner Interconnect provides connectivity through a supported service provider



- **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.
- In order to use Partner Interconnect, you work with a supported service provider to connect your VPC and on-premises networks.
- On-premises networks / service providers have existing physical connections to Google's network that they make available for their customers to use. Once you establish connectivity with a service provider, you can request a Partner Interconnect connection from your service provider.
- Then, you establish a BGP session between your Cloud Router and on-premises router to start passing traffic between your networks via the service provider's network.
- *Partner Interconnect can be configured to offer a 99.9% or a 99.99% uptime SLA between Google and the service provider.*

## Comparison of Interconnect options

Connection	Provides	Capacity	Requirements	Access Type
IPsec VPN tunnel	Encrypted tunnel to VPC networks through the public internet	1.5-3 Gbps per tunnel	On-premises VPN gateway	Internal IP addresses
Dedicated Interconnect	Dedicated, direct connection to VPC networks	10 Gbps per link 100 Gbps <sup>BETA</sup>	Connection in colocation facility	
Partner Interconnect	Dedicated bandwidth, connection to VPC network through a service provider	50 Mbps – 10 Gbps per connection	Service provider	

All these options provide internal IP address access between resources in your on-premises network and in your VPC network. The main differences are the connection capacity and the requirements for using a service.

- The **IPsec VPN tunnels** that Cloud VPN offers have a capacity of 1.5 to 3 Gbps per tunnel and require a VPN device on your on-premises network. The 1.5-Gbps capacity applies to traffic that traverses the public internet, and the 3-Gbps capacity applies to traffic that is traversing a direct peering link.
- **Dedicated Interconnect** has a capacity of 10 Gbps per link and requires you to have a connection in a Google-supported colocation facility. You can have up to 8 links to achieve multiples of 10 Gbps, but 10 Gbps is the minimum capacity.
- **Partner Interconnect** has a capacity of 50 Mbps to 10 Gbps per connection, and requirements depend on the service provider.

**Direct Peering** provides a direct connection between your business network and Google's network.

- With direct peering you will be able to exchange internet traffic between your network and Google's at one of Google's broad-reaching edge network locations.
- With this connection you will be able to exchange internet traffic between your network and Google's at one of Google's broad-reaching edge network locations.
- 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.*
- *PoPs are present on over 90 internet exchanges and at over 100 interconnection facilities around the world.*
- ***If you are not near to any of PoPs, then you will have to consider Carrier Peering.***

**Carrier Peering** provides connectivity through a supported partner.

- If you require access to Google public infrastructure and cannot satisfy Google's peering requirements, you can connect via a Carrier Peering partner. Work directly with your service provider to get the connection you need and to understand the partner's requirements.
- Carrier Peering does not have an SLA.






## Comparison of Peering options

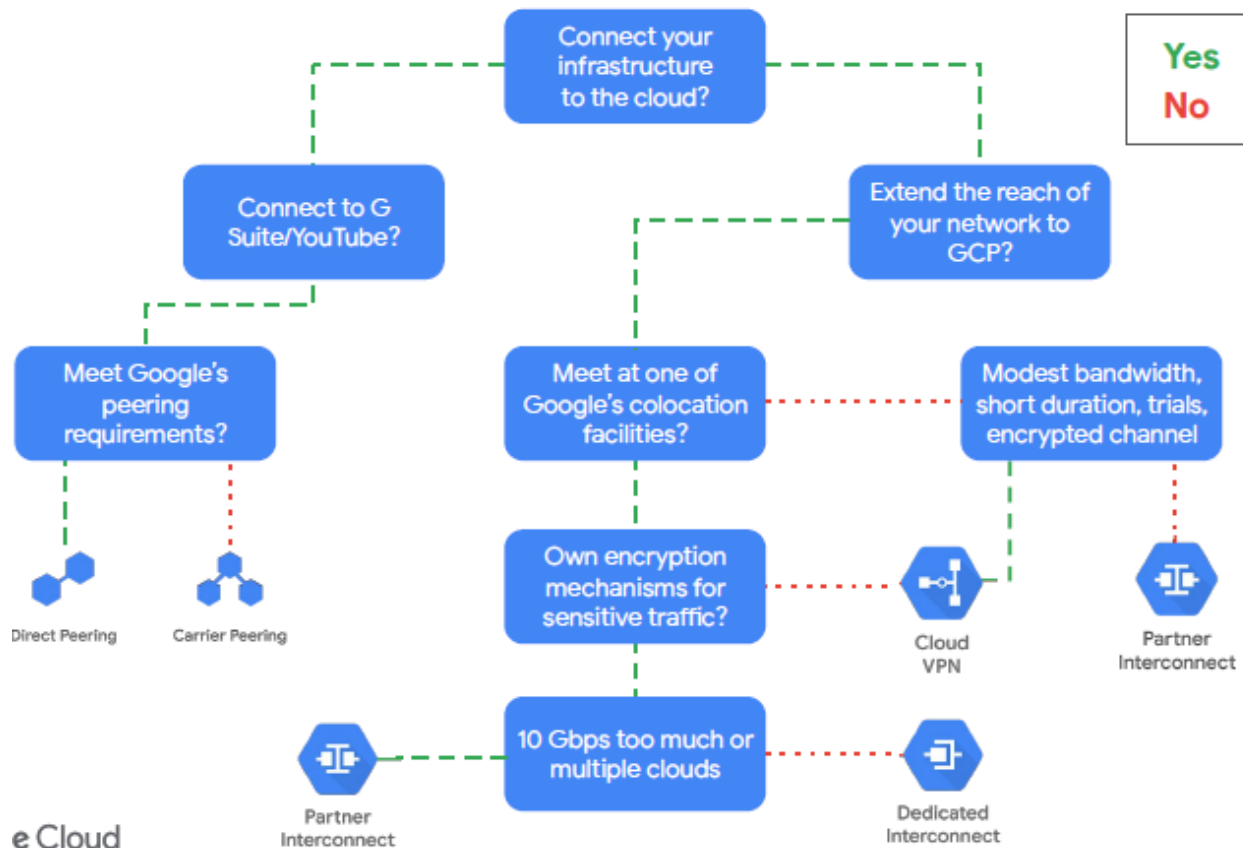
Connection	Provides	Capacity	Requirements	Access Type
Direct Peering	Dedicated, direct connection to Google's network	10 Gbps Per link	Connection in GCP PoPs	Public IP addresses
Carrier Peering	Peering through service provider to Google's public network	Varies based on partner offering	Service provider	

