Google Cloud

Partner Certification Academy

Professional Cloud Architect

The information in this presentation is classified:

Google confidential & proprietary

⚠️ This presentation is shared with you under NDA.

- Do **not** record or take screenshots of this presentation.
- Do **not** share or otherwise distribute the information in this presentation with anyone **inside** or **outside** of your organization.

Thank you!

Session logistics

- When you have a question, please:
  - Click the Raise hand button in Google Meet.
  - Or add your question to the Q&A section of Google Meet.
  - Please note that answers may be deferred until the end of the session.

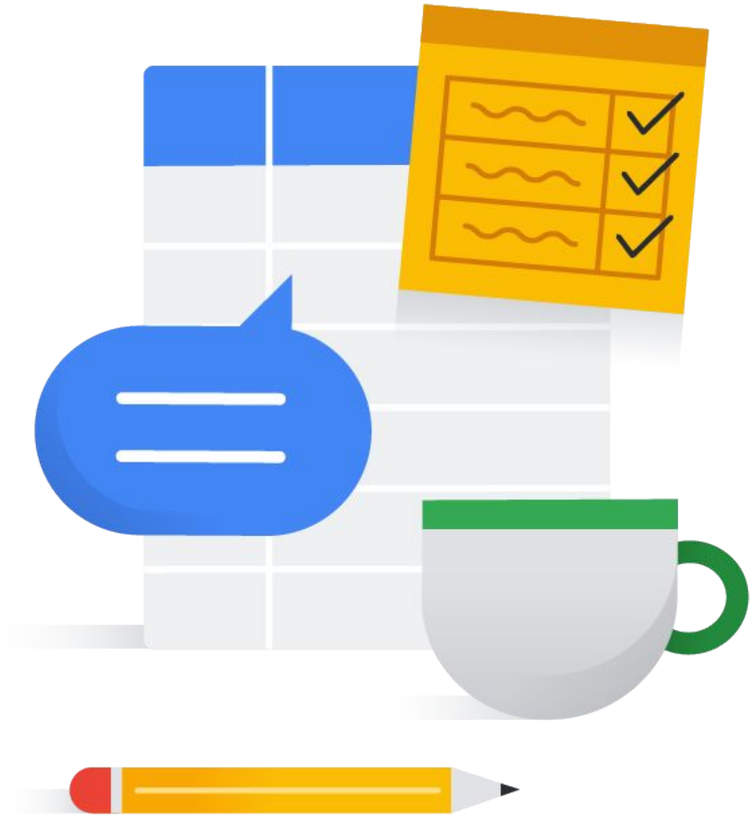
- These slides are available in the Student Lecture section of your Qwiklabs classroom.

- The session is **not recorded.**

- Google Meet does not have persistent chat.
  - If you get disconnected, you will lose the chat history.

Program issues or concerns?

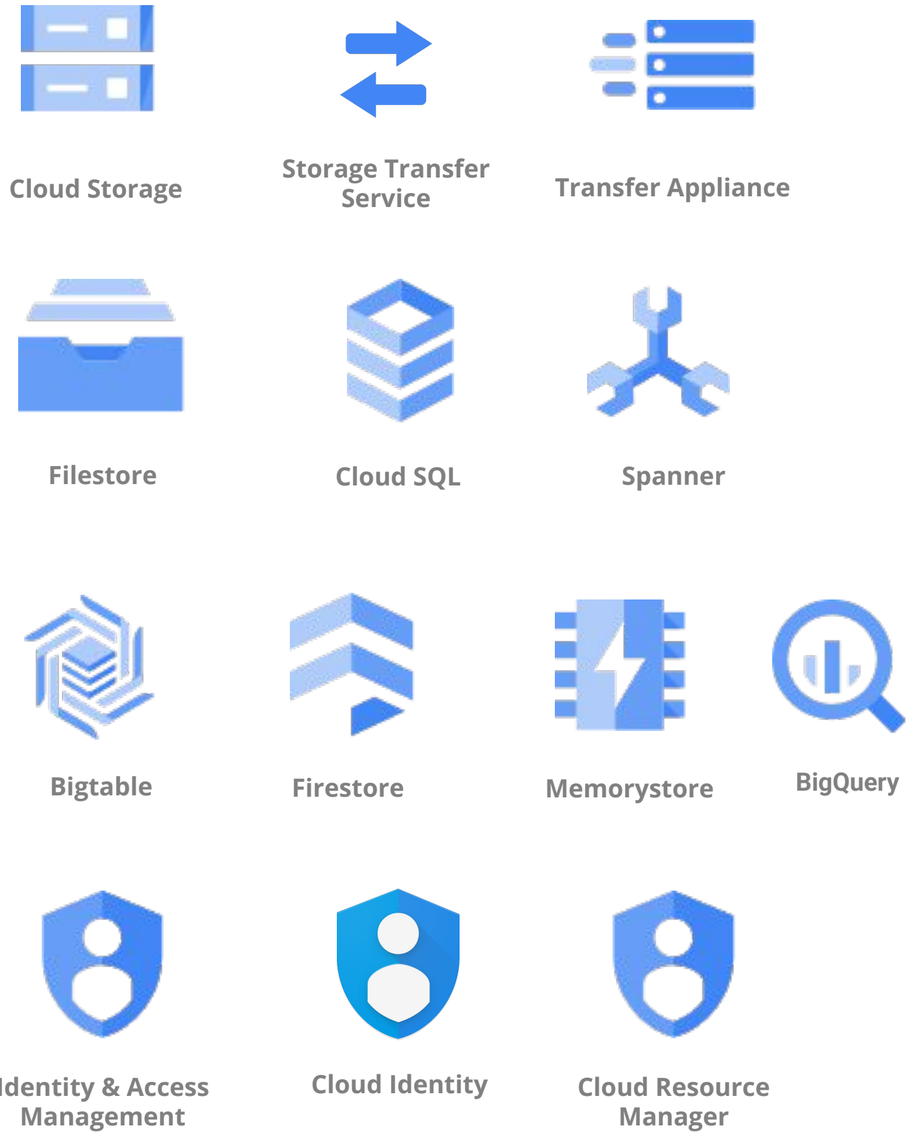
- Problems with Partner Skills Boost access, Qwiklabs, vouchers, etc.
  - cloud-partner-training@google.com



Key concepts in the on-demand content

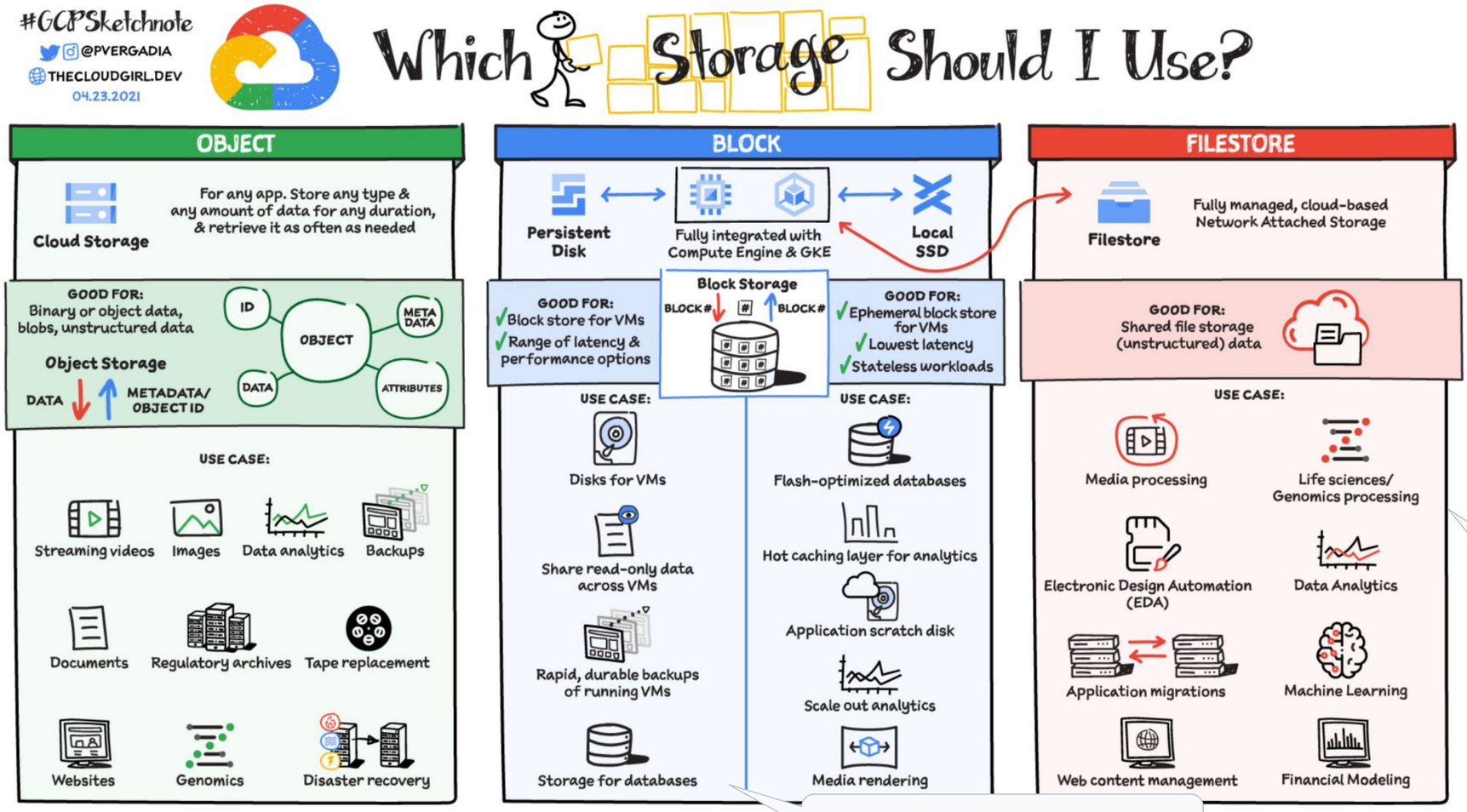
Topics in this module

- Storage options
  - Cloud Storage
    - Storage Transfer Options
  - Filestore
  - Cloud SQL
  - Spanner
  - Firestore
  - Bigtable
  - BigQuery
  - Memorystore
- Identity and Access Management (IAM)



Storage options in Google Cloud

Storage Overview



Covered in Module 1

Covered in Module 1

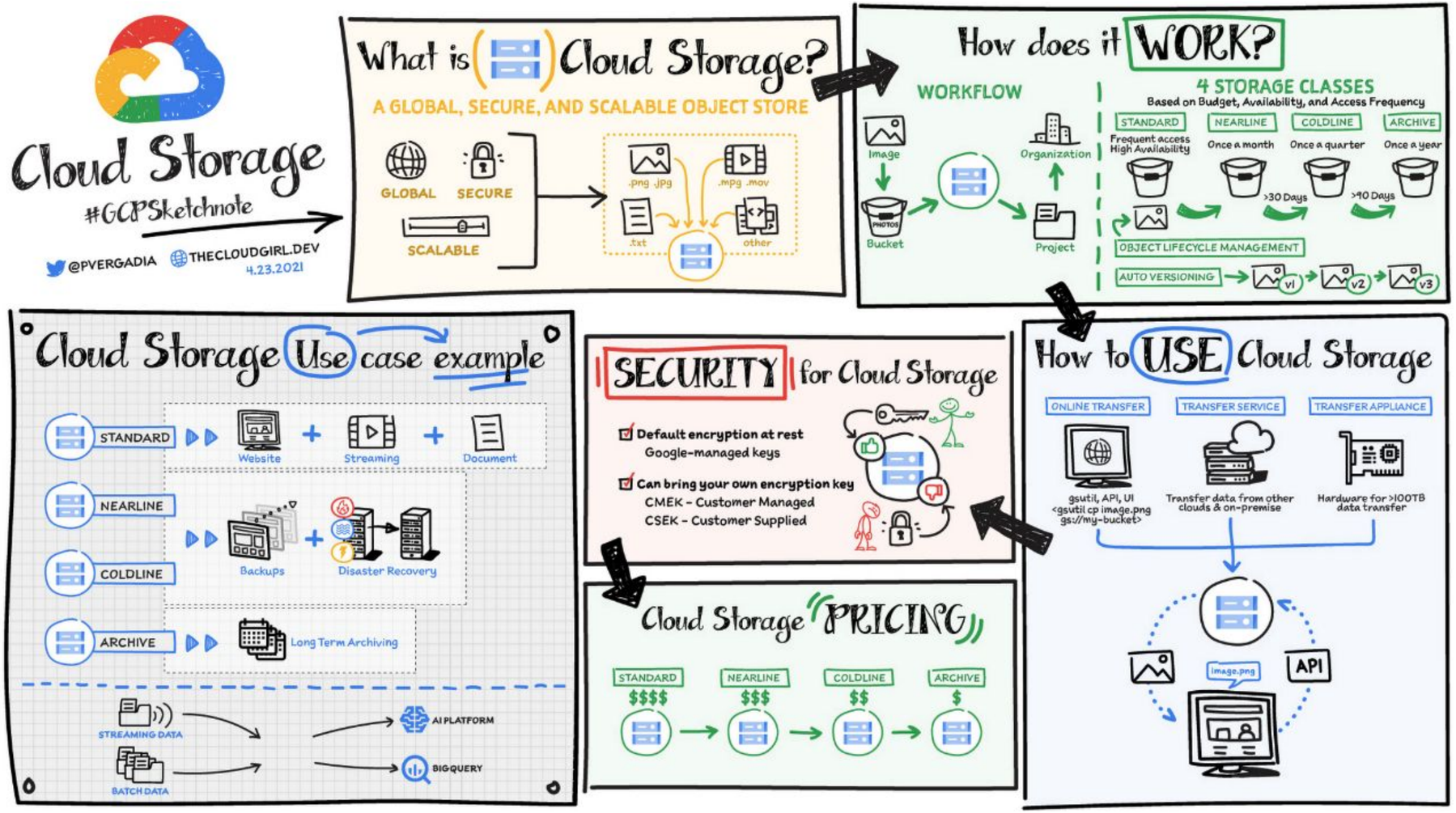
https://cloud.google.com/blog/topics/developers-practitioners/map-storage-options-google-cloud

Storage and database services

Object	File	Relational			Non-relational		Warehouse
Cloud Storage	Filestore	Cloud SQL	Spanner	AlloyDB	Firestore	Bigtable	BigQuery
Good for: Binary or object data	**Good for:** Network Attached Storage (NAS)	**Good for:** Web frameworks	**Good for:** RDBMS + scale, HA, HTAP	**Good for:** Legacy Applications	**Good for:** Hierarchical, mobile, web	**Good for:** Heavy read + write, events,	**Good for:** Enterprise data warehouse
Such as: Images, media serving, backups	**Such as:** Latency sensitive workloads	**Such as:** CMS, ecommerce	**Such as:** User metadata, Ad/Fin/MarTech	**Such as:** Transaction Processing, Enterprise workloads	**Such as:** User profiles, game state	**Such as:** Ad tech, financial, IoT	**Such as:** Analytics, dashboards

Next discussion

All you need to know about Cloud Storage

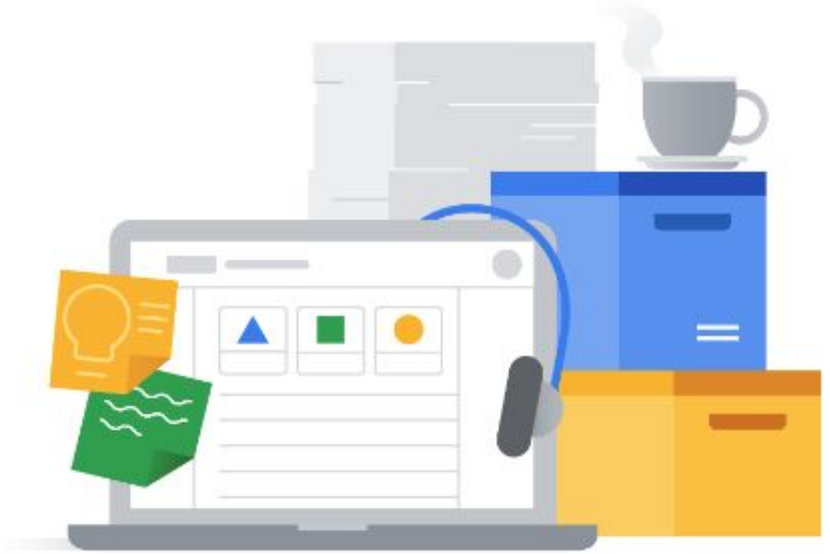


https://cloud.google.com/blog/topics/developers-practitioners/all-you-need-know-about-cloud-storage

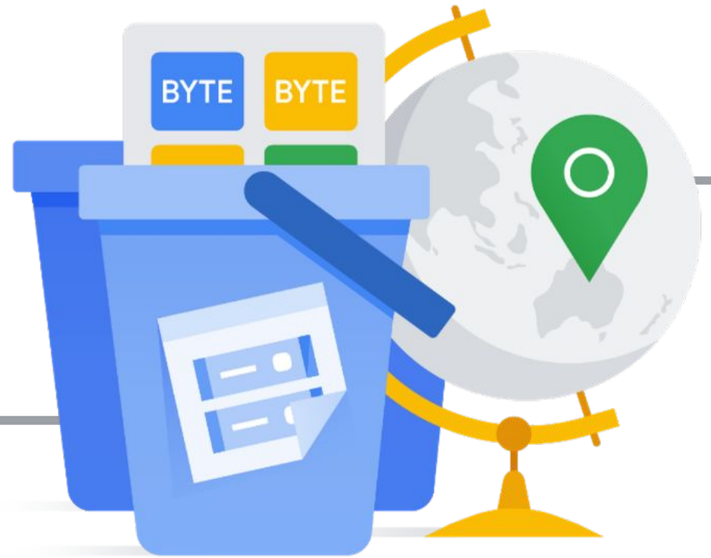
Cloud Storage is a fully managed storage service

Binary large-object (BLOB) storage used for

- Online content
- Backup and archiving
- Storage of intermediate results
- And much more....



Files are organized into buckets



Geographic location

Unique name

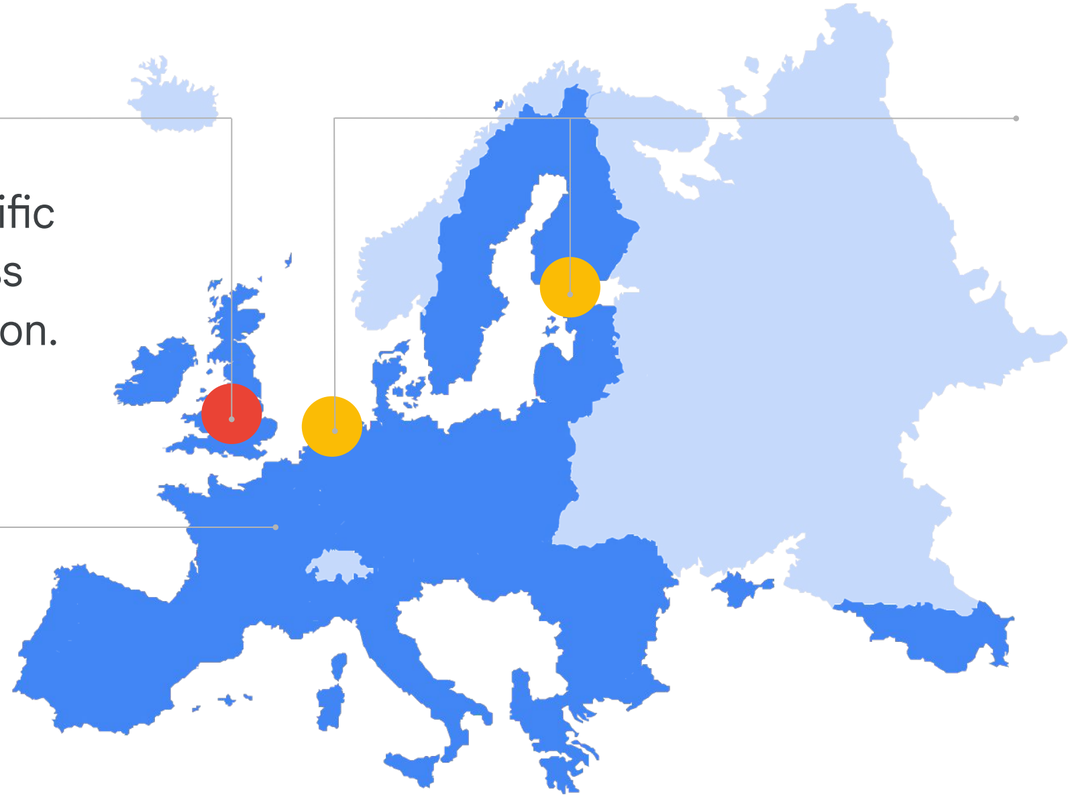
Choosing a location type

Regional

Your data is stored in a specific region with replication across availability zones in that region.

Multi-regional

Your data is distributed redundantly across US, EU or Asia.



Dual-regional

Your data is replicated across a specific pair of regions.

[How to choose between regional, dual-region and multi-region Cloud Storage]

Cloud Storage Classes - Options for any use case

Standard

In multi-region locations for serving content globally.

- Streaming videos
- Images
- Websites
- Documents

In regional or dual-regional locations for data accessed frequently or high throughput needs

- Video transcoding
- Genomics
- General data analytics & compute

Nearline

For data access less than once a month

- Serving rarely accessed docs
- Backup

Coldline

For data accessed roughly less than once a quarter

- Serve rarely used data
- Movie archive
- Disaster recovery

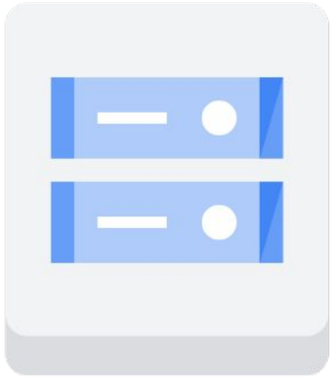
Archive

For long term retention

- Regulatory archives
- Tape replacement

Characteristics applicable to all storage classes

- Archive Storage
- Coldline Storage
- Nearline Storage
- Standard Storage

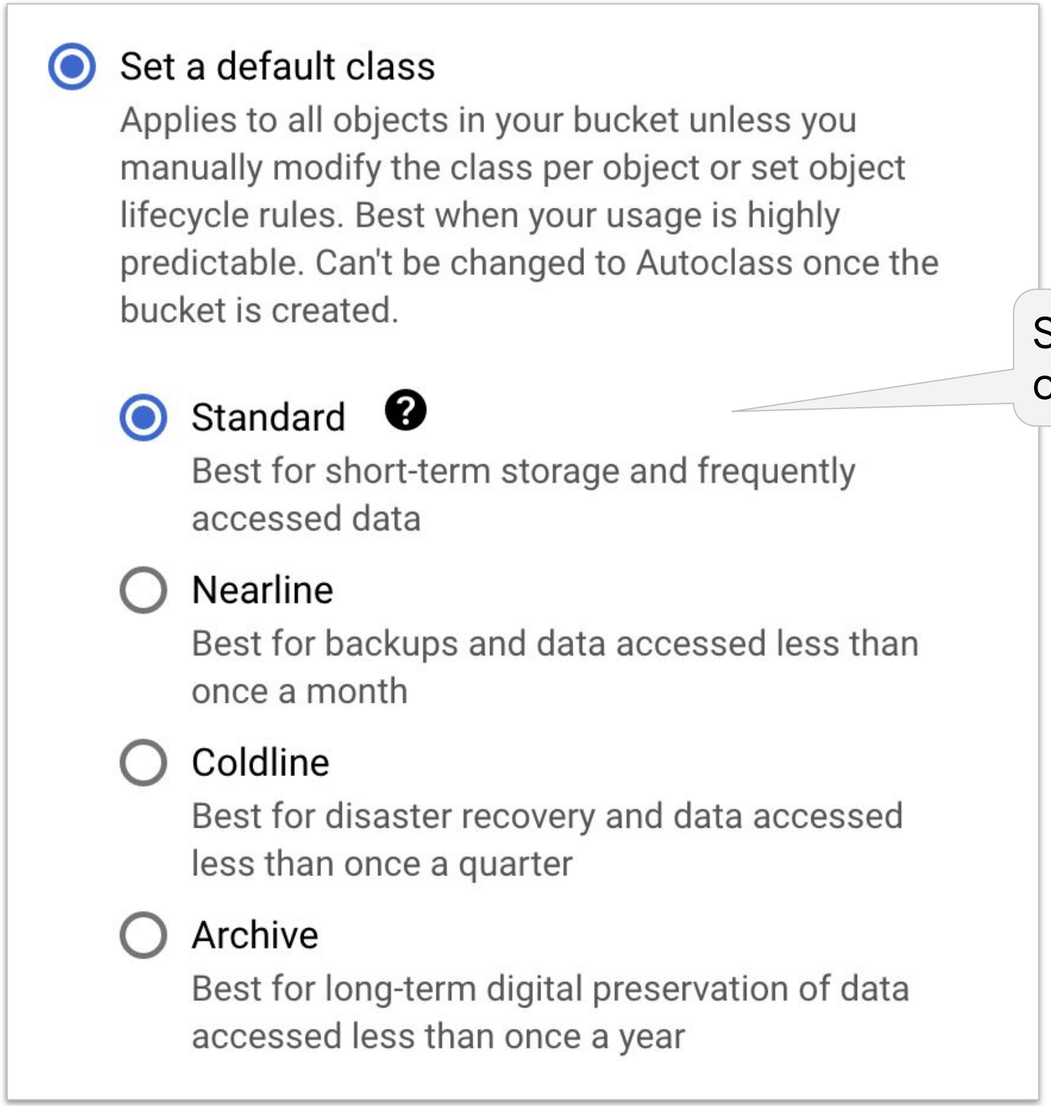


- Unlimited storage (no min object size)
- Worldwide accessibility and locations
- Low latency and high durability
- A uniform experience
- Regional/Geo-redundancy

Additional Cloud Storage features

- Pay only for what you use
- No prior provisioning of capacity
- Encrypts data on the server side
- Use HTTPS/TLS (Transport Layer Security)

Choosing a Cloud Storage class



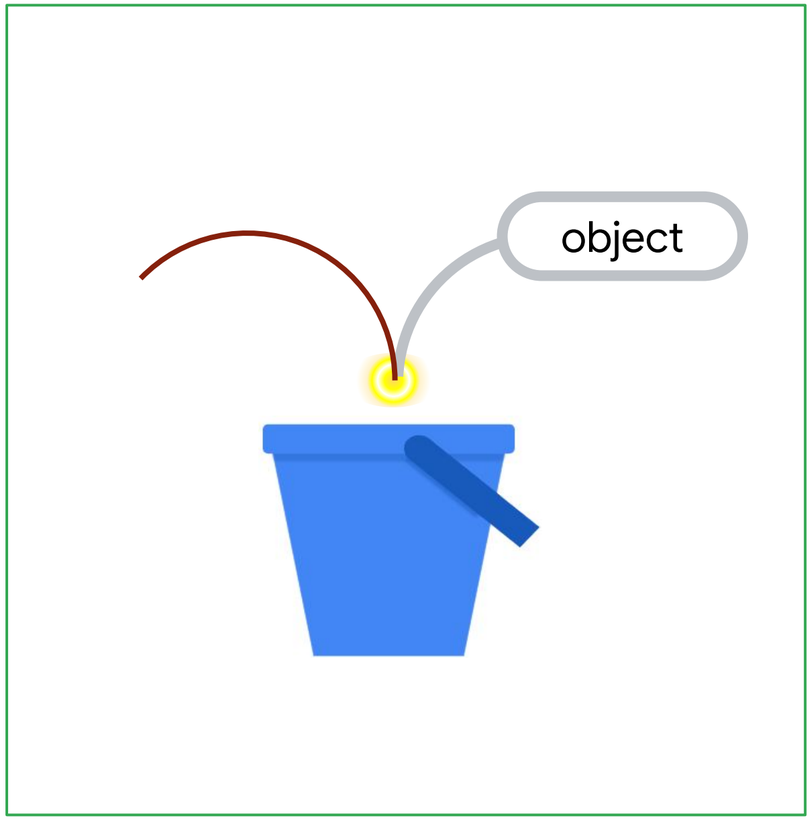
Select the storage class based how often you will be accessing the data

Choosing a storage classes

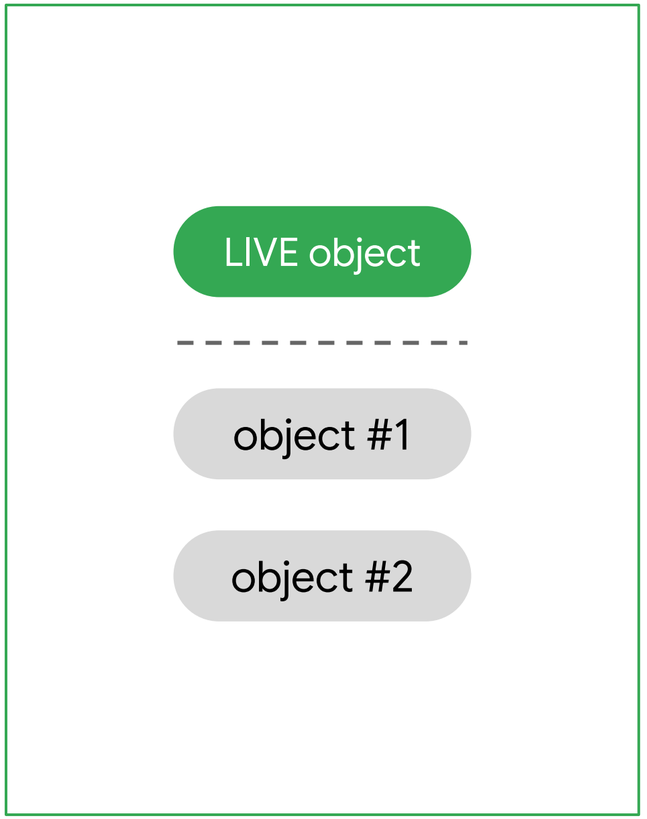
	Standard	Nearline	Coldline	Archive
Use case	“Hot” data and/or stored for only brief periods of time like data-intensive computations	Infrequently accessed data like data backup, long-tail multimedia content, and data archiving	Infrequently accessed data that you read or modify at most once a quarter	Data archiving, online backup, and disaster recovery
Minimum storage duration*	None	30 days	90 days	365 days
Retrieval cost	None	$0.01 per GB	$0.02 per GB	$0.05 per GB
Availability SLA	99.95% (multi/dual) 99.90% (region)	99.90% (multi/dual) 99.00% (region)		None
Durability	99.999999999%			

*Minimum storage duration = if delete file before x days, will still pay for x days

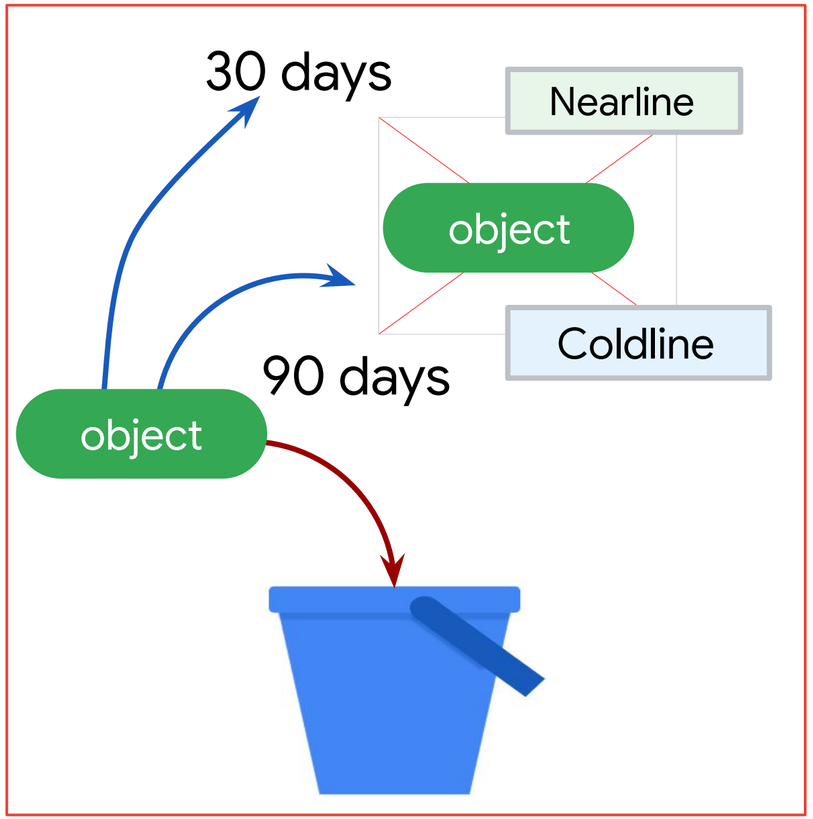
Cloud Storage has many object management features



Retention Policy



Versioning



Lifecycle Management

Options for controlling data lifecycles

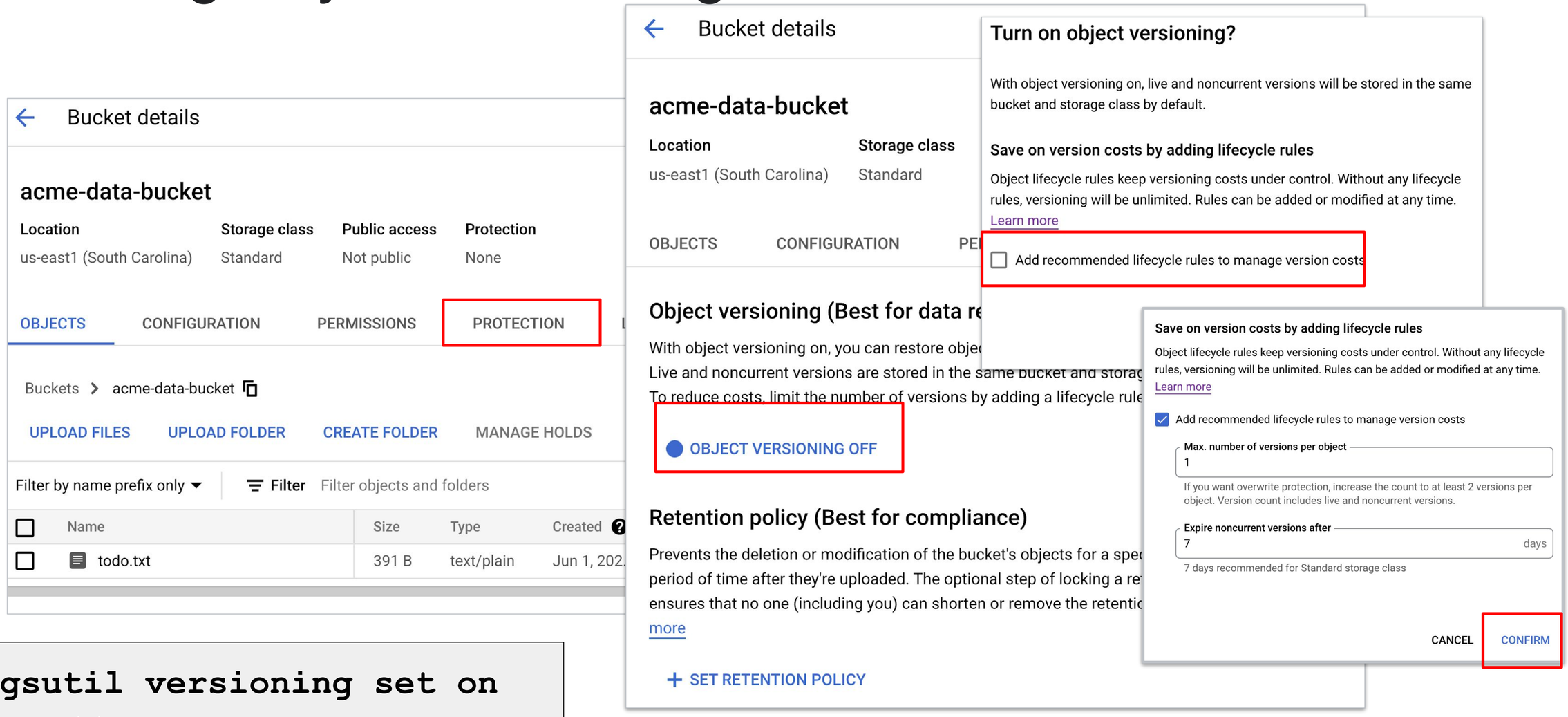
- A **retention policy** specifies a retention period to be placed on a bucket.
  - An object cannot be deleted or replaced until it reaches the specified age.
- **Object Versioning** can be enabled on a bucket in order to retain older versions of objects.
  - When the live version of an object is deleted or replaced, it becomes noncurrent
  - If a live object version is accidentally deleted, can restore the noncurrent version back to the live version.
  - Object Versioning increases storage costs, but can be mitigated by Lifecycle Management to delete older objects
- **Object Lifecycle Management** can be configured for a bucket, which provides automated control over deleting objects and changing storage classes

Consider retention policies and retention policy locks

- Add a retention policy to a bucket to specify a retention period.
  - If no policy exists, you can delete or replace objects
  - If a policy exists, objects can only be deleted or replaced once their age is greater than the policy
  - Applies retroactively to existing and new objects added to the bucket

- Lock a retention policy to permanently set it on the bucket.
  - Once set, you cannot remove or reduce the retention period
  - A bucket cannot be deleted unless every object in the bucket has met the retention period
  - The retention period of a locked object can be increased
  - Locking a retention policy can help with data compliance regulations