- All of these options provide public IP address access to all of Google's services. The main differences are capacity and the requirements for using a service.
- Direct Peering has a capacity of 10 Gbps per link and requires you to have a connection in a GCP Edge Point of Presence.
- Carrier Peering's capacity and requirements vary depending on the service provider that you work with.

## Choosing a network connection option

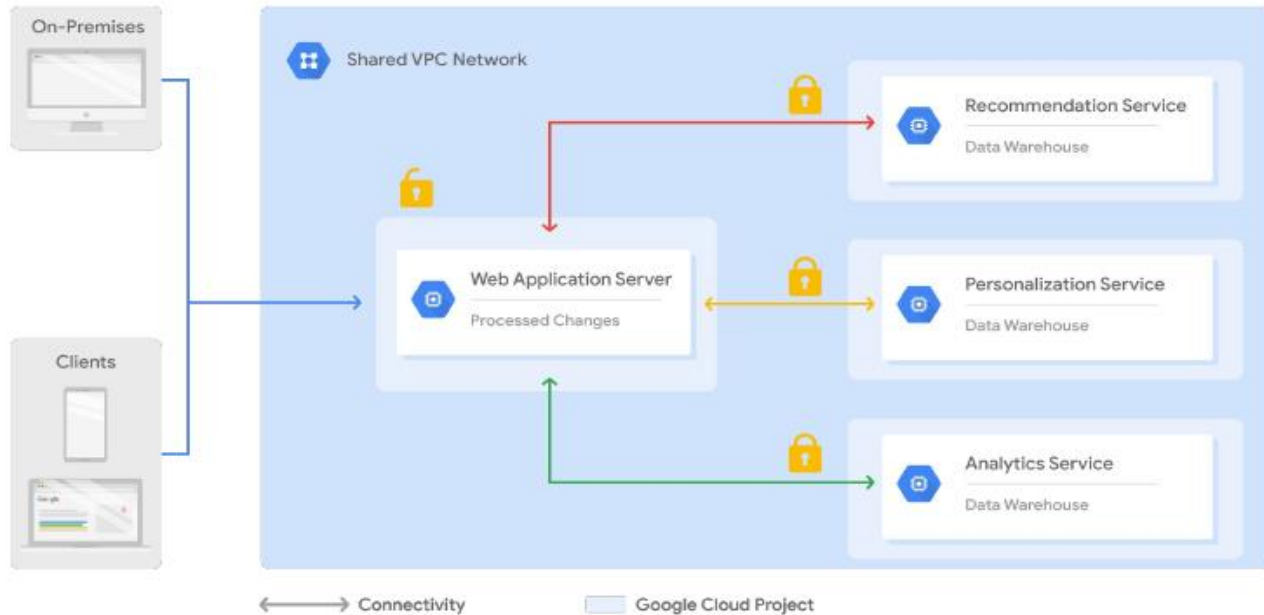
- Interconnect services () provide direct access to RFC1918 IP addresses in your VPC, with an SLA.
- Peering services, in contrast, offer access to Google public IP addresses only, without an SLA.