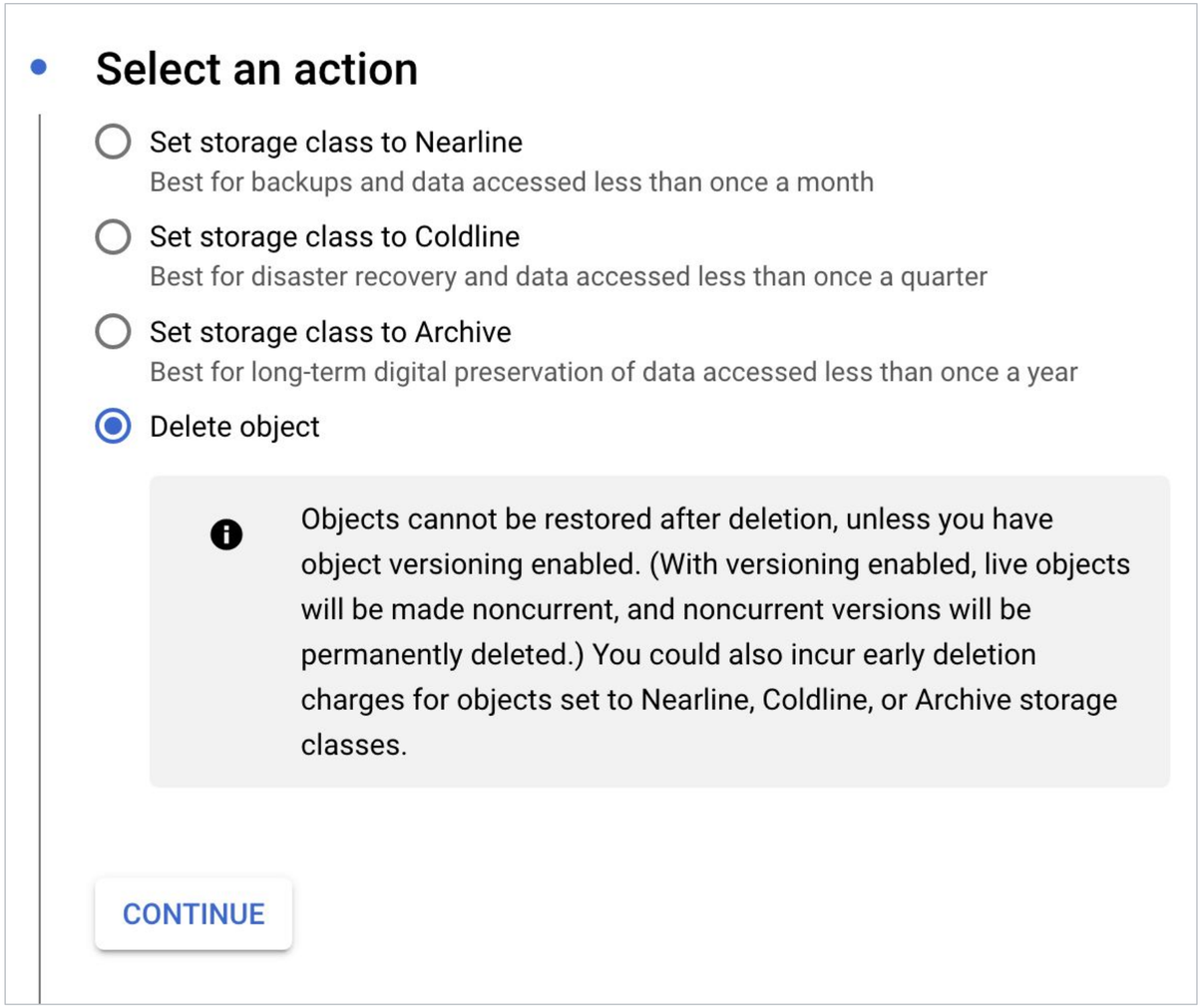
Enabling Object Versioning

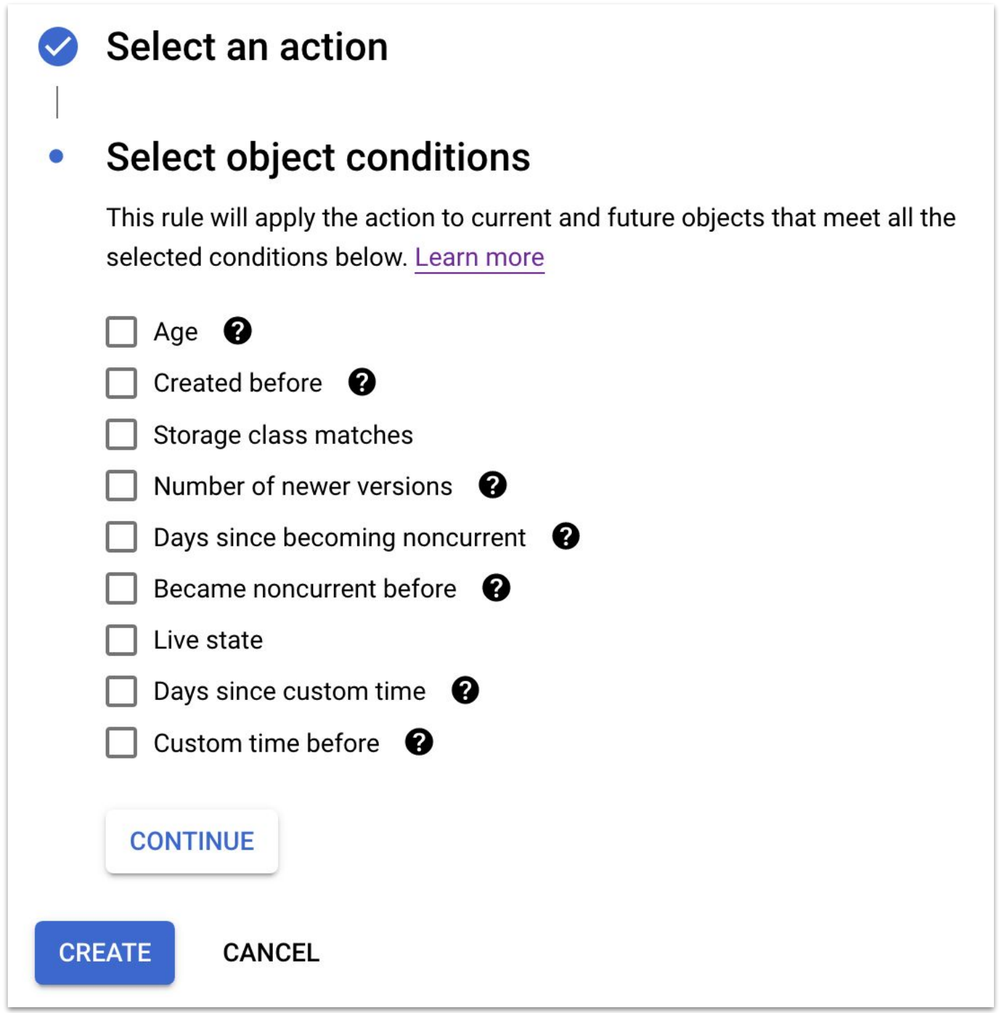


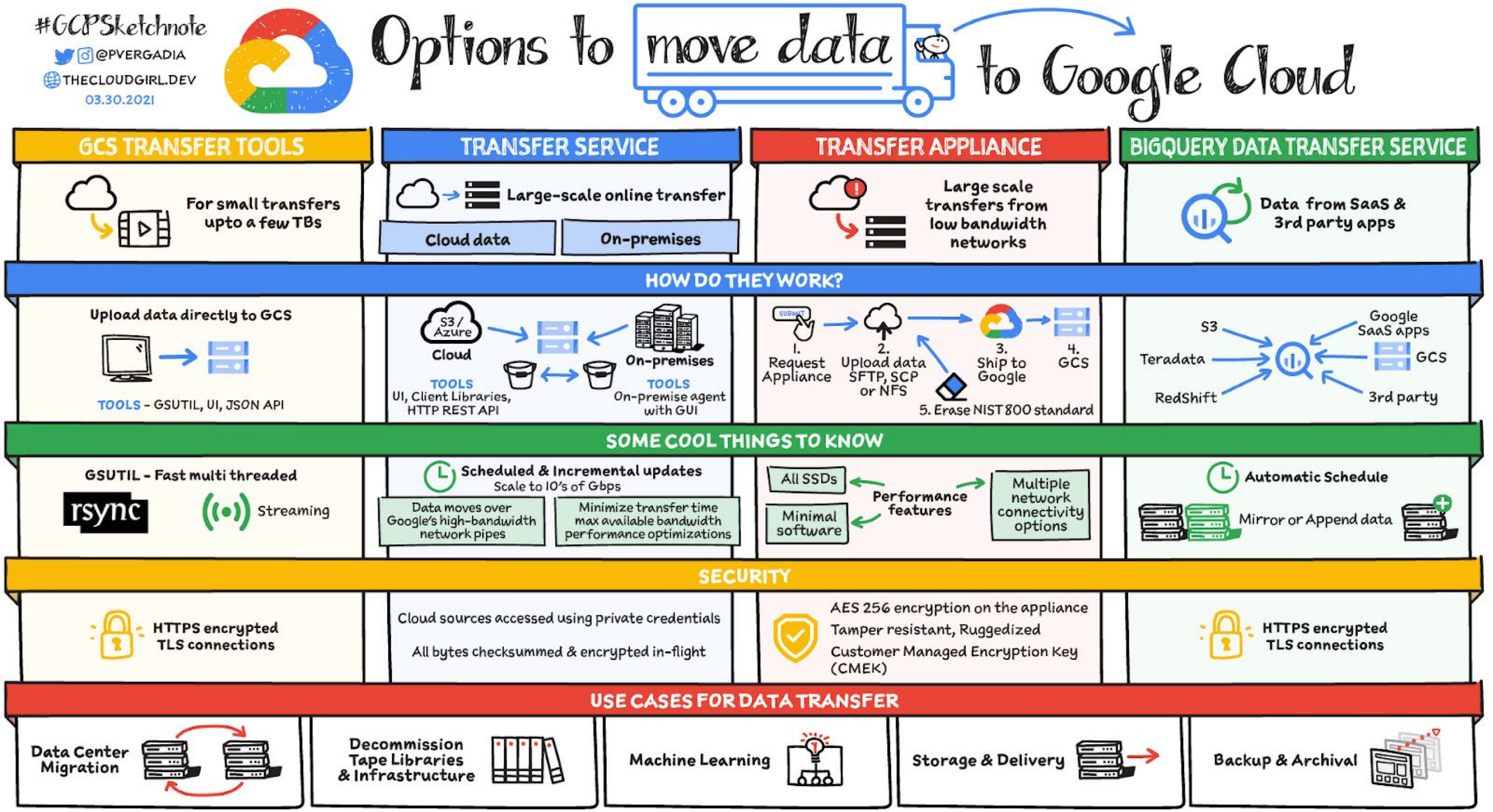

```
gsutil versioning set on
gs://acme-data-bucket
```

[Object Versioning]

Set Lifecycle policy - Console





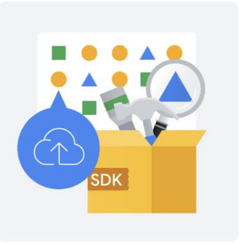


https://www.youtube.com/watch?v=lt9bOxlsKs4

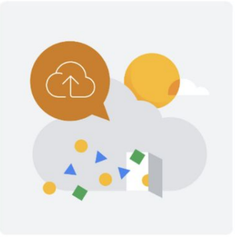
Bringing data into Cloud Storage

- Online transfer
  - Drag and drop in the Goggle Cloud Console
  - Upload / download via the command line
    - `gsutil`/`gcloud`
- Storage Transfer Service
  - Create jobs to run once or on a scheduled bases
  - Transfer data from
    - Other clouds (AWS, Azure), URL, Posix filesystem
    - On-premise
    - Cloud Storage Bucket to Cloud Storage Bucket
- Transfer Appliance
  - Rackable, high-capacity storage server leased from Google
  - Used when the amount of data to be transferred would take too much time given network bandwidth between the customer location and Google

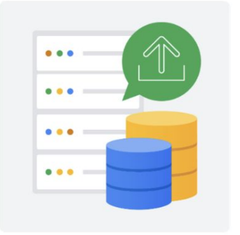
How long will it take to transfer data?



Online transfer



Storage Transfer Service



Transfer Appliance

Saving files to Cloud Storage - gsutil

gsutil - Command line to upload/download files

- For smaller amounts of data (<1TB) if have adequate bandwidth

Upload

```
gsutil cp OBJECT_LOCATION gs://DESTINATION_BUCKET_NAME/
gsutil cp desktop/myfile.png gs://my-bucket/
```

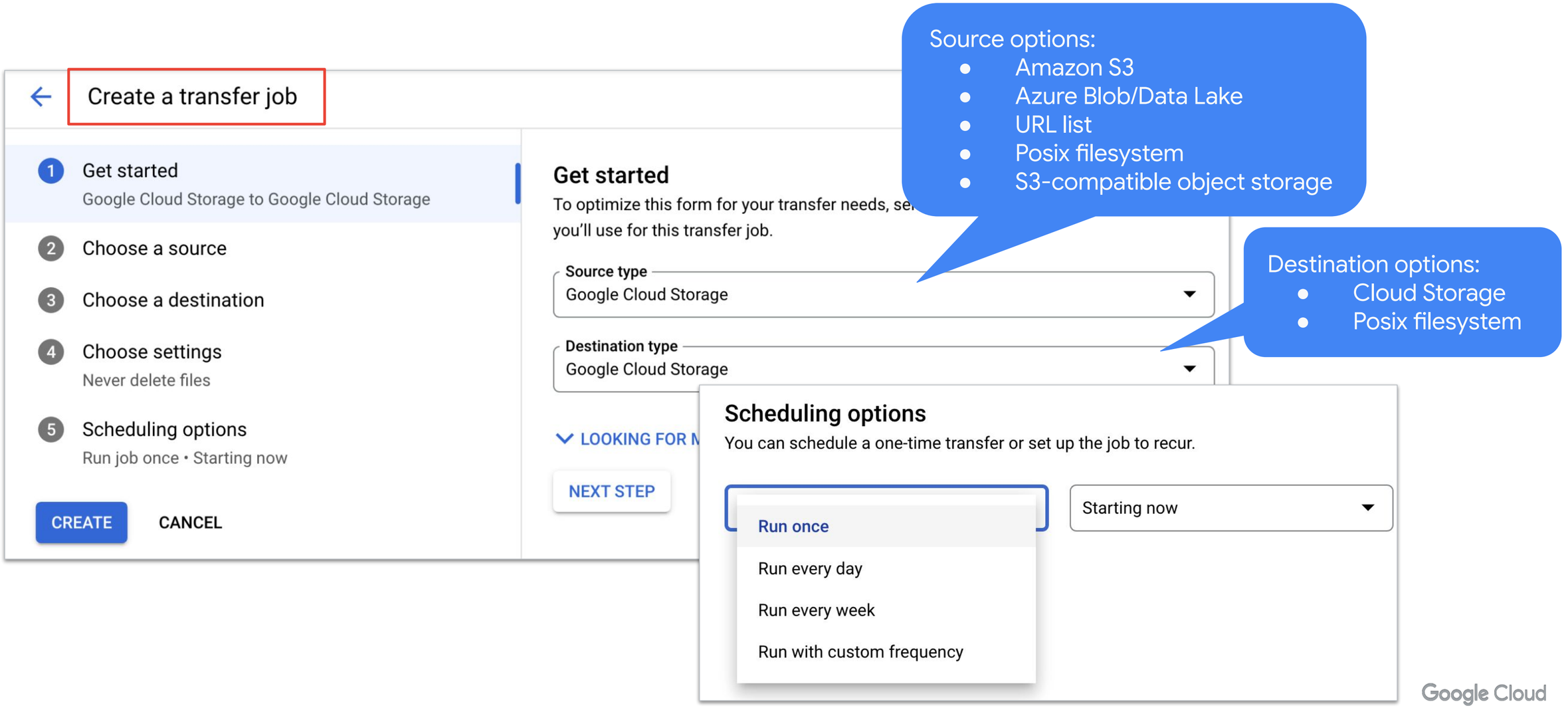
Download

```
gsutil cp gs://BUCKET_NAME/OBJECT_NAME SAVE_TO_LOCATION
gsutil cp gs://my-bucket/* desktop/file-folder/
```

New: `gcloud storage` commands were introduced in 2022

Saving files to Cloud Storage - Transfer Service

- Efficient way of moving data into buckets from other buckets, S3 or on-premise servers



Saving files to Cloud Storage - Transfer Appliance

- If you have more than 10-20TB, consider requesting a transfer appliance
  - Load the data locally and then ship it back
- Use when transfer times would be too long over the internet

Transferring data into the cloud can be challenging

		1 Mbps	10 Mbps	100 Mbps	1 Gbps	10 Gbps	100 Gbps
	1 GB	3 hrs	18 mins	2 mins	11 secs	1 sec	.1 secs
	10 GB	30 hrs	3 hrs	18 mins	2 mins	11 secs	1 sec
	100 GB	12 days	30 hrs	3 hrs	18 mins	2 mins	11 secs
	1 TB	124 days	12 days	30 hrs	3 hrs	18 mins	2 mins
	10 TB	3 years	124 days	12 days	30 hrs	3 hrs	18 mins
Typical enterprise	100 TB	34 years	3 years	124 days	12 days	30 hrs	3 hrs
	1 PB	340 yrs	34 years	3 years	124 days	12 days	30 hrs
	10 PB	3.404 yrs	340 yrs	34 years	3 years	124 days	12 days
	100 PB	34,048 yr	3,404 yrs	340 yrs	34 years	3 years	124 days

Storage and database services

Object	File	Relational			Non-relational		Warehouse
Cloud Storage	Filestore	Cloud SQL	Spanner	AlloyDB	Firestore	Bigtable	BigQuery
Good for: Binary or object data	**Good for:** Network Attached Storage (NAS)	**Good for:** Web frameworks	**Good for:** RDBMS + scale, HA, HTAP	**Good for:** Legacy Applications	**Good for:** Hierarchical, mobile, web	**Good for:** Heavy read + write, events,	**Good for:** Enterprise data warehouse
Such as: Images, media serving, backups	**Such as:** Latency sensitive workloads	**Such as:** CMS, ecommerce	**Such as:** User metadata, Ad/Fin/ MarTech	**Such as:** Transaction Processing, Enterprise workloads	**Such as:** User profiles, game state	**Such as:** Ad tech, financial, IoT	**Such as:** Analytics, dashboards

Next discussion

Filestore

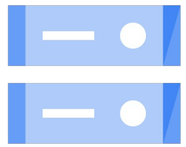
Discussed in module 1

- High-performance, fully managed network attached storage (NAS) for files
  - Mount as file shares on Compute Engine instances
  - Used to store and serve files such as documents, images, videos, audio files, and other data
- Pay for what you use
- Capacity scales automatically scale based on demand
- Use cases:
  - Enterprise application migrations (SAP),
  - Media rendering where file shares are needed
  - Web content management

YouTube video: https://www.youtube.com/watch?v=CUwpXqEitA0

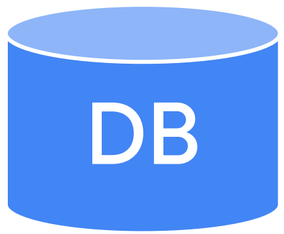
Filestore

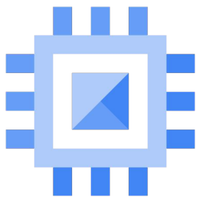
Managed relational and non-relational databases

Object	File	Relational			Non-relational		Warehouse
 Cloud Storage	 Filestore	 Cloud SQL	 Spanner	 AlloyDB	 Firestore	 Bigtable	 BigQuery
Good for: Binary or object data	**Good for:** Network Attached Storage (NAS)	**Good for:** Web frameworks	**Good for:** RDBMS + scale, HA, HTAP	**Good for:** Legacy Applications	**Good for:** Hierarchical, mobile, web	**Good for:** Heavy read + write, events,	**Good for:** Enterprise data warehouse
Such as: Images, media serving, backups	**Such as:** Latency sensitive workloads	**Such as:** CMS, ecommerce	**Such as:** User metadata, Ad/Fin/ MarTech	**Such as:** Transaction Processing, Enterprise workloads	**Such as:** User profiles, game state	**Such as:** Ad tech, financial, IoT	**Such as:** Analytics, dashboards

Coming up

Custom and Managed Solutions

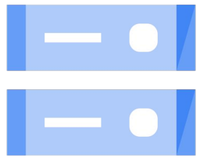




Compute Engine

Storage

Bigtable



Cloud Storage

Cloud SQL

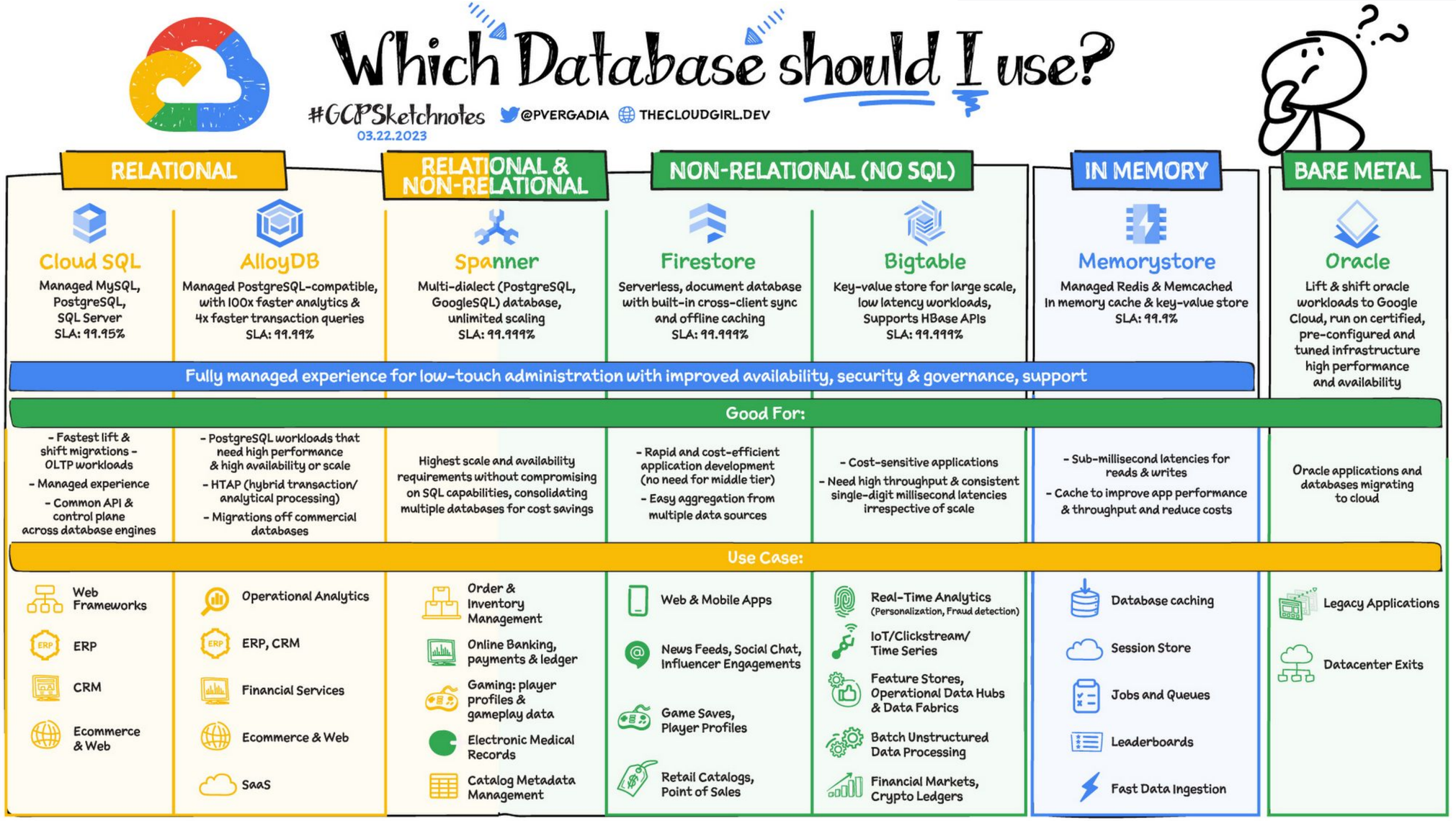
Spanner

Firestore

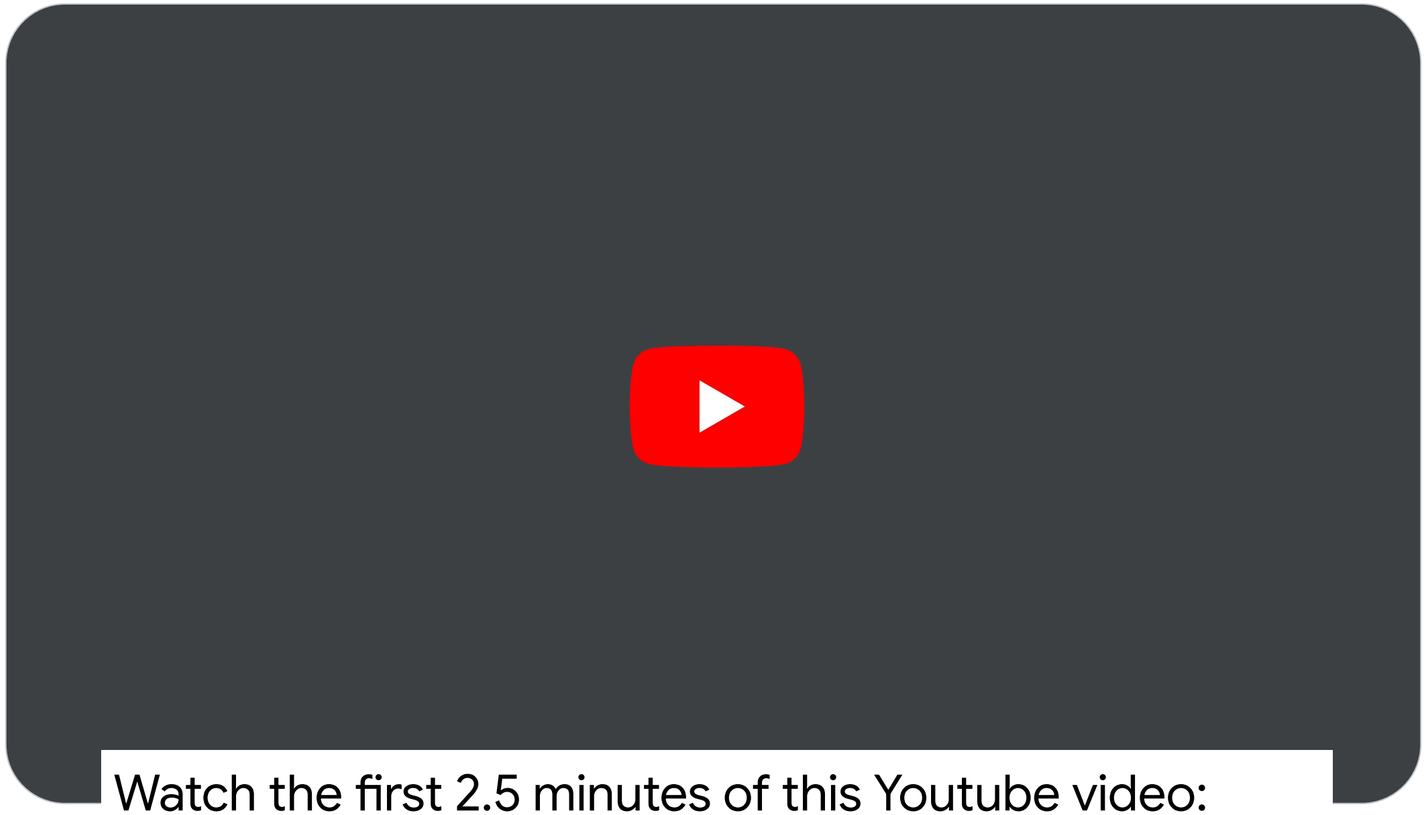
AlloyDB

Database Options

[Your Google Cloud database options, explained]



Database Options



Watch the first 2.5 minutes of this Youtube video:
Compares relational database to NoSQL databases
https://www.youtube.com/watch?v=v_hR4K4auoQ

Relational vs non-relational databases

Relational databases store information in tables, rows and columns that structure the data. They use relational semantics (i.e. a column in one table can point to data in another table) to ensure data consistency and enable complex queries across multiple tables. Relational databases are used when the structure of the data doesn't change often, such as in banking or supply chain inventory management.

- Cloud SQL
- AlloyDB
- Spanner

Non-relational databases (or NoSQL databases) store complex, unstructured data in a non-tabular form such as documents and key-value stores. Non-relational databases are often used when large quantities of complex and diverse data need to be organized, or where the structure of the data is regularly evolving to meet new business requirements, such as personalization and web and mobile applications.

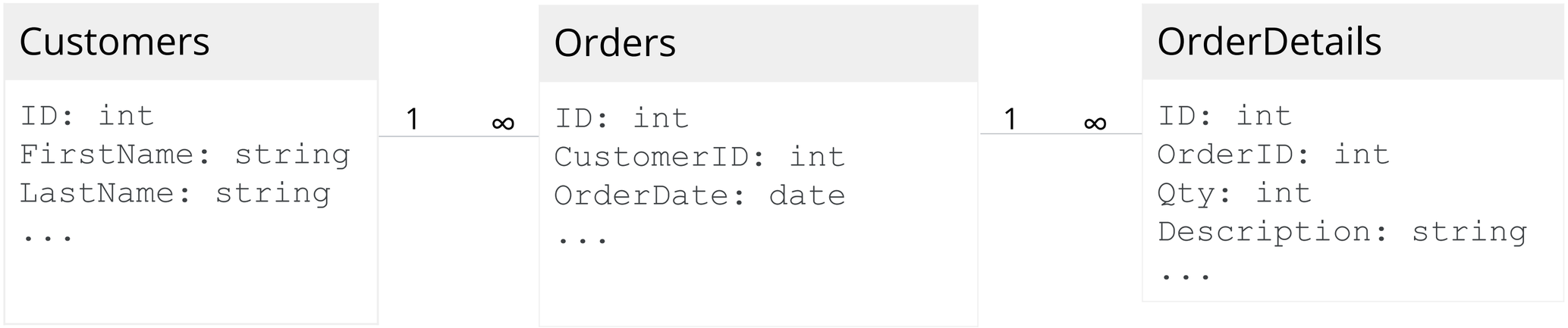
- Bigtable
- Cloud Firestore

From: Make your database your secret advantage with Google Cloud

What does data in a relational database look like?

Look at developer slide for better example

- Tables contain fields, indexes, and constraints
- Primary key ensures each row in a table is unique
- Relationships are constraints that ensure a parent row cannot be deleted if there are child rows in another table



Types of NoSQL Databases

Database Model	Description	Google Cloud	Other Databases
Key-value stores	Data is stored in key-value pairs	**Cloud Memorystore**	Redis Memcached
Document stores	Data is stored in some standard format like XML or JSON. Nested and hierarchical data can be stored together	**Cloud Firestore**	MongoDB CouchDB DynamoDB
Wide-column stores	Key identifies a row in a table. Columns can be different within each row	**Bigtable**	Cassandra HBase

What does data in a NoSQL database look like?

Depends on the type of database

Firestore

Documents live in collections, which are simply containers for documents. For example, you could have a `users` collection to contain your various users, each represented by a document:

users

alovelace

```
first : "Ada"
last : "Lovelace"
born : 1815
```

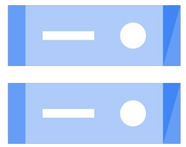
aturing

```
first : "Alan"
last : "Turing"
born : 1912
```

Bigtable

	Flight_Information				
Row Key	Origin	Destination	Departure	Arrival	Passengers
ATL#arrival#20190321-1121	ATL	LON	20190321-0311	20190321-1121	158
ATL#arrival#20190321-1201	ATL	MEX	20190321-0821	20190321-1201	187
ATL#arrival#20190321-1716	ATL	YVR	20190321-1014	20190321-1716	201

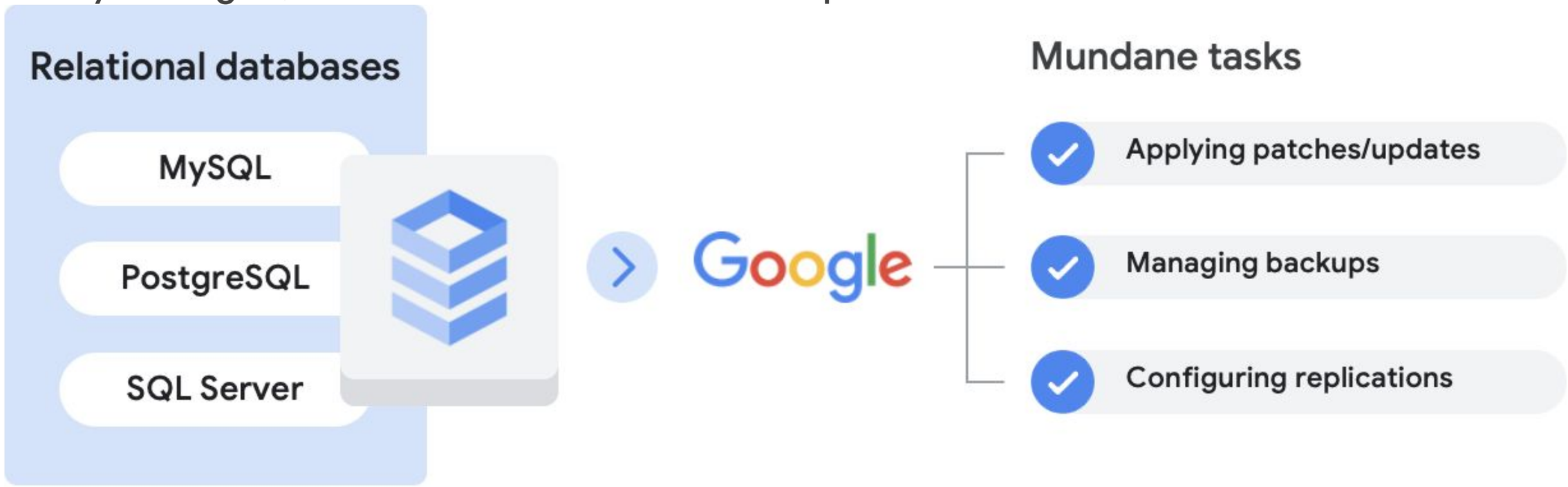
Storage and database services

Object	File	Relational			Non-relational		Warehouse
							
Cloud Storage	Filestore	Cloud SQL	Spanner	AlloyDB	Firestore	Bigtable	BigQuery
Good for: Binary or object data	**Good for:** Network Attached Storage (NAS)	**Good for:** Web frameworks	**Good for:** RDBMS + scale, HA, HTAP	**Good for:** Legacy Applications	**Good for:** Hierarchical, mobile, web	**Good for:** Heavy read + write, events,	**Good for:** Enterprise data warehouse
Such as: Images, media serving, backups	**Such as:** Latency sensitive workloads	**Such as:** CMS, ecommerce	**Such as:** User metadata, Ad/Fin/	**Such as:** Transaction Processing, Enterprise orkloads	**Such as:** User profiles, game state	**Such as:** Ad tech, financial, IoT	**Such as:** Analytics, dashboards

Next discussion

Cloud SQL

A fully managed, cloud-based alternative to on-premise



[The business value of Cloud SQL: how companies speed up deployments, lower costs and boost agility]

Cloud SQL - shared responsibilities

Application Development
Schema Design
Query Optimization

Managed by Customer

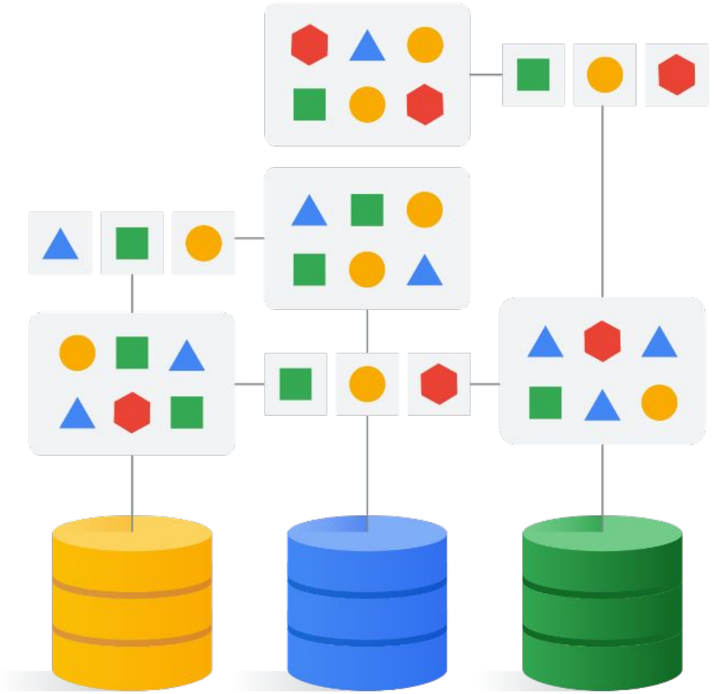
Storage Scaling
Availability
Database Maintenance
Monitoring
Security
OS
Hardware & Network

Managed by Google

Cloud SQL supports a variety of use cases

Here are just a few:

- Storing and managing relational data such as customer orders, product inventory, and financial transactions
- Migrating on-premises MySQL, PostgreSQL, Microsoft SQL Server databases to the cloud
- Replicating data from an on-premise database to Cloud SQL for disaster recovery and high availability



Cloud SQL - customer use case

- Qlue: Boosting intelligence about activity in Indonesian cities
- Applications include
  - Dashboard to track and resolve incidents of flooding, crime, fire, illegal rubbish dumping, and more
  - Monitor energy usage and traffic congestion in real time
  - Mobile app that enables citizens to report incidents

> "If we used a conventional Infrastructure-as-a Service offering, we would have had to dedicate a team to manage operating systems, libraries, services, and load balancing. With Google Cloud, we are able to focus purely on developing our applications and growing the business."
>
> —*Surya Darmadi, Chief Operating Officer, Qlue*

Scaling Cloud SQL Databases

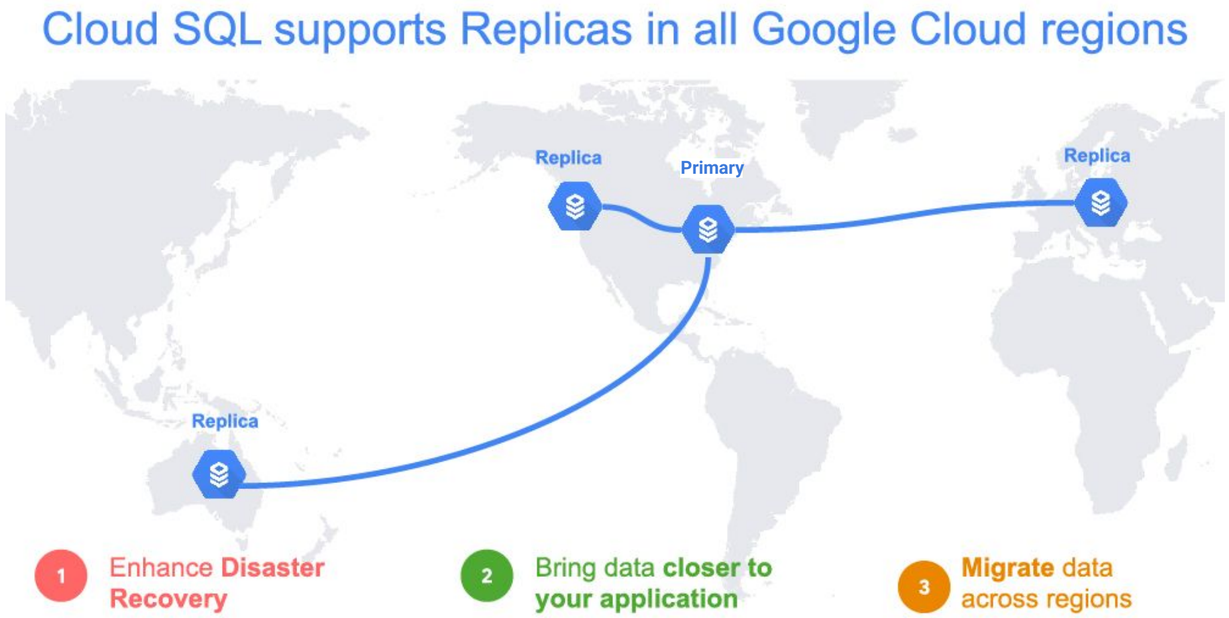
Cloud SQL can scale in the following ways

- **Vertical scaling:**
  - Increasing the amount of computing resources (such as CPU and memory) allocated to a single database instance.
  - Can be done with a few clicks in the Cloud Console, and requires no downtime.
- **Horizontal scaling:**
  - Adding additional read replicas to distribute read workloads and improve performance.
  - Can be added on demand, and there's no limit to the number of replicas that can be added.

Cloud SQL Read Replicas

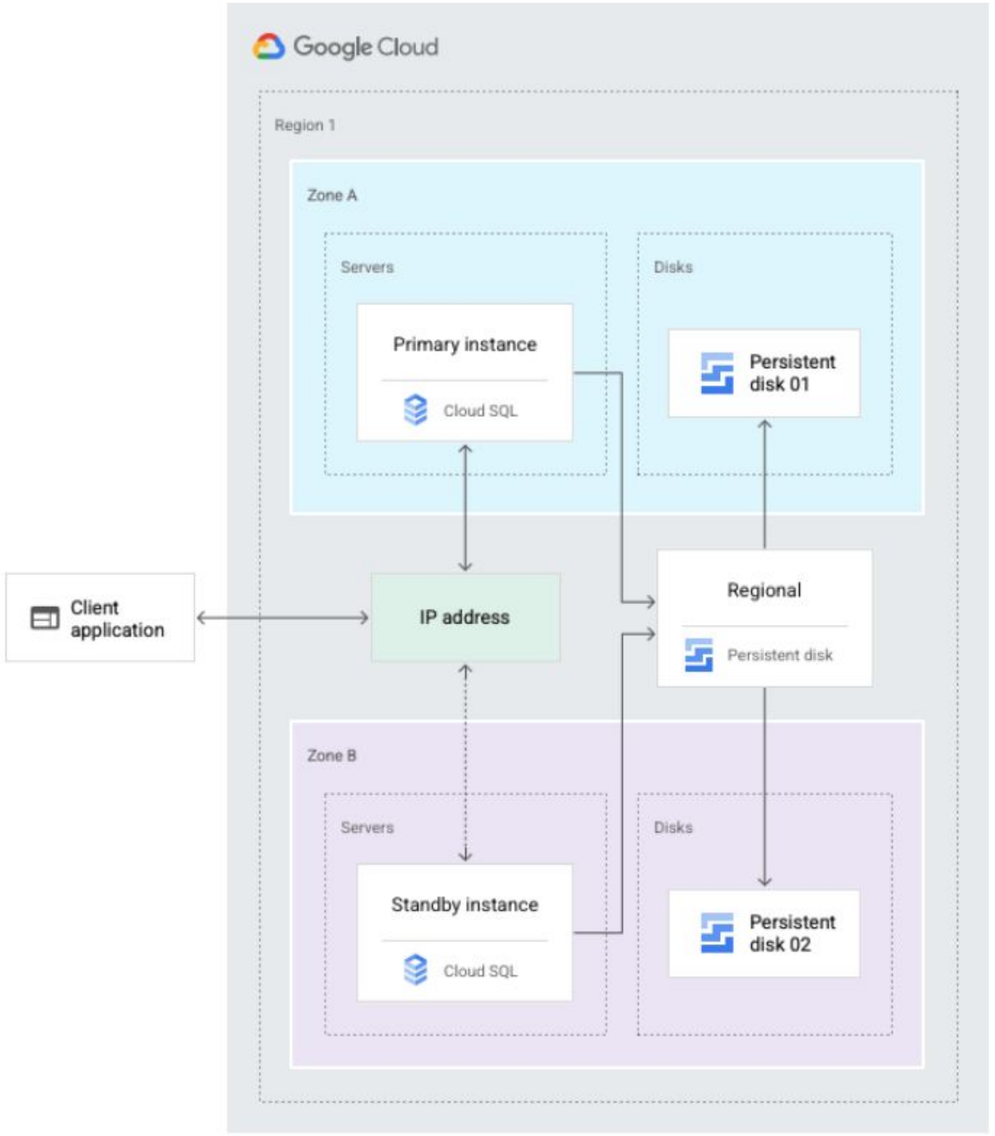
Fully managed read replica in a different region(s) than that of the primary instance

- Enhance DR
- Bring data closer to your applications (performance and cost implications)
- Migrate data across regions
- Data and other changes on the primary instance are updated in almost real time on the read replicas

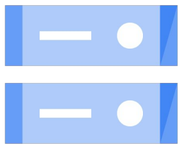


Cloud SQL - High Availability

- Provides automatic failover if a zone or instance become unavailable
- The primary instance is in one zone in a region
  - The failover instance is in another zone
- Synchronous replication is used to copy all writes from the primary disk to the replica



Storage and database services

Object	File	Relational			Non-relational		Warehouse
							
Cloud Storage	Filestore	Cloud SQL	Spanner	AlloyDB	Firestore	Bigtable	BigQuery
Good for: Binary or object data	**Good for:** Network Attached Storage (NAS)	**Good for:** Web frameworks	**Good for:** RDBMS + scale, HA, HTAP	**Good for:** Legacy Applications	**Good for:** Hierarchical, mobile, web	**Good for:** Heavy read + write, events,	**Good for:** Enterprise data warehouse
Such as: Images, media serving, backups	**Such as:** Latency sensitive workloads	**Such as:** CMS, ecommerce	**Such as:** User metadata, Ad/Fin/ MarTech	**Such as:** Transaction Processing, Enterprise workloads	**Such as:** User profiles, game state	**Such as:** Ad tech, financial, IoT	**Such as:** Analytics, dashboards

Next discussion

Spanner

A enterprise-grade, globally distributed, externally consistent relational database having unlimited scalability and industry-leading 99.999% availability

- Powers Google's most popular, globally available products, like YouTube, Drive, and Gmail
- Capable of processing more than 1 billion queries per second at peak
- For any workload, large or small, that cannot tolerate downtime, and requires high availability
- Regional and multi-regional deployments
  - SLA: Multi-regional: 99.999%
  - SLA: Regional: 99.99%
- Supports ANSI standard SQL

[Spanner myths busted]

Spanner

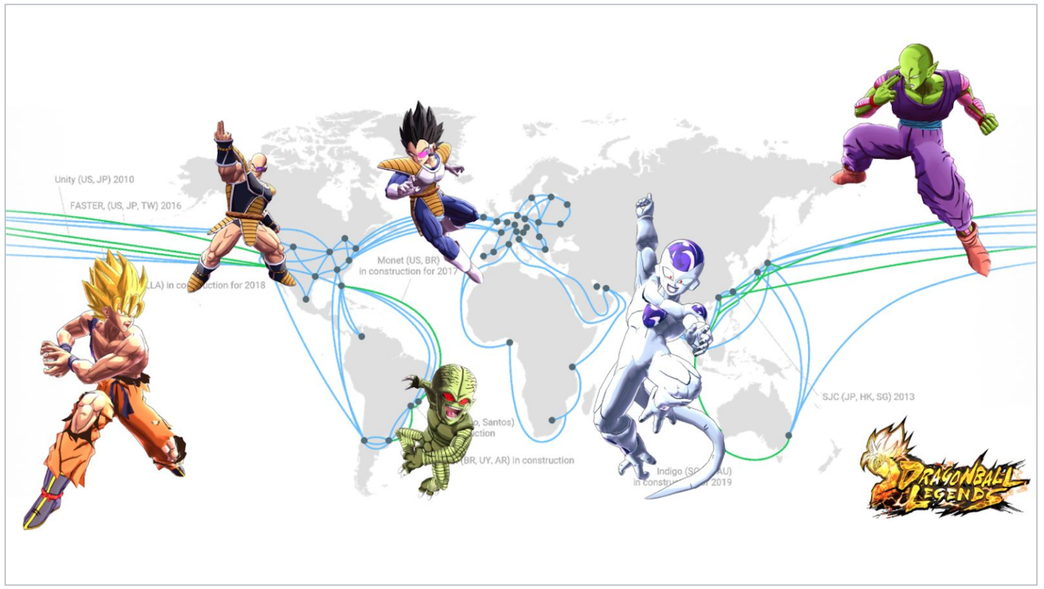
Spanner supports a variety of use cases

- Large-scale, multi-regional data storage
    - Can store and manage terabytes or petabytes of data across multiple regions
- Online transaction processing (OLTP) applications:
    - Can support OLTP workloads with low latency and high throughput, making it suitable for applications that require real-time data processing, such as e-commerce or financial services
- Banking and financial services
    - Spanner's high-availability and consistency guarantees make it well-suited for use cases in the financial services industry, such as stock trading or payment processing
- Geographically distributed data management
    - Spanner supports multiple geographic locations and provides a globally consistent view of data, making it ideal for use cases that require a database that can handle data distributed across multiple regions or continents

Spanner - customer use case

[Behind the scenes with the Dragon Ball Legends GC backend]

- [Dragon Ball Legends] - mobile game from Bandai Namco Entertainment
- Requirements were:
  - Global backend that could scale with millions of players and still perform well.
  - Global reliable, low latency network to support multi-region player-versus-player battles
  - Real-time data analytics to measure and evaluate how people are playing the game and adjust it on-the-fly.