Interconnect	Peering
Direct access to RFC1918 IPs in your VPC – with SLA	Access to Google public IPs only – without SLA
 Dedicated Interconnect	 Direct Peering
 Partner Interconnect	 Carrier Peering
 Cloud VPN	



## Shared VPC

- Shared VPC allows an organization to connect resources from multiple projects to a common VPC network.
- Shared VPC allows the resources to communicate with each other securely and efficiently using internal IPs from that network.



- When you use shared VPC, you designate a project as a host project and attach one or more other service projects to it. In this case, the Web Application Server's project is the host project, and the three other projects are the service projects. The overall VPC network is called the shared VPC network.
- The Web Application Server communicates with clients and on-premises using the server's external IP address. The backend services, in contrast, cannot be reached externally because they only communicate using internal IP addresses.

**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.

- VPC Network Peering is a decentralized or distributed approach to multi-project networking, because each VPC network may remain under the control of separate administrator groups and maintains its own global firewall and routing tables.
- Historically, such projects would consider external IP addresses or VPNs to facilitate private communication between VPC networks. VPC
- Network Peering does not incur the network latency, security, and cost drawbacks that are present when using external IP addresses or VPNs.

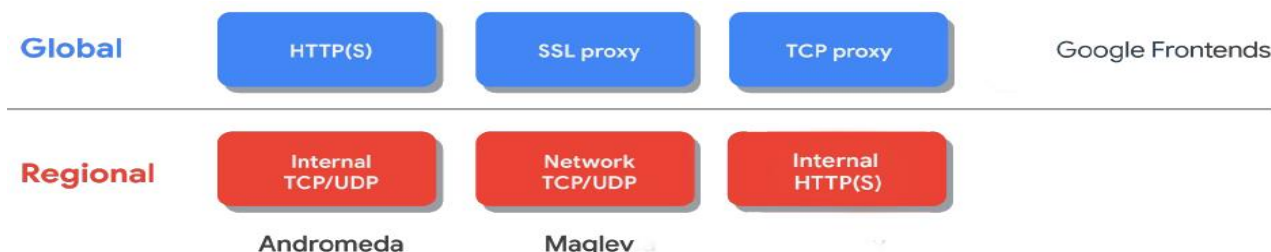
## Shared VPC vs. VPC peering

Consideration	Shared VPC	VPC Network Peering
Across organizations	No	Yes
Within project	No	Yes
Network administration	Centralized	Decentralized

- VPC Network Peering : private communication between VPC networks in different organizations or within project.
- Shared VPC : private communication between same organization or across projects

## Load Balancing and Autoscaling

- Cloud Load Balancing gives you the ability to distribute load-balanced compute resources in single or multiple regions to meet your high availability requirements.
- Cloud Load Balancing put your resources behind a single anycast IP address, and to scale your resources up or down with intelligent autoscaling.
- Cloud Load Balancing will serve content as close as possible to your users on a system that can respond to over 1 million queries per second.
- Cloud Load Balancing is a fully distributed, software-defined, managed service.
- Cloud Load Balancing is not instance or device-based, so you do not need to manage a physical load balancing infrastructure.



- Cloud Load Balancing reacts quickly to changes in users, traffic, network, backend health, and other related conditions.
- You can put Cloud Load Balancing in front of all of your traffic: HTTP(S), other TCP and SSL traffic, and UDP traffic too.
- It provides cross-region load balancing, including automatic multi-region failover, which gently moves traffic infractions if backends become unhealthy.

### Google VPC offers a suite of load-balancing options

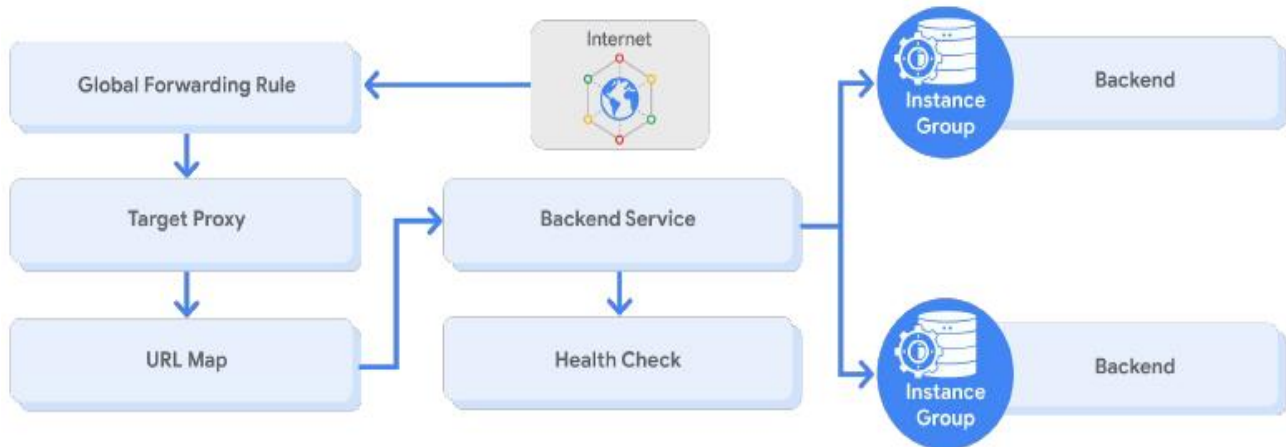
Global HTTP(S)	Global SSL Proxy	Global TCP Proxy	Regional	Regional internal
Layer 7 load balancing based on load	Layer 4 load balancing of non-HTTPS SSL traffic based on load	Layer 4 load balancing of non-SSL TCP traffic	Load balancing of any traffic (TCP, UDP)	Load balancing of traffic inside a VPC
Can route different URLs to different back ends	Supported on specific port numbers	Supported on specific port numbers	Supported on any port number	Use for the internal tiers of multi-tier applications