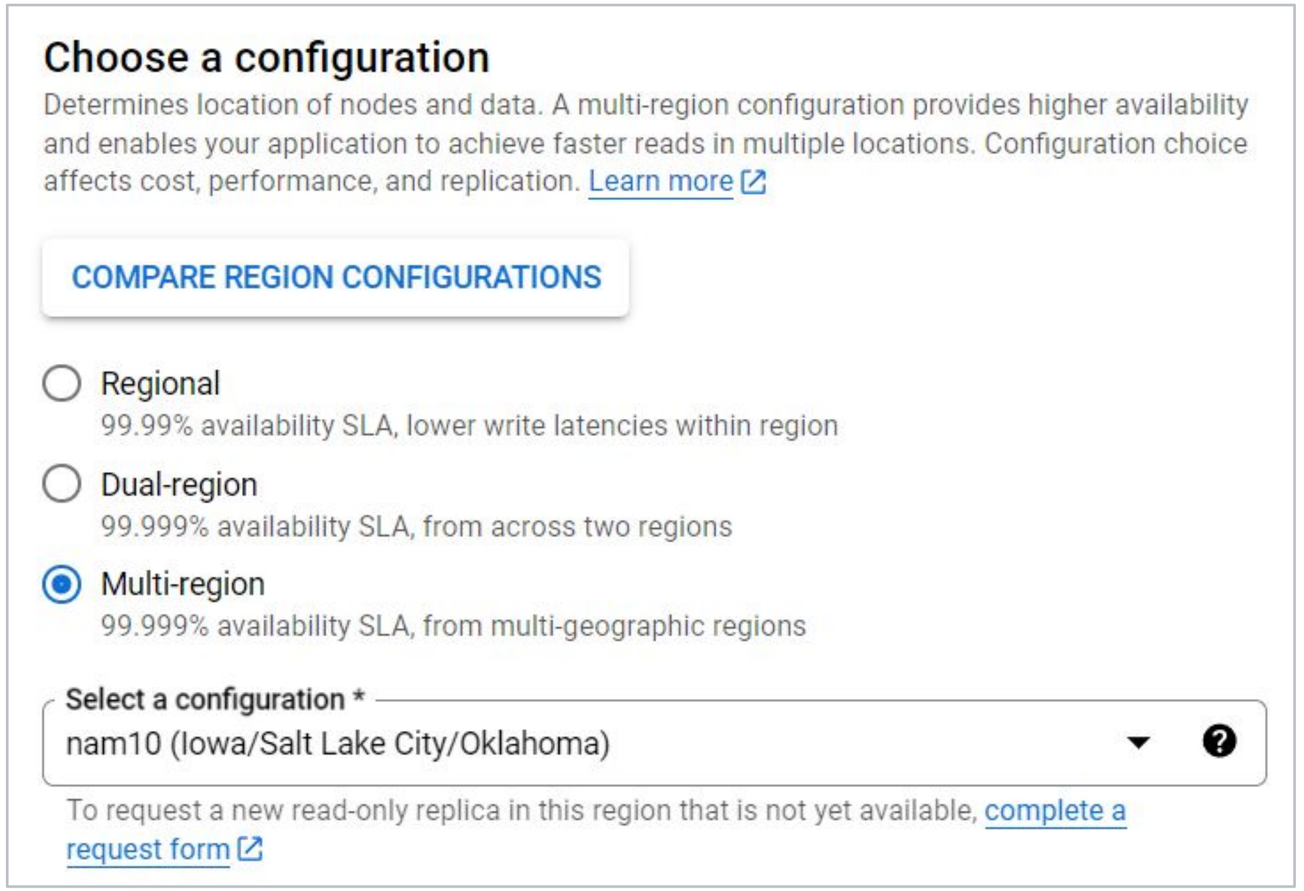


Another use case: [Google Photos]

Scaling Spanner

Scales out (horizontal scaling)

- Manually add nodes/processing units to support more data and users as needed
- Turn on autoscaling to automatically adjust the number of nodes in an instance to handle changing traffic patterns and load



Cloud SQL and Spanner

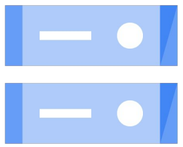
Cloud SQL

- Max 64TB data
- Max 60,000 IOPS
- High Availability via master / standby
- 99.95% SLA
- Vertically Scalable via reprovisioning
- Horizontally Scalable by DB Sharding and read replicas
- Planned maintenance windows

Spanner

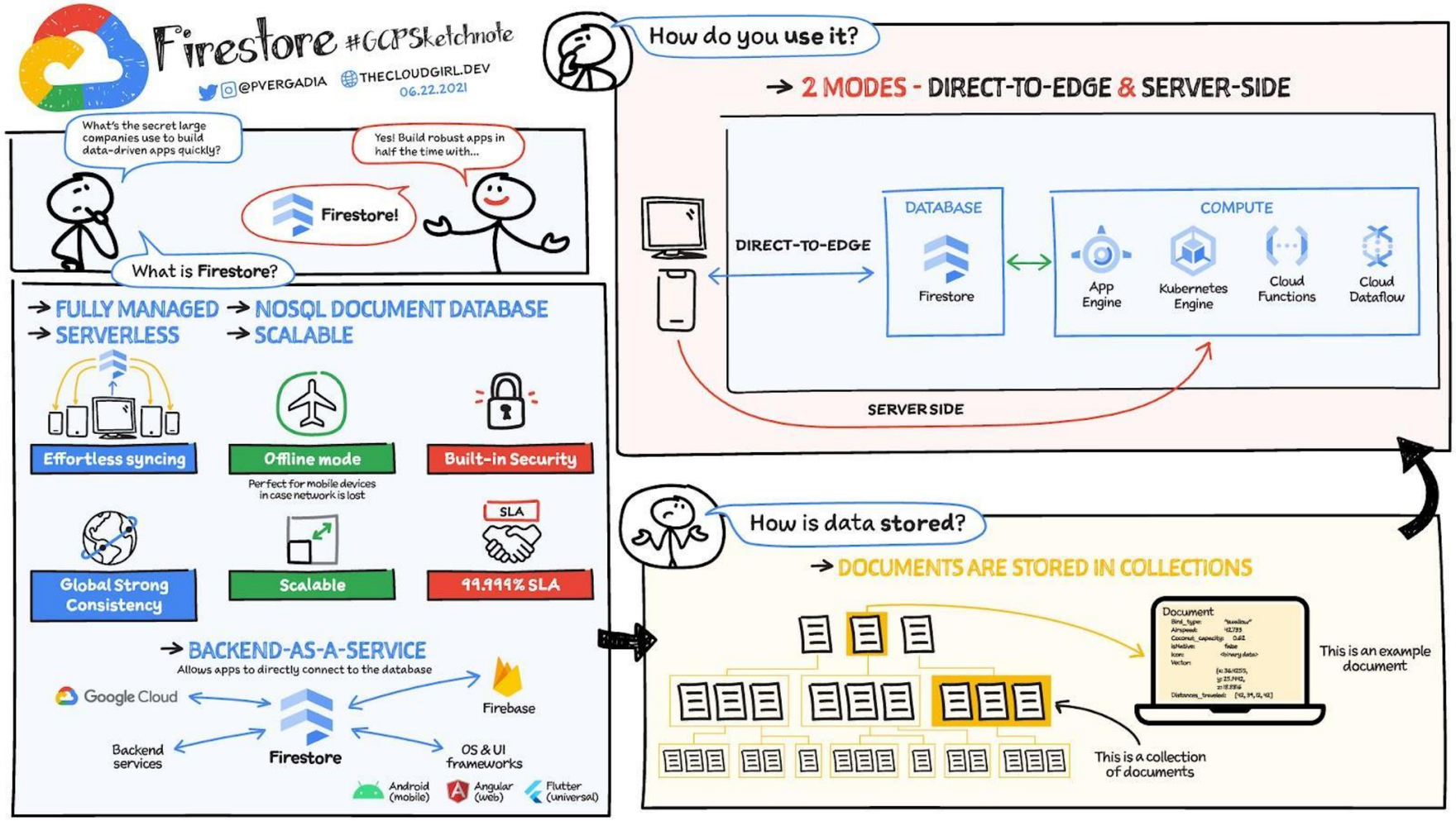
- 10TB / node data
- 10,000 Reads / node / sec
- 2,000 Writes / node / sec
- Distributed service - always available
- 99.99% SLA (regional)
- 99.999% SLA (multi-region)
- Horizontally Scalable
- Dynamically re-scale within minutes
- Dynamic schema updates
- No maintenance managed service

Storage and database services

Object	File	Relational			Non-relational		Warehouse
							
Cloud Storage	Filestore	Cloud SQL	Spanner	AlloyDB	Firestore	Bigtable	BigQuery
Good for: Binary or object data	**Good for:** Network Attached Storage (NAS)	**Good for:** Web frameworks	**Good for:** RDBMS + scale, HA, HTAP	**Good for:** Legacy Applications	**Good for:** Hierarchical, mobile, web	**Good for:** Heavy read + write, events,	**Good for:** Enterprise data warehouse
Such as: Images, media serving, backups	**Such as:** Latency sensitive workloads	**Such as:** CMS, ecommerce	**Such as:** User metadata, Ad/Fin/ MarTech	**Such as:** Transaction Processing, Enterprise workloads	**Such as:** User profiles, game state	**Such as:** Ad tech, financial, IoT	**Such as:** Analytics, dashboards

Next discussion

Cloud Firestore



[All you need to know about Firestore: A cheatsheet]

Cloud Firestore

- Completely managed, document store, NoSQL Database
  - No administration, no maintenance, nothing to provisio
- 1 GB per month free tier
- Indexes created for every property by default
  - Secondary indexes and composite indexes are suppor
- Supports ACID* transactions
- For mobile and web at global scale
  - Live synchronization and offline support
- Multi-region replication

*ACID explained: https://en.wikipedia.org/wiki/ACID

Firestore example data

An index is created for every property so that queries are extremely fast

Collections

Documents live in collections, which are simply containers for documents. For example, you could have a `users` collection to contain your various users, each represented by a document:

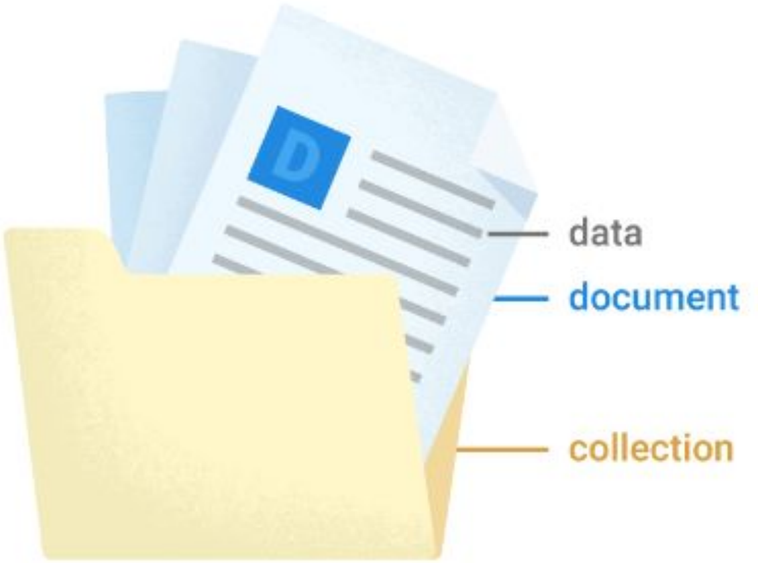
users

alovelace

```
first : "Ada"
last : "Lovelace"
born : 1815
```

aturing

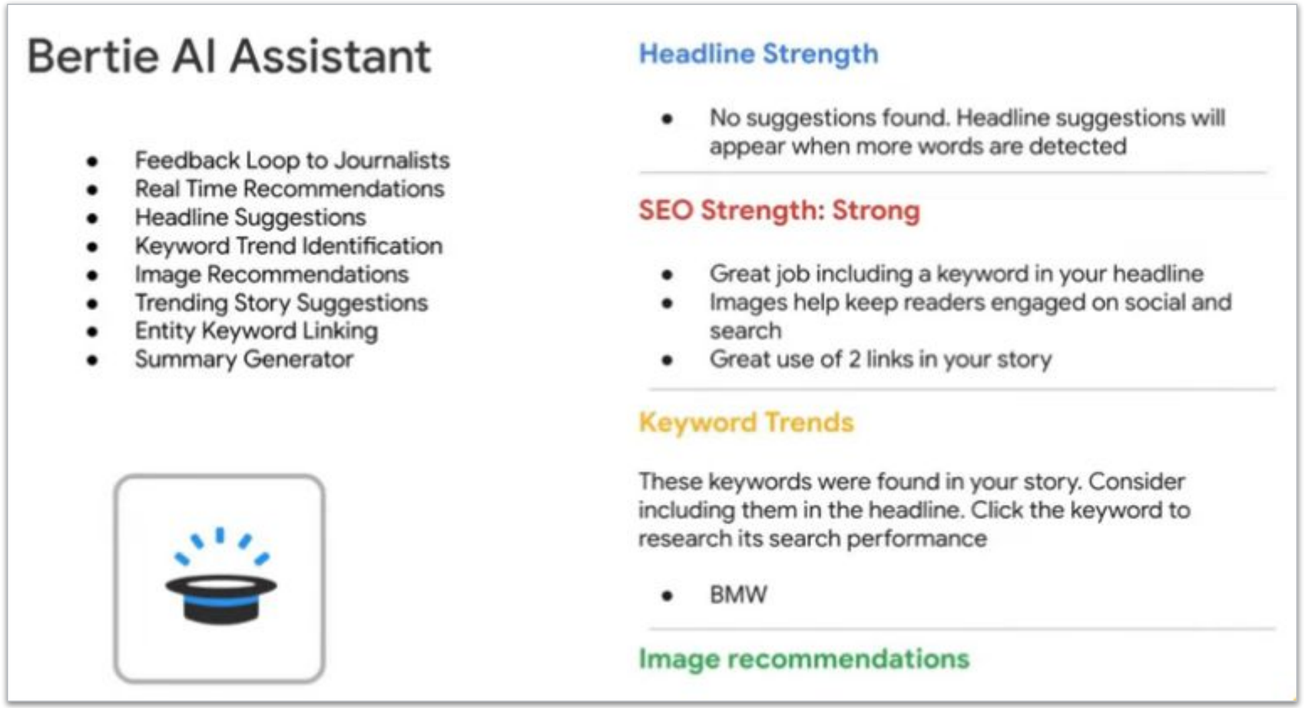
```
first : "Alan"
last : "Turing"
born : 1912
```



Cloud Firestore is schemaless, so you have complete freedom over what fields you put in each document and what data types you store in those fields. Documents within the same collection can all contain different fields or store different types of data in those fields. However, it's a good idea to use the same fields and data types across multiple documents, so that you can query the documents more easily.

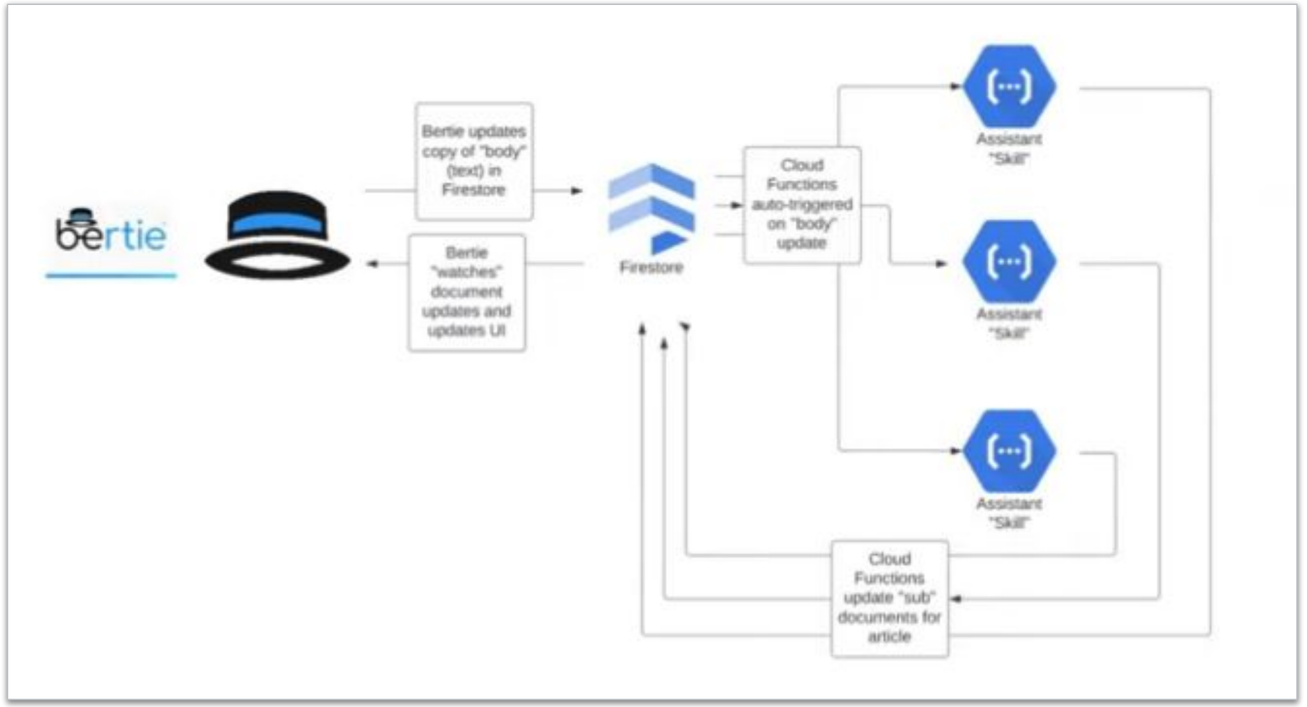
Firestore - customer use case

- Forbes created Bertie - an AI assistant for journalists
- Journalists upload their content and Bertie provides feedback
  - Strength of the article's headline
  - Keywords needed for search engine optimization
  - Words to add to the headline to increase search performance



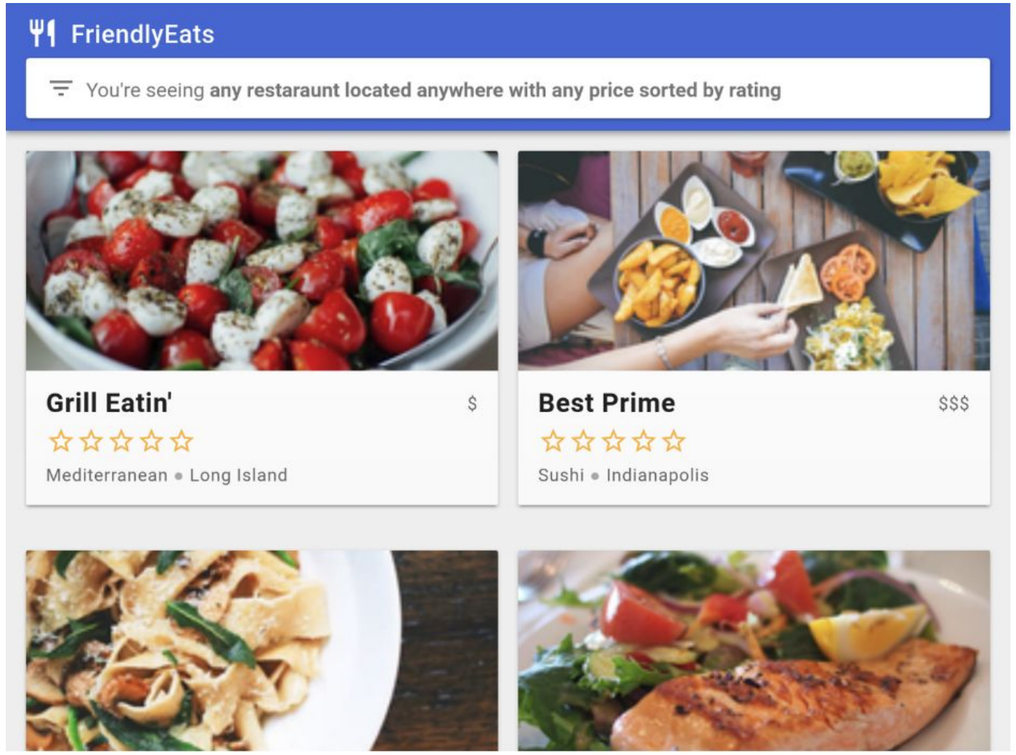
Forbes - Bertie AI Architecture

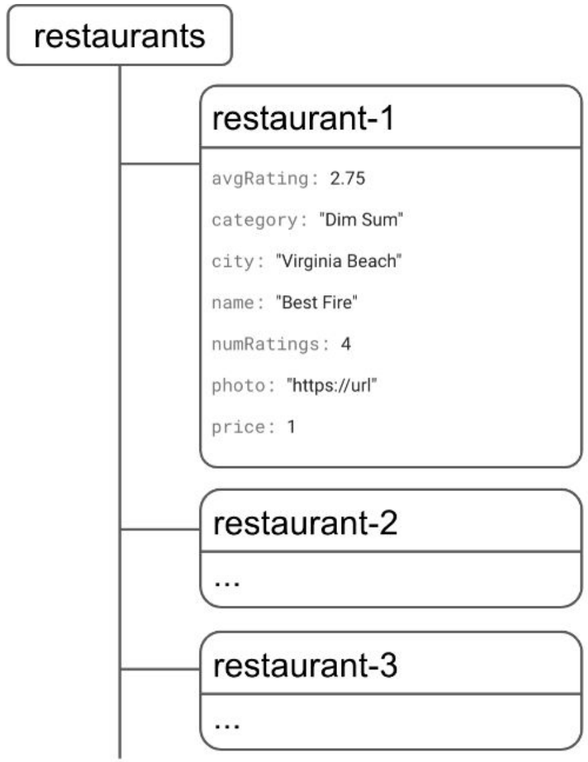
- Content is stored in Firestore
- Data updates trigger Cloud Functions
  - Each Cloud Function performs a different task
  - Results are written back to Firestore
  - Website is refreshed with the recommendations



YouTube video: https://www.youtube.com/watch?v=KVRxsRPhmoo

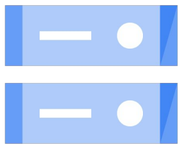
Firestore





Codelab: https://firebase.google.com/codelabs/firestore-web#0

Storage and database services

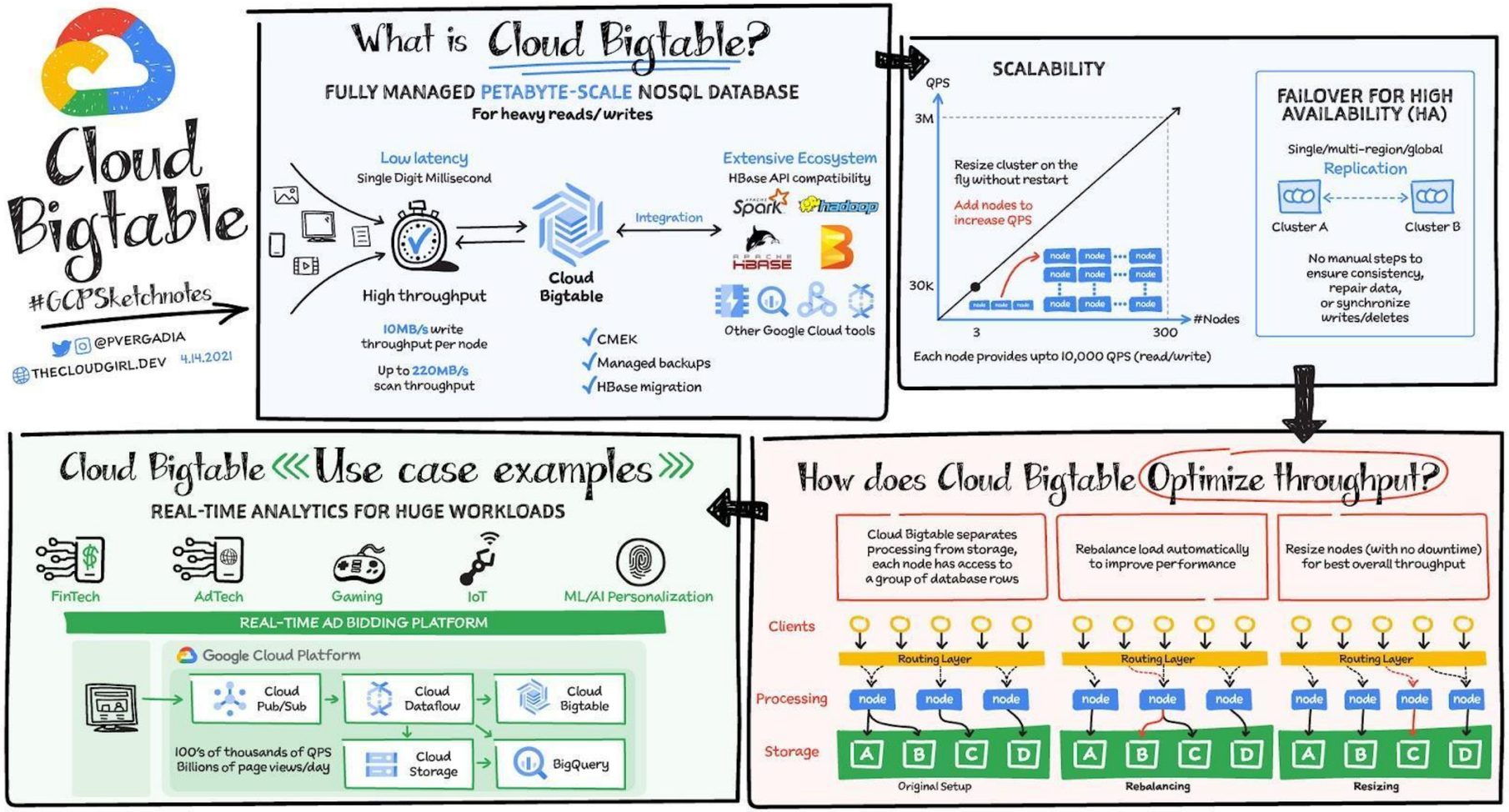
Object	File	Relational			Non-relational		Warehouse
							
Cloud Storage	Filestore	Cloud SQL	Spanner	AlloyDB	Firestore	Bigtable	BigQuery
Good for: Binary or object data	**Good for:** Network Attached Storage (NAS)	**Good for:** Web frameworks	**Good for:** RDBMS + scale, HA, HTAP	**Good for:** Legacy Applications	**Good for:** Hierarchical, mobile, web	**Good for:** Heavy read + write, events,	**Good for:** Enterprise data warehouse
Such as: Images, media serving, backups	**Such as:** Latency sensitive workloads	**Such as:** CMS, ecommerce	**Such as:** User metadata, Ad/Fin/ MarTech	**Such as:** Transaction Processing, Enterprise workloads	**Such as:** User profiles, game state	**Such as:** Ad tech, financial, IoT	**Such as:** Analytics, dashboards

Next discussion

Bigtable is a “big table”

- Analogy: spreadsheet with millions and millions of rows
  - Each row may have thousands of columns
  - Rows are uniquely identified by a single key column
- Used internally by Google for Search, Maps, etc.
- Provides extremely low latency
- Integrates easily with big data tools
  - Same API as HBase
    - Easily migrate on-premises HBase applications into Bigtable
- Example use cases
  - Streaming data from IoT devices
  - Time series data such as:
    - Daily stock prices of companies over a period of time
    - Temperature readings for cities every hour/day

Bigtable



[How BIG is Bigtable?]