- GCP offers different types of load balancers that can be divided into two categories:
  - Global ( HTTP(S), SSL proxy, and TCP proxy )
  - Regional ( Internal TCP/UDP, Network TCP/UDP, Internal HTTP(S) )
- For cross-regional load balancing for a Web application, use HTTP(S) load balancing
- For other TCP traffic that does not use Secure Sockets Layer, use the Global TCP Proxy load balance
- For load balance UDP traffic, or traffic on any port number, you can still load balance across a GCP region with the Regional load.
- For 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.

## HTTP(S) Load Balancing:

- HTTP(S) load balancing provides global load balancing for HTTP(S) requests destined for your instances, means your applications are available to your customers at a single anycast IP address, which simplifies your DNS setup.
- HTTP requests are load balanced on port 80 or 8080
- HTTPS requests are load balanced on port 443.
- HTTPS load balancer supports both IPv4 and IPv6 clients.
- HTTPS load balancer can configure URL maps that route some URLs to one set of instances and route other URLs to other instances.
- Requests are generally routed to the instance group that is closest to the user.

### Architecture of an HTTP(S) load balancer



Backend Services can be one of following :

- Health check
- Session affinity (optional)
- Time out setting (30-sec default)
- One or more backends
- An instance group (managed or unmanaged)
- A balancing mode (CPU utilization or RPS)
- A capacity scaler (ceiling percentage of CPU/Rate targets)

**Content-based load balancing :** HTTP load balancer is a content-based load balancer.

The traffic is split by the load balancer based on the URL header as specified in the URL map.

**HTTP(S) load balancer** has the same basic structure as an HTTP load balancer, but differs in the following ways:

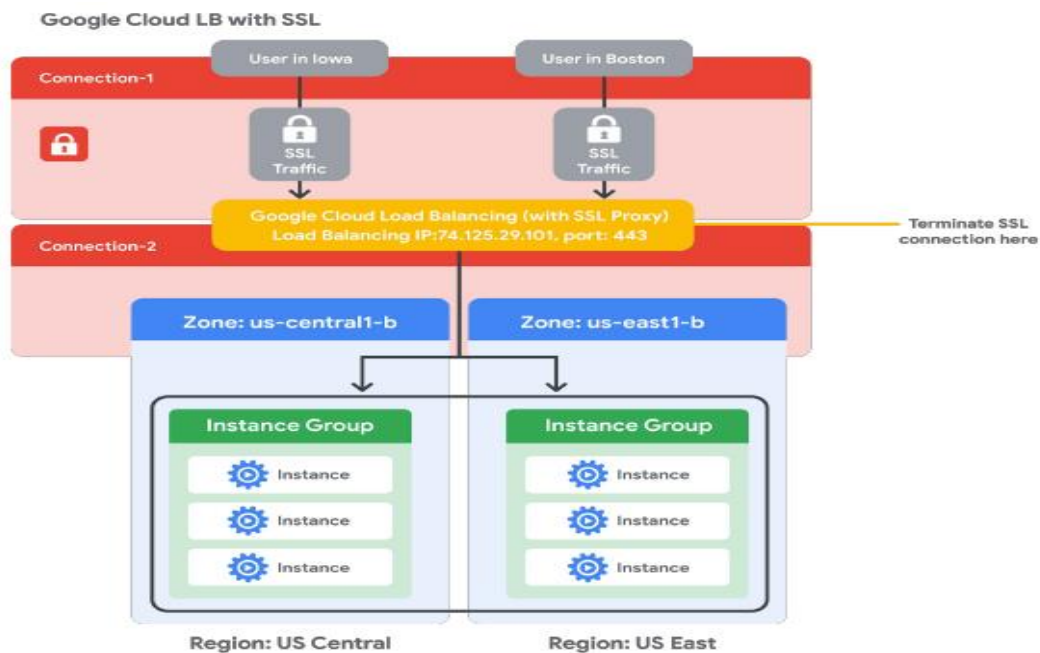
- An HTTP(S) load balancer uses a target HTTPS proxy instead of a target HTTP proxy.
- An HTTP(S) load balancer requires at least one signed SSL certificate installed on the target HTTPS proxy for the load balancer.
- The client SSL session terminates at the load balancer.
- HTTP(S) load balancers support 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.

### SSL certificates

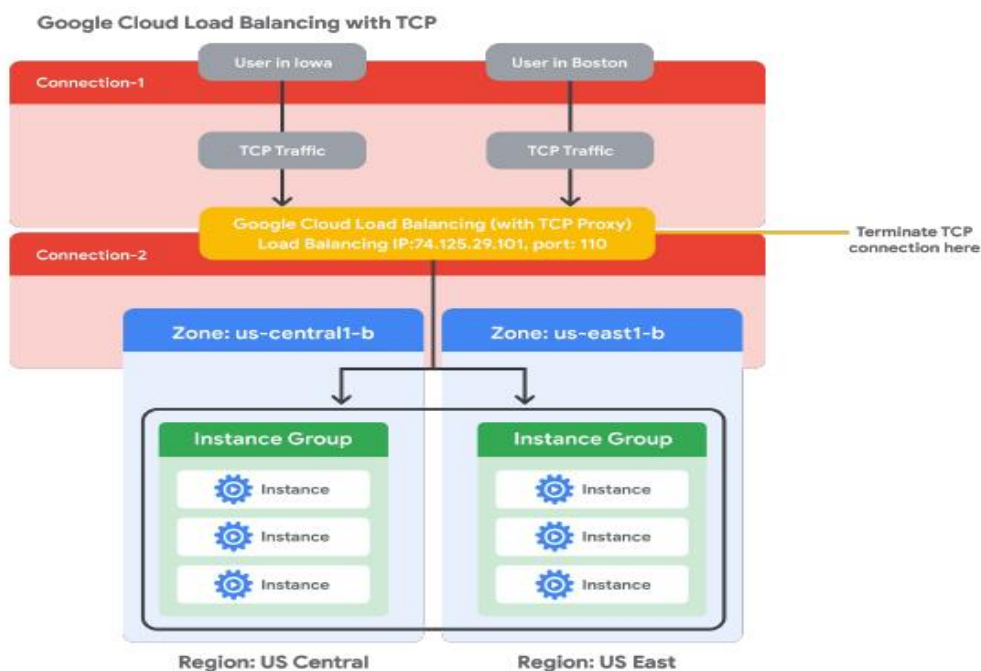
- Required for HTTP(S) load balancing
- To use HTTPS, you must create at least one SSL certificate that can be used by the target proxy for the load balancer.
- Up to 10 SSL certificates (per target proxy)
- You can configure the target proxy with up to ten SSL certificates.
- Create an SSL certificate resource
- each SSL certificate, you first create an SSL certificate resource, which contains the SSL certificate information. SSL certificate resources are used only with load balancing proxies such as a target HTTPS proxy or target SSL proxy

## SSL Proxy & TCP Proxy Load Balancing

- SSL / TCP proxy is a global load balancing service for *encrypted, non-HTTP* traffic.
- SSL / TCP proxy load balancer *terminates user SSL / TCP 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 / TCP 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 balancer 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.
  - If vulnerabilities arise in the SSL / TCP stack, GCP will apply patches at the load balancer automatically in order to keep your instances safe.