Bigtable example data

Data that's likely to be accessed via the same request are grouped into column families

Column Families

Row Key	Flight_Information					Aircraft_Information			
	Origin	Destination	Departure	Arrival	Passengers	Capacity	Make	Model	Age
ATL#arrival#20190321-1121	ATL	LON	20190321-0311	20190321-1121	158	162	B	737	18
ATL#arrival#20190321-1201	ATL	MEX	20190321-0821	20190321-1201	187	189	B	737	8
ATL#arrival#20190321-1716	ATL	YVR	20190321-1014	20190321-1716	201	259	B	757	23

One key column

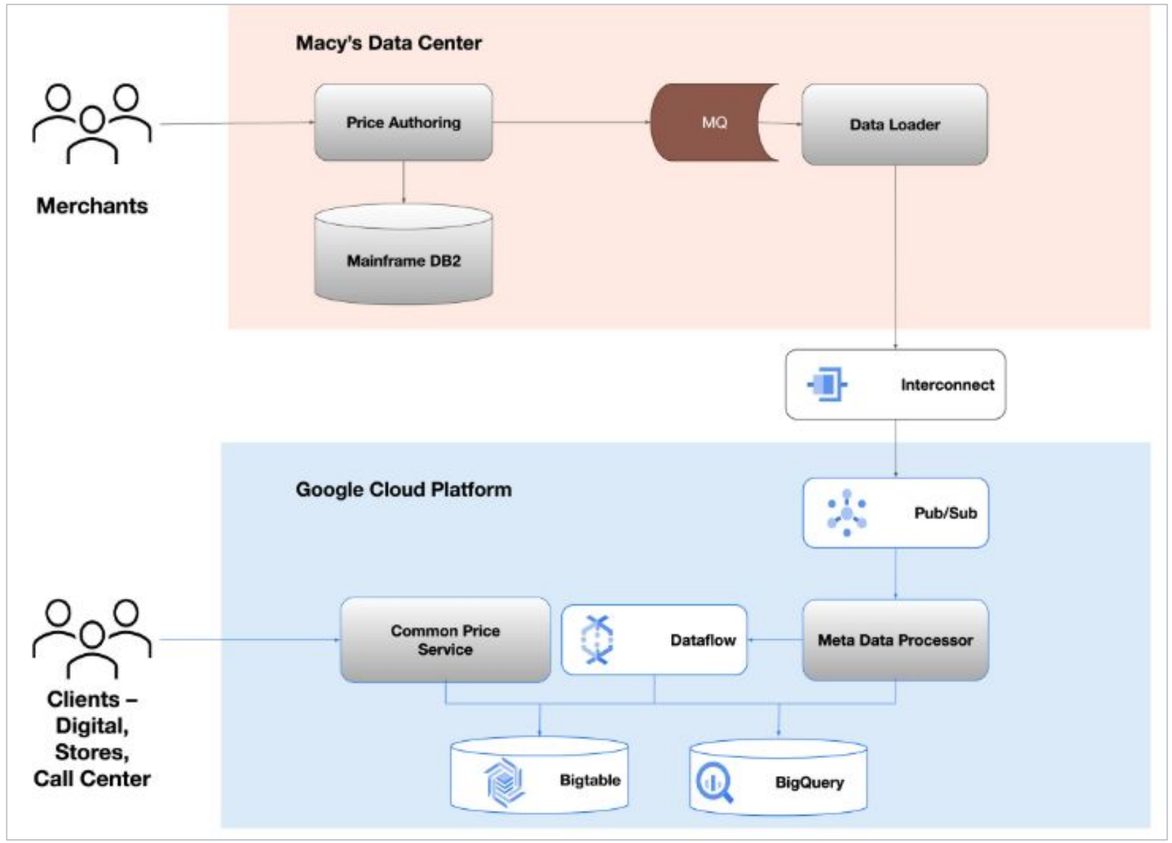
Max size of row is 256 MB

Bigtable example use cases

Healthcare	Data streaming in from connected healthcare devices, e.g., monitors in a hospital room
Retail	Streaming data used to generate product recommendations given items in a shopping cart
Gaming	Real-time player data for game analytics
IoT	Streaming data from connected devices and sensors, e.g., electric car battery status
Logs and Metrics	Storage of log data and metrics for analysis
Time series data	Storage of resource consumption like CPU and memory usage over time for multiple servers

Bigtable - customer use case: Macy's

- Macy's sells a wide range of merchandise, including apparel and accessories (men's, women's and children's), cosmetics, home furnishings and other consumer goods
  - 700+ stores across the US
- Bigtable provides data for the pricing system
  - Access pattern entails finding an item's ticket price based on a given division, location, and the universal price code
  - Can search millions of product codes within single digit milliseconds



[How Macy's enhances the customer experience with Google Cloud services]

Bigtable - customer use case: Bitly

- Bitly, the link & QR Code management platform, migrated 80 billion rows of link data to Bigtable.
    - Data was moved from a MySQL database to Bigtable in just six days

[From MySQL to NoSQL: Bitly's big move to Bigtable]

Where Bigtable comes in

As mentioned, after researching our options, the solution that best fit our needs was Bigtable. It offered the features we were looking for, including:

- A 99.999% SLA
- Limitless scale
- Single-digit millisecond latency
- A built-in monitoring system
- Multi-region replication
- Geo-distribution, which allows for seamless replication of data across regions and reduces latency
- On-demand scaling of compute resources and storage, which adjusts to user traffic and allows our system to grow and scale as needed
- Seamless integration with our general architecture; we use Google Cloud services for many other parts of our system, including the APIs that interact with these databases
- A NoSQL database that doesn't require relational semantics, as the datasets we're migrating are indexed on a single primary key in our applications

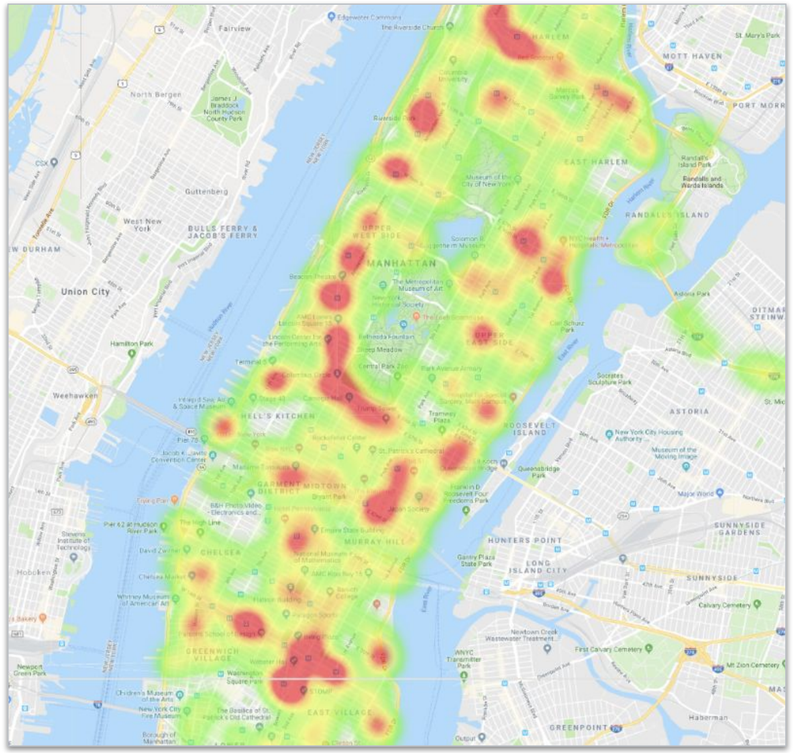
Bigtable codelab

Introduction to Bigtable codelab imports New York city bus data

Generates a heat map of where buses are at a given moment in time

Example of the raw data

	RecordedAtTime	Dire	PublishedLin	OriginName	OriginLat	OriginLong	DestinationN	DestinationL	DestinationL	VehicleRef	VehicleLocat	VehicleLocat
2	12/1/17 0:05	0	B67	MC DONALD	40.63816	-73.978939	DNTWN BKL'	40.700253	-73.98703	NYCT_406	40.671578	-73.977672
3	12/1/17 0:06	1	Bx7	RIVERDALE A	40.912363	-73.902699	WASHINGTC	40.839813	-73.939745	NYCT_4223	40.866243	-73.925258
4	12/1/17 0:05	0	S51	LINCOLN AV/	40.581245	-74.11199	ST GEORGE	40.643585	-74.07261	NYCT_7080	40.581364	-74.112033
5	12/1/17 0:05	0	M8	WEST ST/CH	40.732847	-74.010081	AVENUE "D"	40.724689	-73.974548	NYCT_3809	40.734194	-73.999677
6	12/1/17 0:06	0	M101	ASTOR PL/3	40.729567	-73.990052	FT GEORGE	40.855666	-73.925259	NYCT_5902	40.773784	-73.957659
7	12/1/17 0:06	0	M5	6 AV/W 31 S	40.748042	-73.988957	WASHINGTC	40.848264	-73.937455	NYCT_6365	40.76087	-73.979718
8	12/1/17 0:06	0	B41	E 70 ST/VETI	40.619935	-73.908708	EMPIRE BL	40.663012	-73.962211	NYCT_5045	40.619318	-73.920946
9	12/1/17 0:05	1	S53	4 AV/86 ST	40.622312	-74.028688	PT RICHMON	40.640296	-74.13133	NYCT_8249	40.608292	-74.088821
10	12/1/17 0:06	0	M101	ASTOR PL/3	40.729567	-73.990052	96 ST	40.785724	-73.948791	NYCT_5268	40.730379	-73.989303
11	12/1/17 0:05	1	B6	LIVONIA AV/	40.666382	-73.883617	BENSONHUF	40.592947	-73.993383	NYCT_5090	40.653041	-73.889844



More about the data set

Storage and database services

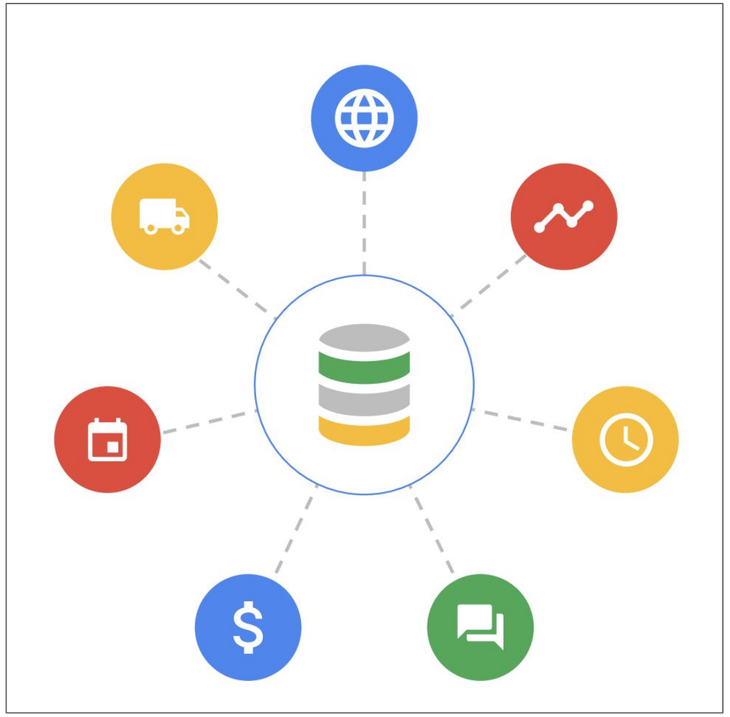
Object	File	Relational			Non-relational		Warehouse
Cloud Storage	Filestore	Cloud SQL	Spanner	AlloyDB	Firestore	Bigtable	BigQuery
Good for: Binary or object data	**Good for:** Network Attached Storage (NAS)	**Good for:** Web frameworks	**Good for:** RDBMS + scale, HA, HTAP	**Good for:** Legacy Applications	**Good for:** Hierarchical, mobile, web	**Good for:** Heavy read + write, events,	**Good for:** Enterprise data warehouse
Such as: Images, media serving, backups	**Such as:** Latency sensitive workloads	**Such as:** CMS, ecommerce	**Such as:** User metadata, Ad/Fin/ MarTech	**Such as:** Transaction Processing, Enterprise workloads	**Such as:** User profiles, game state	**Such as:** Ad tech, financial, IoT	**Such as:** Analytics, dashboards

Next discussion

What is a Data Warehouse?

A data warehouse is a central hub for business data.
Different types of data can be transformed and consolidated into the warehouse for analysis

- Allows data from multiple sources to be combined and analyzed
  - Historical archive of data
- Data sources could be:
  - Relational databases
  - Logs
  - Web data
- Optimized for analytical processing
  - Can handle large amounts of data and complex queries, and is well-suited for reporting and data analysis



Online Transactional Processing

(OLTP) vs Data warehouse (OLAP)

Aspect	Online Transaction Processing (OLTP) Cloud SQL, Spanner	Data Warehouse (BigQuery)
Example Use cases	Point of Sale systems, inventory management, airline reservation systems, hospital medical records	Identify patterns and trends, and develop predictive models to anticipate future events; Develop targeted marketing campaigns to improve customer engagement; Analyze healthcare data to improve patient outcomes and optimize healthcare operations
Purpose	Manage transactional data (inserts, updates, deletes) in real-time	Support analytical queries and reporting based on historical data
Data Volume	Handles relatively small amounts of data in comparison with a data warehouse (with exceptions, e.g., Spanner)	Stores and manages larger volumes of data, often in the terabytes or even petabytes
Data Latency	Data is available immediately after it is entered into the system	May have some latency, since data is extracted from OLTP systems on a regular basis and transformed before being loaded into the data warehouse
Usage	Used by operational staff to support day-to-day business processes	Used by business analysts and decision-makers to perform complex analysis and reporting, such as forecasting, trend analysis, and predictive modeling

BigQuery

Google Cloud data warehouse

Serves as a central hub for storage of (potentially) petabytes of business data

Serverless - compute is automatically provisioned behind the scenes to run SQL queries

Pay only for query processing and data storage, not compute power

Can process massive amounts of data within a few seconds
Not possible to do this with on-premise compute in a cost effective way

[MLB brings AI up to bat with Google Cloud]

[From data warehouse to a unified, AI-ready data platform]

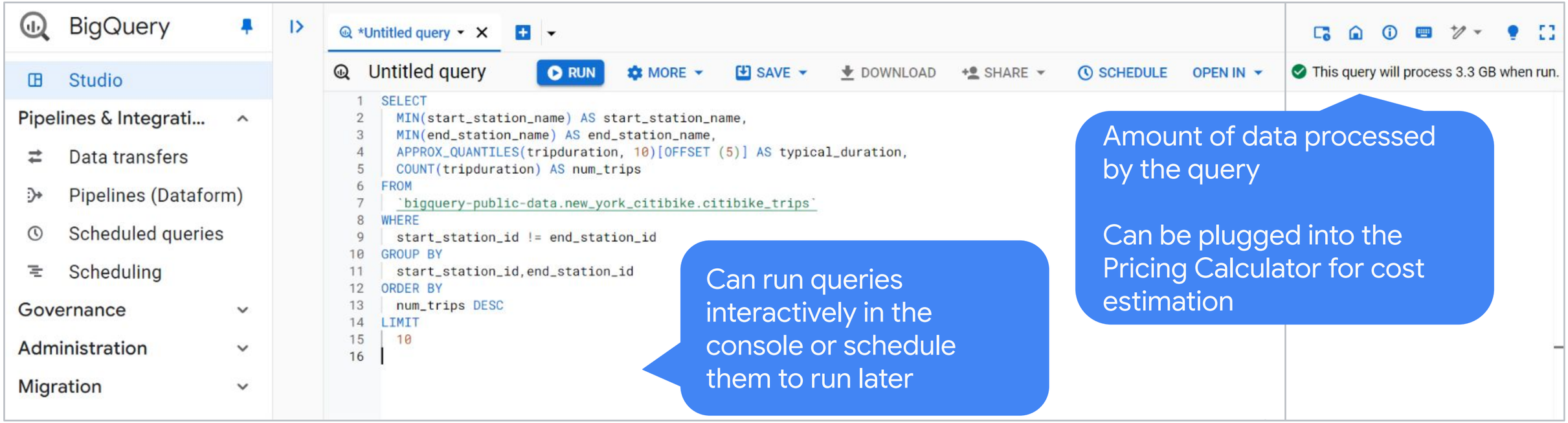
Complex queries run 50 percent faster on BigQuery

BigQuery

Google Cloud data warehouse

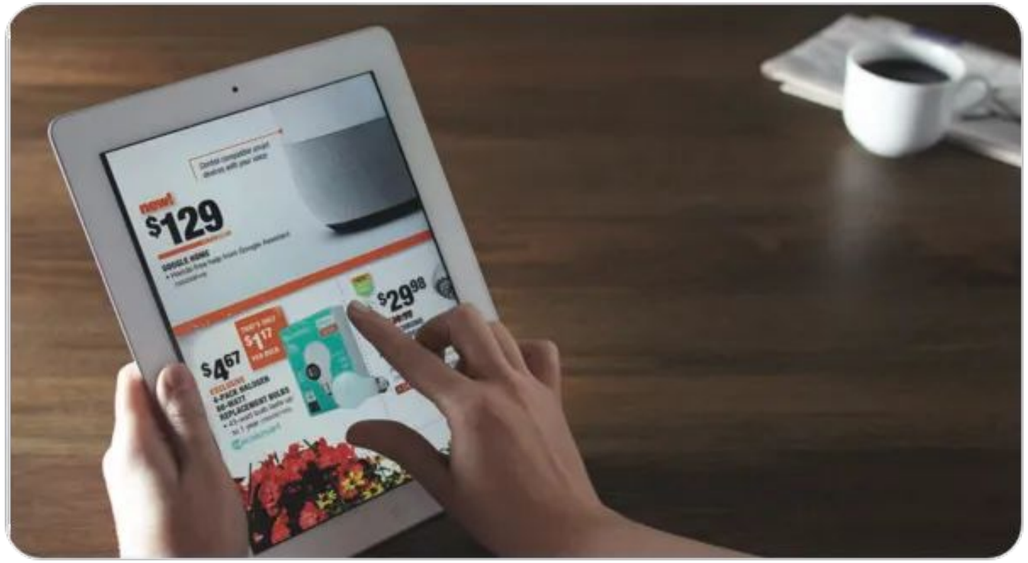
- Multi-cloud capabilities using standard SQL
- Automatic high availability
- Supports federated queries
  - Cloud SQL, Spanner and AlloyDB
  - Bigtable
  - Files in Cloud Storage
- Use cases:
  - Near real-time analytics of streaming data to predict business outcomes with built-in machine learning, geospatial analysis and more
  - Analysis of historical data