SSL Proxy Load Balancing



TCP Proxy Load Balancing



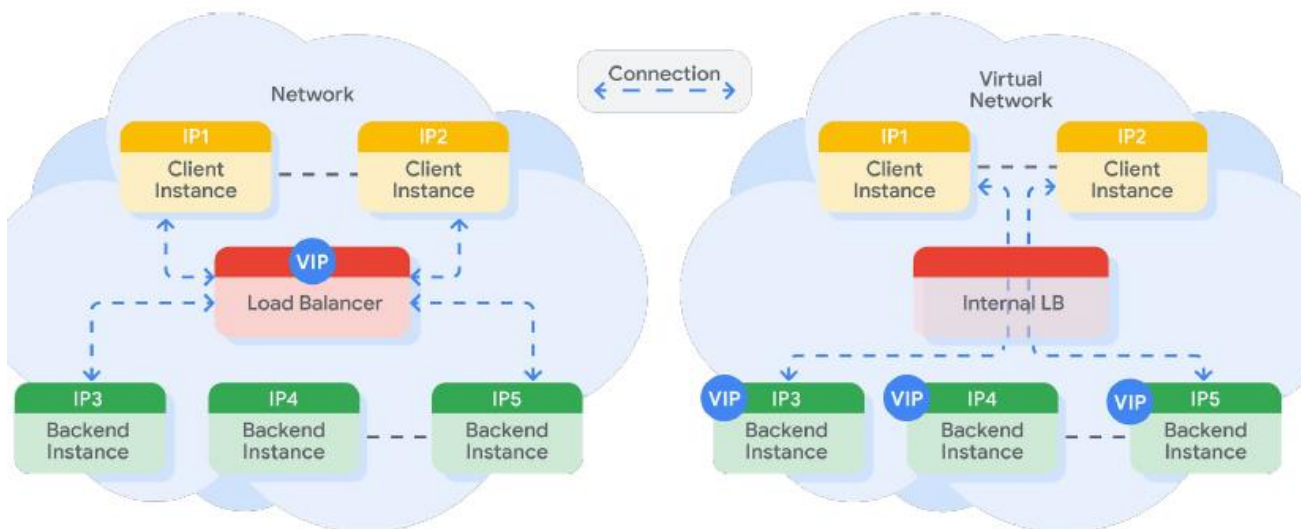
## Network load balancing

- Network load balancing is a regional, non-proxied load balancing service.
- 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.
- 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.
- Network load balancing used to load balance UDP traffic.
- Network load balancing used to load balance TCP and SSL traffic on ports that are not supported by the TCP proxy and SSL proxy load balancers.
- The backends of a network load balancer can be a **template-based instance group** or **target pool resource group**.
- **Target pool resource** defines a group of instances that receive incoming traffic from forwarding rules ((TCP and UDP)).
  - each project can have up to 50 target pools
  - all the instances of a target pool must be in the same region

## Internal Load Balancing

- Internal load balancing is a regional, private load balancing service for TCP- and UDP-based traffic.
- Internal load balancer enables you to run and scale your services behind a private load balancing IP address, means that it is only accessible through the internal IP addresses of virtual machine instances that are in the same region.
- TO use internal load balancing, 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.

## Software-defined, fully distributed load balancing



1. Proxy Internal Load Balancing

2. Google Cloud Internal Load Balancing

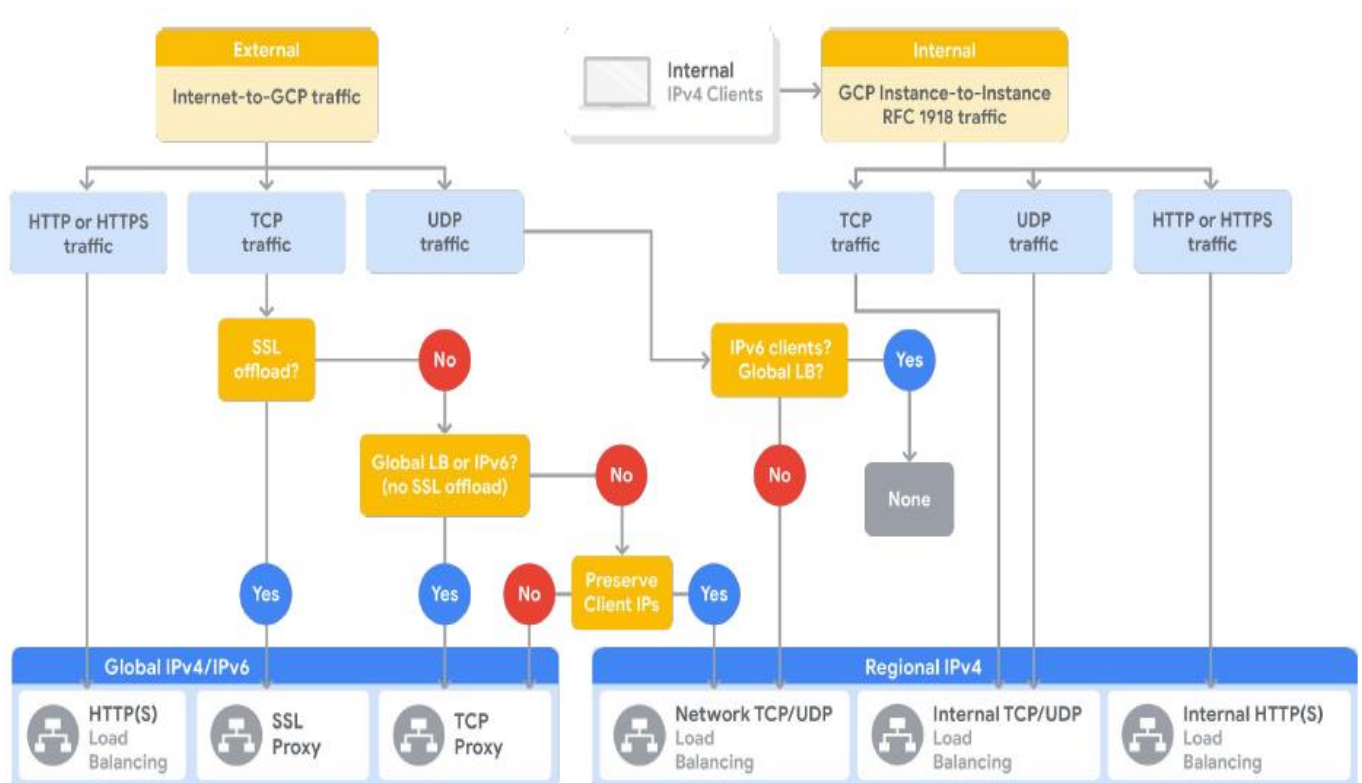
- GCP internal load balancing is not based on a device or a VM instance. Instead, it is a software-defined, fully distributed load balancing solution.
- In the traditional proxy model of internal load balancing, as shown on the left, you configure an internal IP address on a load balancing device or instances, and your client instance connects to this IP address. Traffic coming to the IP address is terminated at the load balancer, and the load balancer selects a backend to establish a new connection to.
- Essentially, there are two connections: one between the Client and the Load Balancer, and one between the Load Balancer and the Backend.
- GCP internal load balancing distributes client instance requests to the backends using a different approach, as shown on the right. It uses lightweight load balancing built on top of **Andromeda** (Google's network virtualization stack) to provide software-defined
- load balancing that directly delivers the traffic from the client instance to a backend instance.