BigQuery: Executing queries in the console



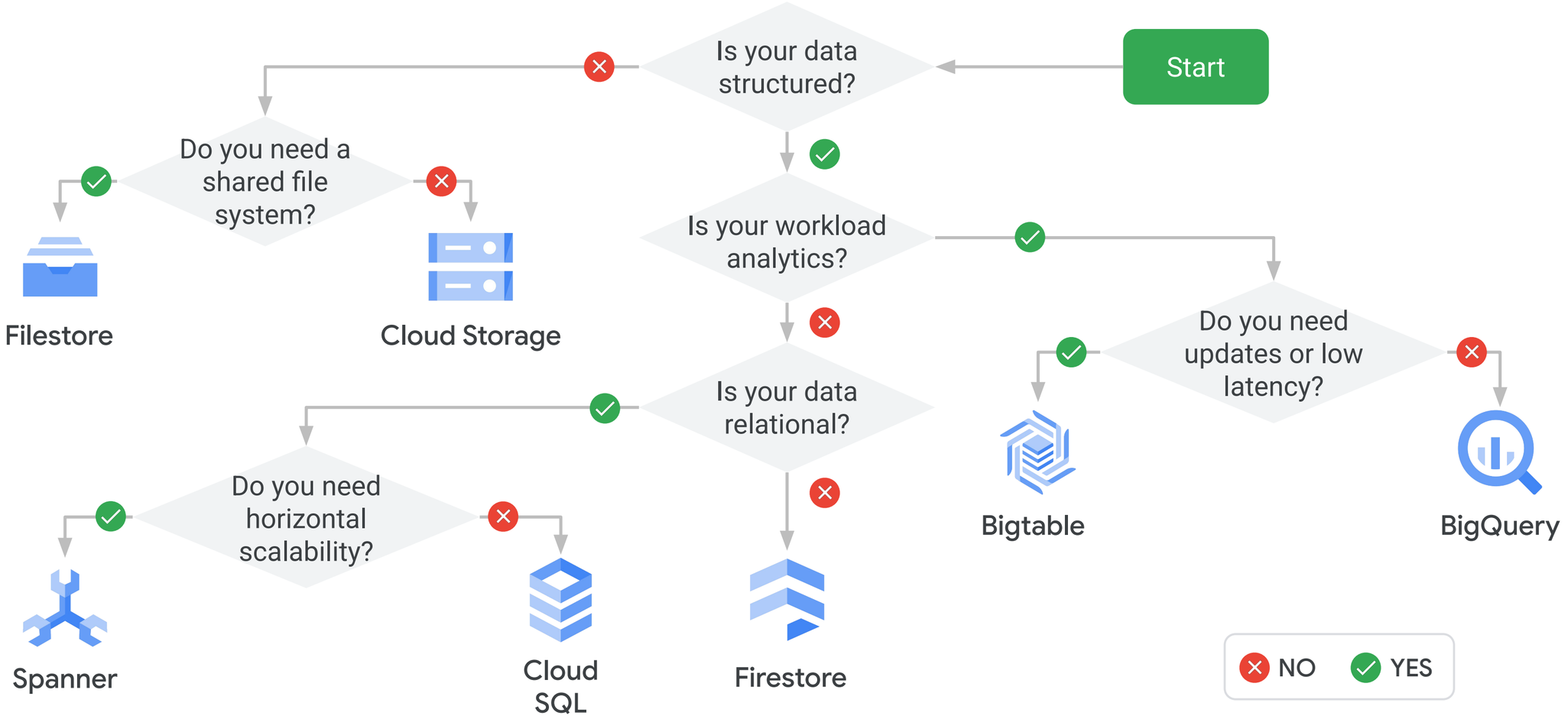
BigQuery - customer use case

- The Home Depot is the world's largest home improvement retailer
  - 2,300 stores in North America + online retail
  - Annual sales > $100 billion
- BigQuery provides timely data to help keep 50,000+ items stocked at over 2,000 locations, to ensure website availability, and provide relevant information through the call center
- No two Home Depots are alike, and the stock in each has to be managed at maximum efficiency.
  - Migrating to Google Cloud, THD's engineers built one of the industry's most efficient stock replenishment systems



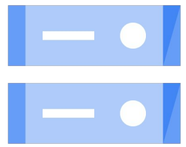
The Home Depot's data-driven focus on customer success

Storage and database decision chart



[Design an optimal storage strategy for your cloud workload]

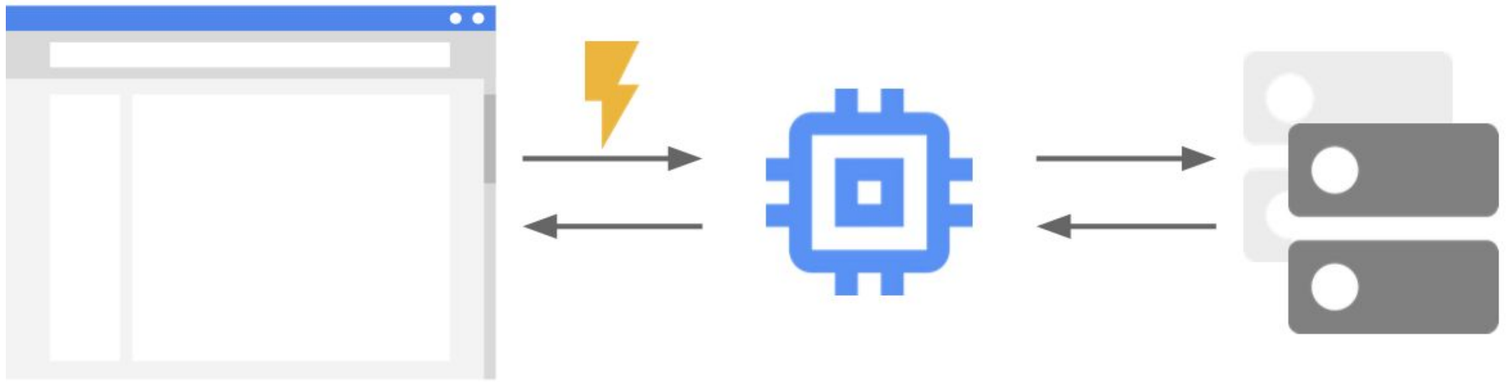
Which database should I use?

Object	File	Relational			Non-relational		Warehouse
							
Cloud Storage	Filestore	Cloud SQL	Spanner	AlloyDB	Firestore	Bigtable	BigQuery
Good for: Binary or object data	**Good for:** Network Attached Storage (NAS)	**Good for:** Web frameworks	**Good for:** RDBMS + scale, HA, HTAP	**Good for:** Legacy Applications	**Good for:** Hierarchical, mobile, web	**Good for:** Heavy read + write, events,	**Good for:** Enterprise data warehouse
Such as: Images, media serving, backups	**Such as:** Latency sensitive workloads	**Such as:** CMS, ecommerce	**Such as:** User metadata, Ad/Fin/ MarTech	**Such as:** Transaction Processing, Enterprise workloads	**Such as:** User profiles, game state	**Such as:** Ad tech, financial, IoT	**Such as:** Analytics, dashboards

Memorystore

- Fully managed implementation of the open source in-memory databases Redis and Memcached
- High availability, failover, patching and monitoring
- Sub-millisecond latency
- Instances up to 300 GB
- Network throughput of 12 Gbps
- Use cases:
  - Lift and shift of Redis, Memcached
  - Anytime need a managed service for cached data

In-memory caching



Application

Web browser

Cache

A caching layer is a high-speed storage layer that stores a subset of data.

Databases

Benefit

- Reduce latency
- Reduce back-end load

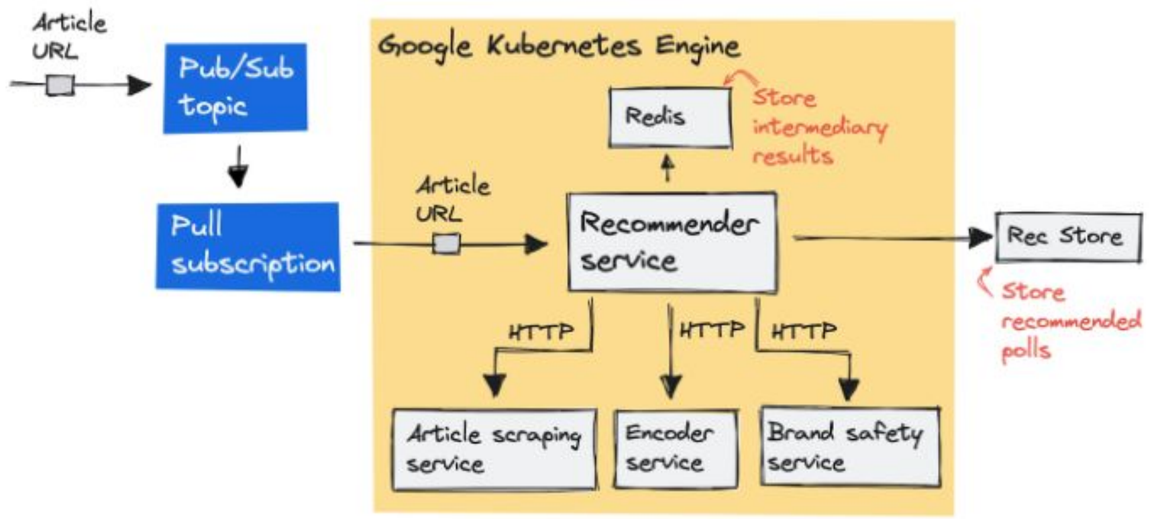
Main use case

- Gaming leaderboard
- Real-time application
- Social Media

```
gcloud redis instances create my-redis-instance
gcloud memcache instances create my-memcache-instance
```

Memorystore - customer use case

- Opinary creates polls that appear alongside news articles on various sites around the world
  - Machine learning is used to decide which poll to display by which article
- The polls let users share their opinion with one click and see how they compare to other readers.
- Publishers benefit by increased reader retention, and increased subscriptions
- Advertisers benefit from high-performing interaction with their audiences



Opinary generates recommendations faster on Cloud Run

Summary: Choosing the correct database

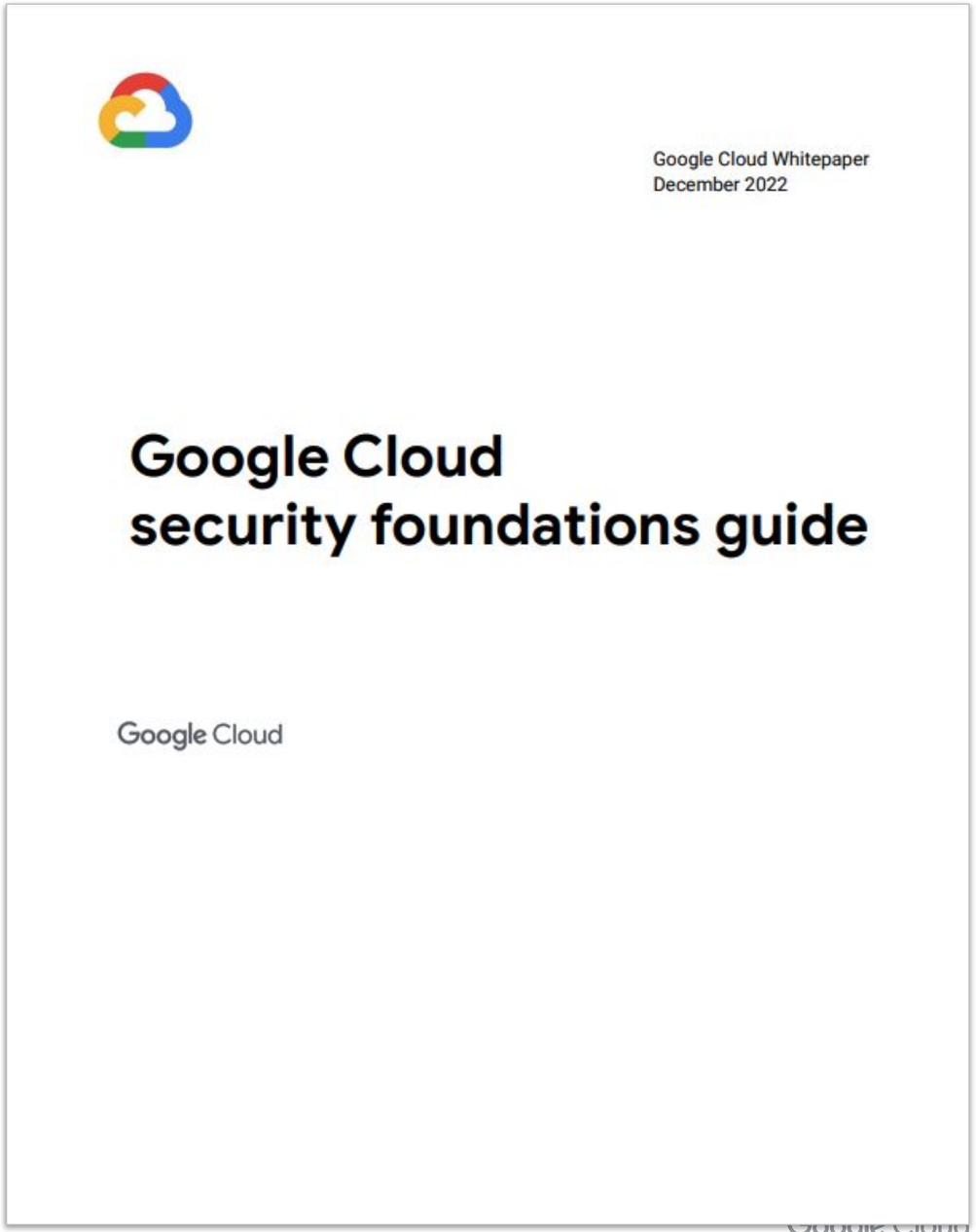
What are the requirements?

- Relational (SQL) vs Non-Relational (NoSQL)
  - Structured vs Unstructured
- Transactional (OLTP) vs Analytical (OLAP)
- Fully Managed vs Requires Provisioning
- Global vs Regional

IAM and Resource Hierarchy

Security Foundations Guide

[Google Security Foundations Guide]



From the exam guide

3.1 Designing for security. Considerations include:

- **Identity and access management (IAM)**
- **Resource hierarchy (organizations, folders, projects)**
- Data security (key management, encryption, secret management)
- **Separation of duties (SoD)**
- Security controls (e.g., auditing, VPC Service Controls, context aware access, **organization policy**)
- Managing customer-managed encryption keys with Cloud Key Management Service
- Remote access

Principle of least privilege

Least privilege is a practice of granting a user only the minimal set of permissions required to perform a duty.

- Users should only be able to do the tasks that are required by their jobs
- This should also apply to machine instances and run-time processes

Separation of duties

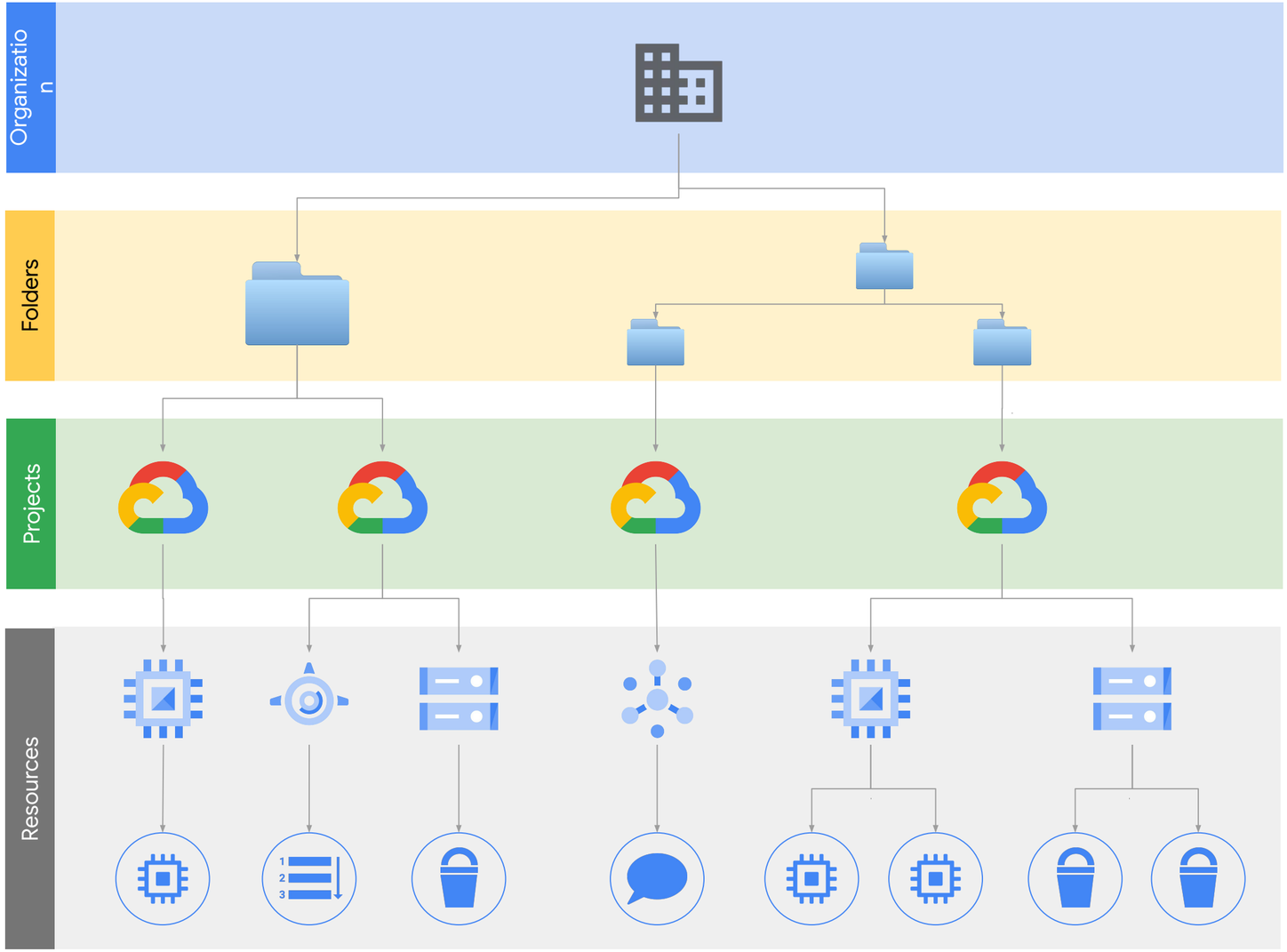
Examples

- No one person can change or delete data without being detected
- No one person is in charge of designing, implementing, and reporting on sensitive systems
- The people who write the code shouldn't deploy the code; those who deploy the code shouldn't be able to change it

Implementation Suggestions

- Use multiple projects to separate duties
- Different people can be given different rights in different projects
- Use folders to help organize projects

Google Cloud Resource Hierarchy



Organization

Folders

Projects

Resources

Organization and Folders

Organization is a registered domain

Rights can be granted at the organization level

Folders can be added to the organization

Rights can be granted to folders

Folders can contain other folders

Projects are added to either the organization or to a folder

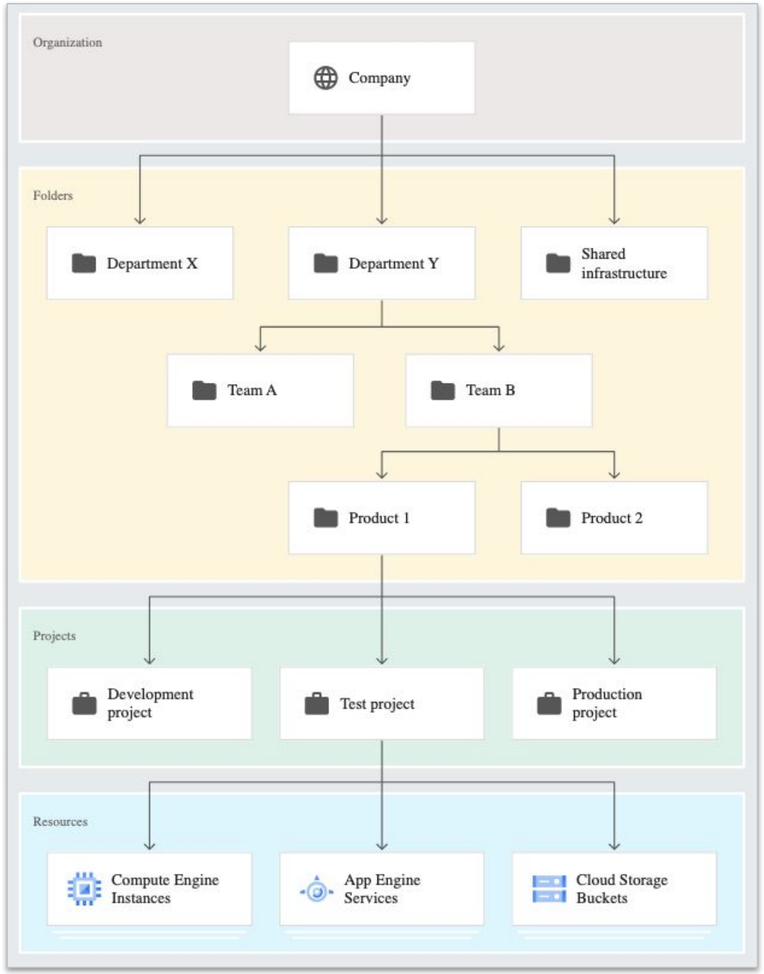
Rights can be granted at the project level

Resources are added to projects

Rights can be granted at the resource level

Folders provide a logical way to organize teams and projects

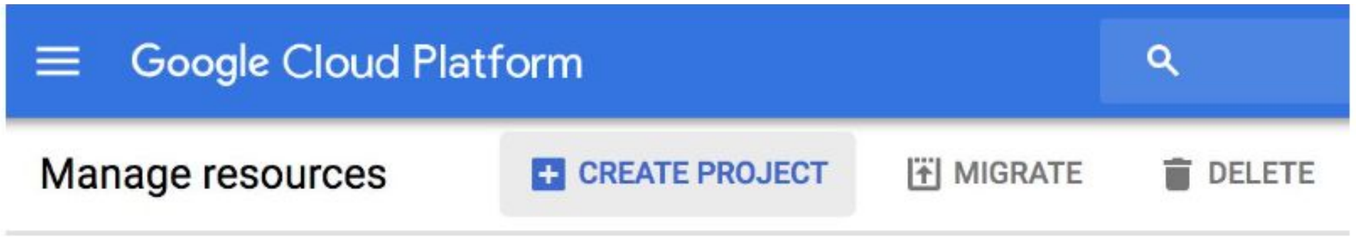
- Must have an Organization node in order to create folders
- Folders contain projects, other folders, or both
- Roles and Organization policies can be applied at the folder level (and also at the project/organization level)



All products & services are associated with a project

- Enable/disable APIs
- Enabling billing and track resource usage
- Contains resources such as VPC networks, VMs, Databases, etc.
- IAM permissions can be applied at this level