## Summary of load balancers

Load balancer	Traffic type	Global/ Regional	External/ Internal	External ports for load balancing
HTTP(S)	HTTP or HTTPS	Global IPv4 IPv6	External	HTTP on 80 or 8080; HTTPS on 443
SSL Proxy	TCP with SSL offload			25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
TCP Proxy	<ul style="list-style-type: none"> <li>TCP without SSL offload</li> <li>Does not preserve client IP addresses</li> </ul>			25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
Network TCP/UDP	<ul style="list-style-type: none"> <li>TCP/UDP without SSL offload</li> <li>Preserves client IP addresses</li> </ul>	Regional IPv4	Internal	Any
Internal TCP/UDP	TCP or UDP			Any
Internal HTTP(S)	HTTP or HTTPS			HTTP on 80 or 8080; HTTPS on 443

## Choosing Load Balancer



## Managed Instance Groups

- Managed instance group is a collection of identical VM instances that you control as a single entity, using an **instance template**.
- Managed instance groups 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.
- Managed instance groups can automatically identify and recreate unhealthy instances in a group to ensure that all the instances are running optimally.
- *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.
- This replication protects against zonal failures and unforeseen scenarios where an entire group of instances in a single zone malfunction. If that happens, your application can continue serving traffic from instances running in another zone of the same region.



Managed instance groups offer autoscaling capabilities

- Dynamically add/remove instances:
  - Increases in load
  - Decreases in load
- Autoscaling policy:
  - CPU utilization
  - Load balancing capacity
  - Monitoring metrics
  - Queue-based workload



Target CPU utilization = 75%

## Deployment Manager

- Deployment Manager is an infrastructure deployment service that automates the creation and management of GCP resources for you.
- Specify all the resources needed for your application in a declarative format and deploy your configuration.
- Repeatable deployment process : over and over with consistent results
- Declarative language : jinja files with YAML based syntax
- Focus on the application
- Parallel deployment
- Template-driven
- compute.v1.network , compute.v1.firewall
- Terraform, Chef, Puppet, Ansible, or Packer

## GCP Marketplace

- Deploy production-grade solutions
- Single bill for GCP and third-party services
- Manage solutions using Deployment Manager
- Notifications when a security update is available
- Direct access to partner support

## Managed Services

- Managed services are partial or complete solutions offered as a service.
- Managed service allows you to outsource a lot of the administrative and maintenance overhead to Google.



### BigQuery BigQuery

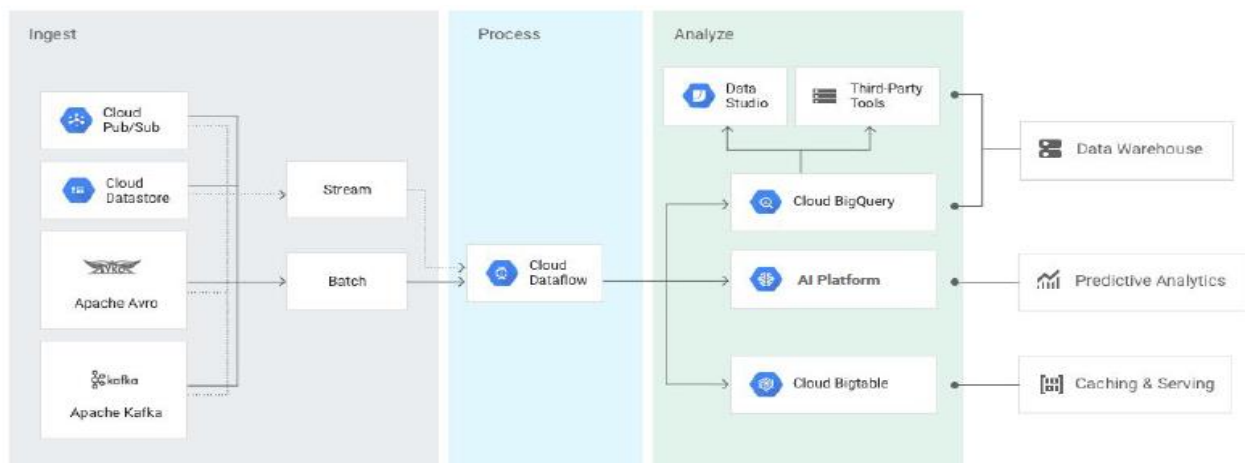
- BigQuery is GCP's serverless, highly scalable, and cost-effective cloud data warehouse
- Petabyte scale data warehouse
- SQL interface
- Super-fast queries (Query 100 billion rows in less than 1 minute)
- BigQuery REST API (Java, .Net, Python)
- Free usage tier
- Ex : 

```
SELECT language, SUM(views) as views
FROM (
  SELECT title, language, MAX(views) as views
  FROM [bigquery-samples:wikipedia_benchmark.Wiki100B]
  WHERE REGEXP_MATCH(title, "G.*o.*")
  GROUP EACH BY title, language
)
GROUP EACH BY language
ORDER BY views desc
```



### Cloud Dataflow Cloud DataFlow

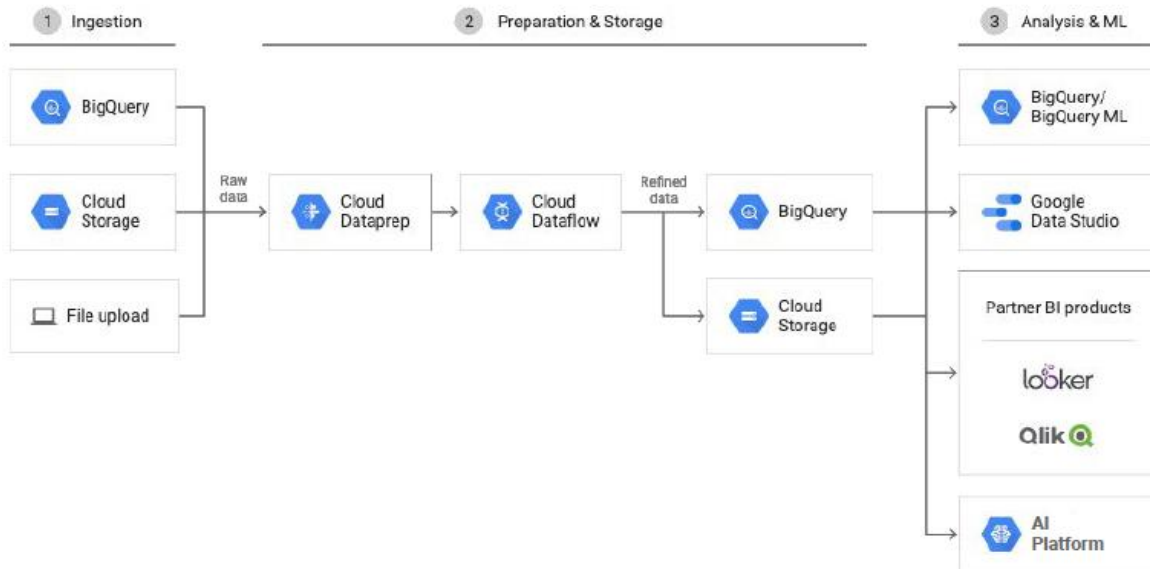
- Cloud Dataflow is a managed service for executing a wide variety of data processing patterns.
- Cloud Dataflow is a fully managed service for transforming and enriching data in stream and batch modes with equal reliability and expressiveness.
- Serverless, fully managed data processing
- Batch and stream processing with autoscaling
- Open source programming using SQL, Java, and Python APIs of APACHE BEAM
- Intelligently scale to millions of QPS
- Data transformation with Cloud Dataflow :





## Cloud DataPrep Cloud DataPrep

- Cloud DataPrep is an intelligent data service for visually exploring, cleaning, and preparing structured and unstructured data for analysis, reporting, and machine learning.
- Serverless, works at any scale
- Suggests ideal data transformation
- Focus on data analysis
- Integrated partner service operated by Trifacta
- Cloud DataPrep architecture :



## Cloud Dataproc Cloud DataProc

- Cloud Dataproc is a service for running Apache Spark and Apache Hadoop clusters.
- Pay for the resources you use with per-second billing
- leverage preemptible instances in your cluster, you can reduce your costs even further
- Cloud Dataproc clusters are quick to start, scale, and shut down, with each of these operations taking 90 seconds or less, on average.
- Built-in integration with other GCP services, such as BigQuery, Cloud Storage, Cloud Bigtable, Stackdriver Logging, and Stackdriver Monitoring.

### Cloud Dataflow vs. Cloud Dataproc