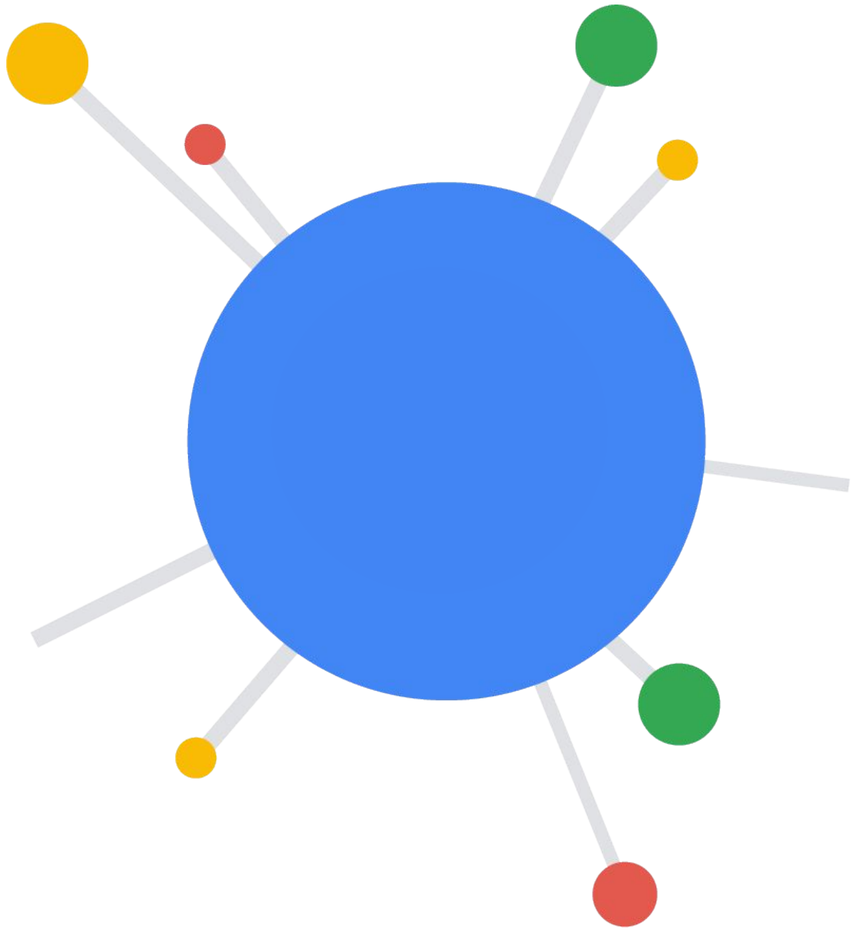
Creating a project



Project ID	Globally unique	Chosen by you	Immutable
Project name	Need not be unique	Chosen by you	Mutable
Project number	Globally unique	Assigned by Google Cloud	Immutable

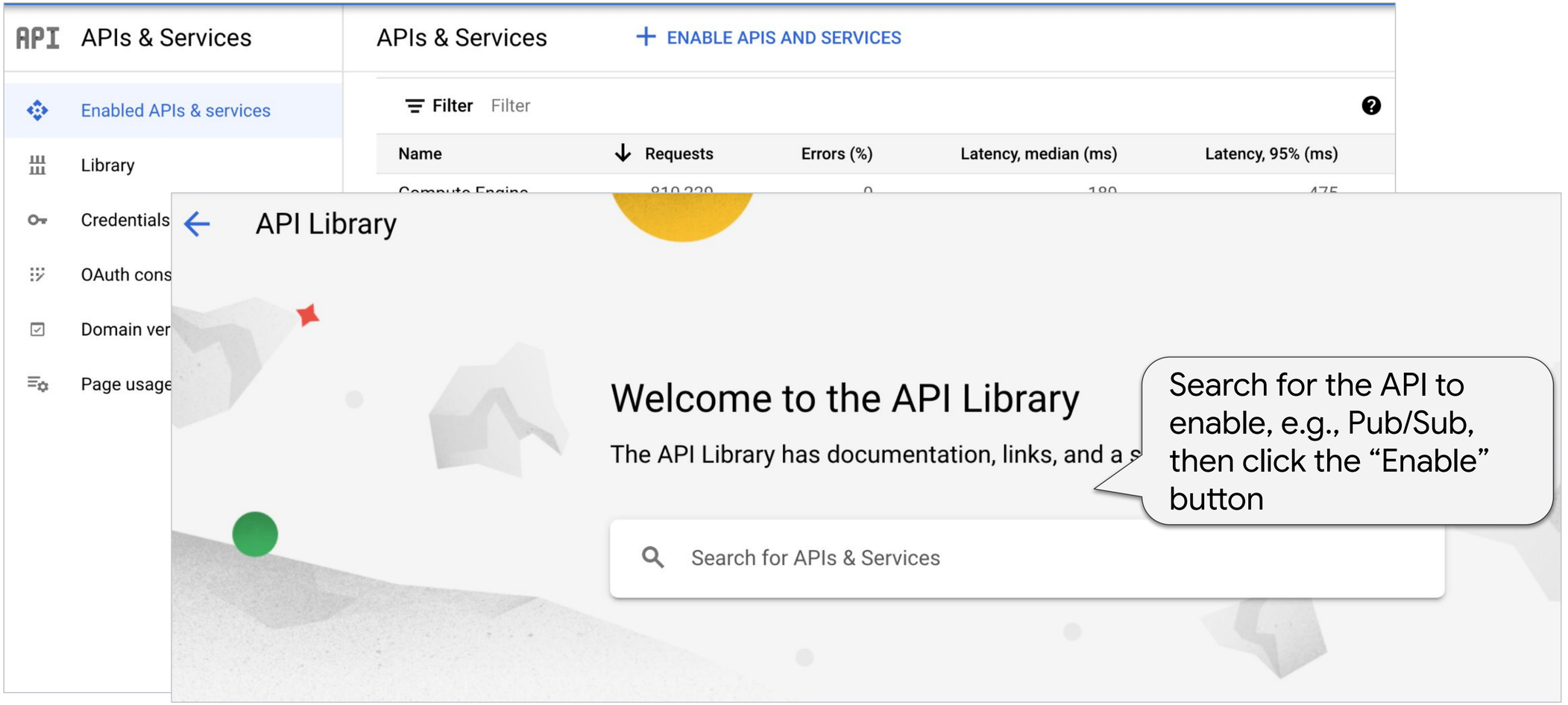
Resources

- Resources are anything created when using services
- All resources are associated with a project
- Examples of resources are:
  - Virtual machines
  - Persistent disk
  - Cloud Storage buckets
  - BigQuery datasets and tables
  - Spanner databases
  - Kubernetes Engine clusters
- Must enable service-specific APIs before creating resources within a project



Enabling APIs - Console & Command Line

```
gcloud services enable pubsub.googleapis.com
```

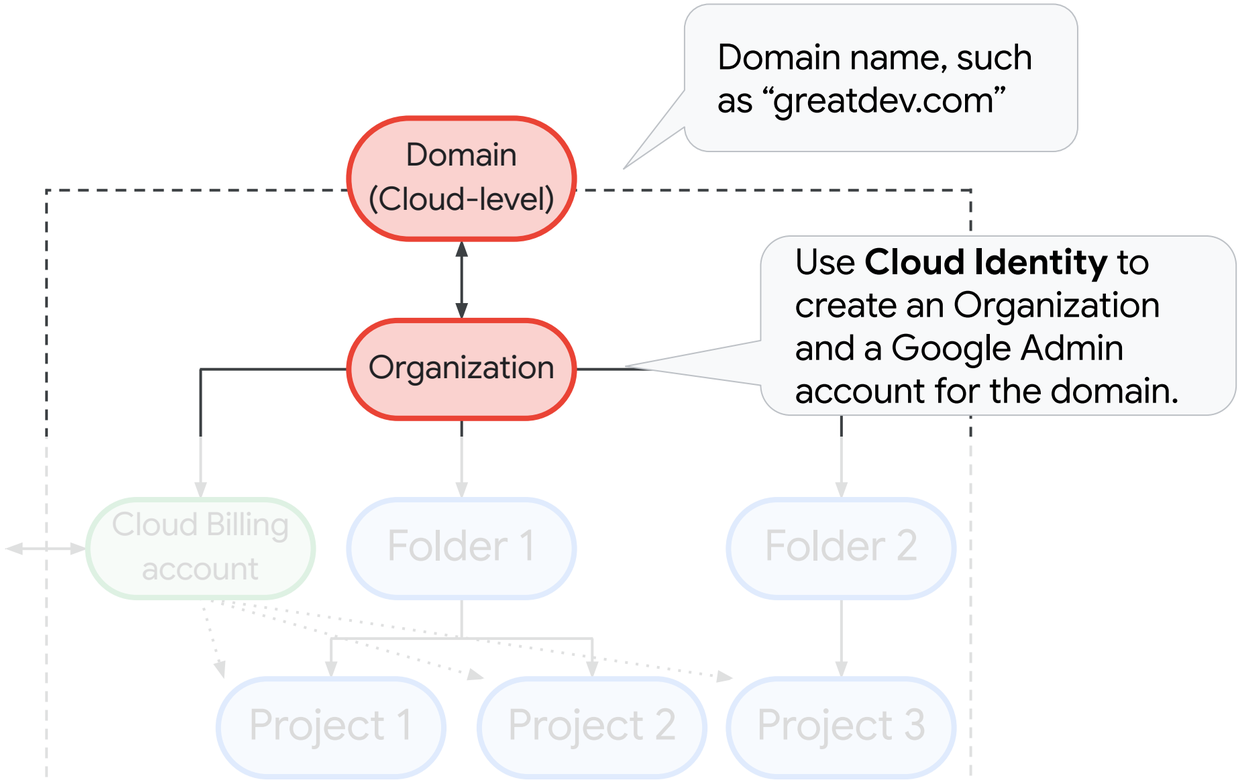


How is an Organization Created?

Cloud Identity manages the users and groups that have access to Google Cloud

Federated identities from Google Workspace and other identity providers, such as Active Directory and Azure Active Directory

- Bring existing users/groups into Cloud Identity
- Use Identity and Access Management (IAM) to manage access to Google Cloud resources



Cloud Identity

Centrally manages users and groups for Google Workspace and Google Cloud

Google's Identity as a Service (IDaaS) solution

- Users and groups that are to be added to Google Cloud need accounts in Cloud Identity

Manually creating user accounts

- Add users individually using the Google Admin console
- Add several users at once by uploading their names in a CSV file

Focus of this module

Cloud Identity

Centrally manages users and groups for Google Workspace and Google Cloud

Options for large organizations

- Use Google Cloud Directory Sync to synchronize user data in your existing LDAP directory with your Google account
- Use the Admin SDK Directory API to provision a large number of users with data from your existing LDAP directory, such as Microsoft® Active Directory®
  - Requires programming

*Cloud Identity has advanced management features not covered in this module, e.g., mobile app management, 2-Step verification, etc.

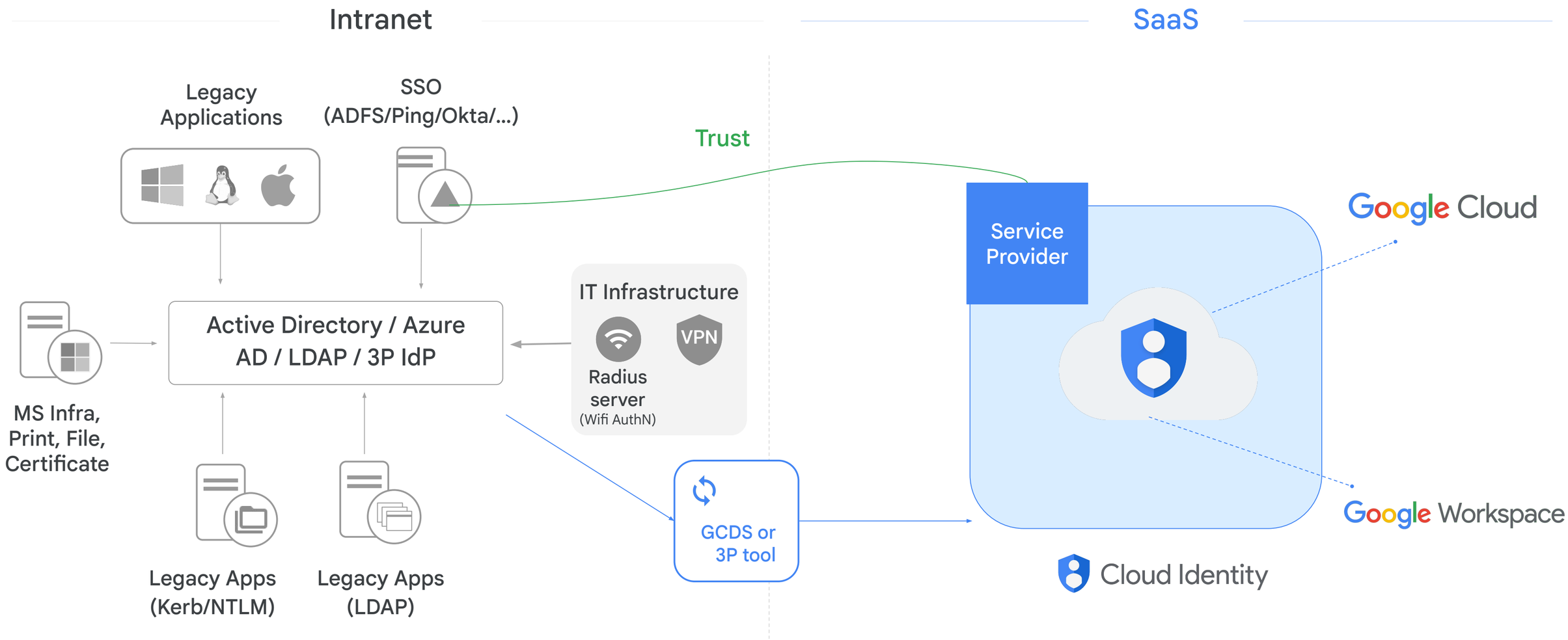
Google Cloud Directory Sync (GCDS)

- One-way synchronization of corporate data (no writing to LDAP system)
- Only synchronizes deltas for fastest possible provisioning
- Syncs all object types (users, aliases, profiles, groups)
- Utilizes Google APIs to provision all object types, the same APIs available to customers

Best practice

Provision from a single source of truth

Third-party as an identity provider: Typical architecture



Controlling access

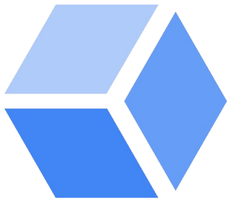
Authentication

Cloud Identity

Authorization

Identity Access Management (IAM)

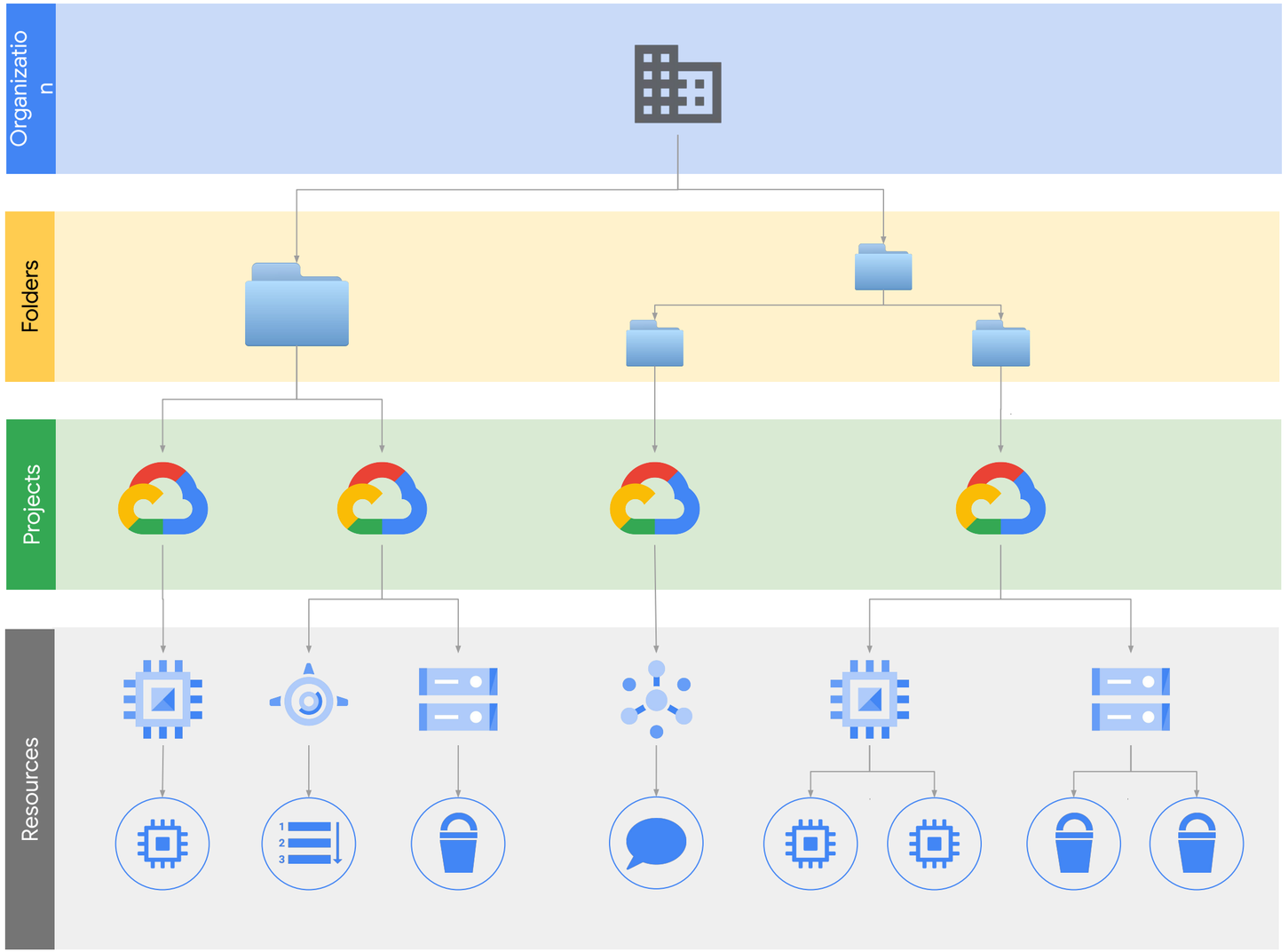
Auditing



Google Cloud's Operations suite

Cloud Audit Logs & Reports API

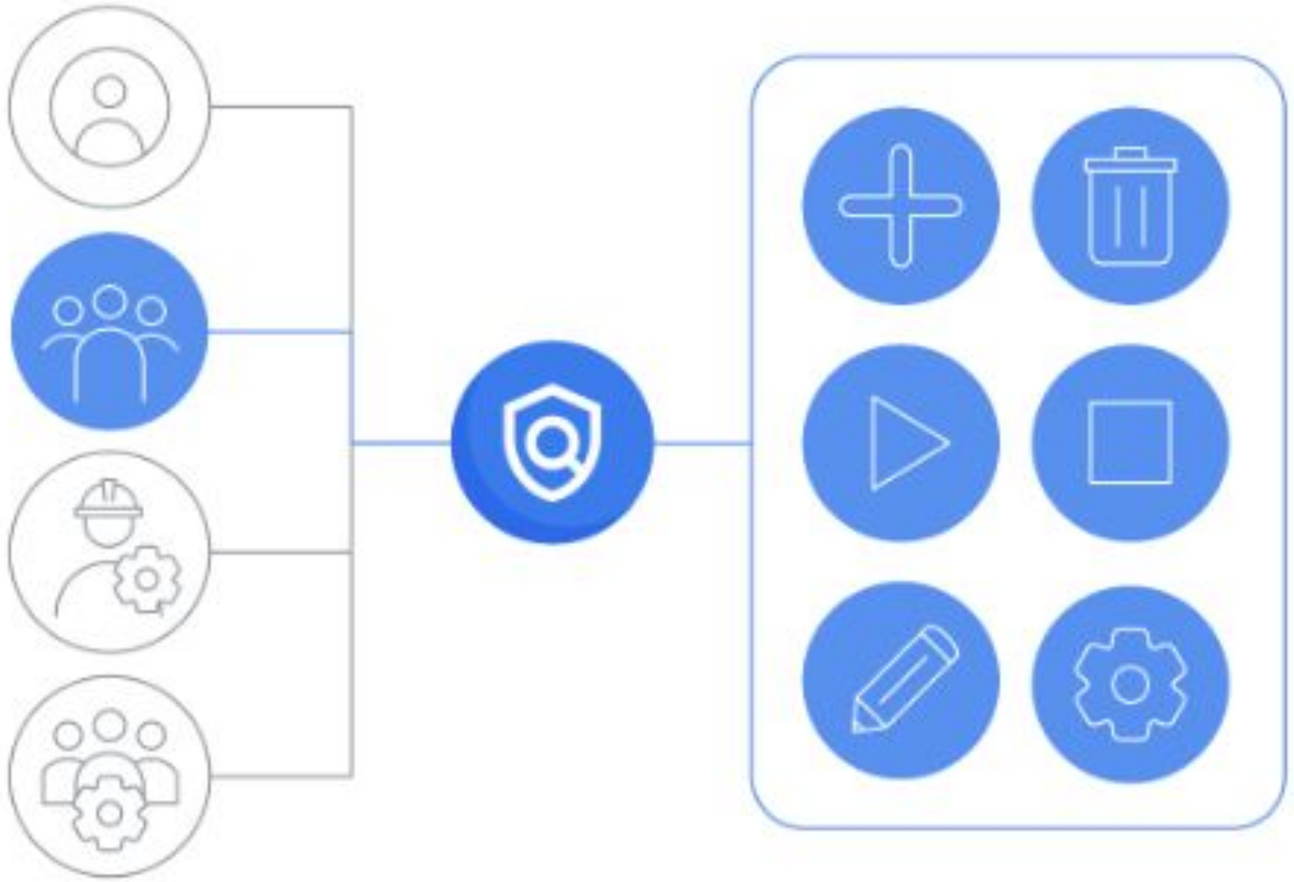
Hierarchy inheritance



Next discussion:
Identity and Access
Management

Identity and Access Management (IAM) applies policies

Administrators can apply policies that define who can do what on which resources



Who ...?

The “who” part of an IAM policy can be a

- Google account
- Google group
- Service account
- Google Workspace or Cloud Identity domain

Service Accounts are discussed later

Also

- allAuthenticatedUsers
- allUsers

… can do what …?

The “can do what” is defined by an IAM role.

Three kinds of IAM Roles

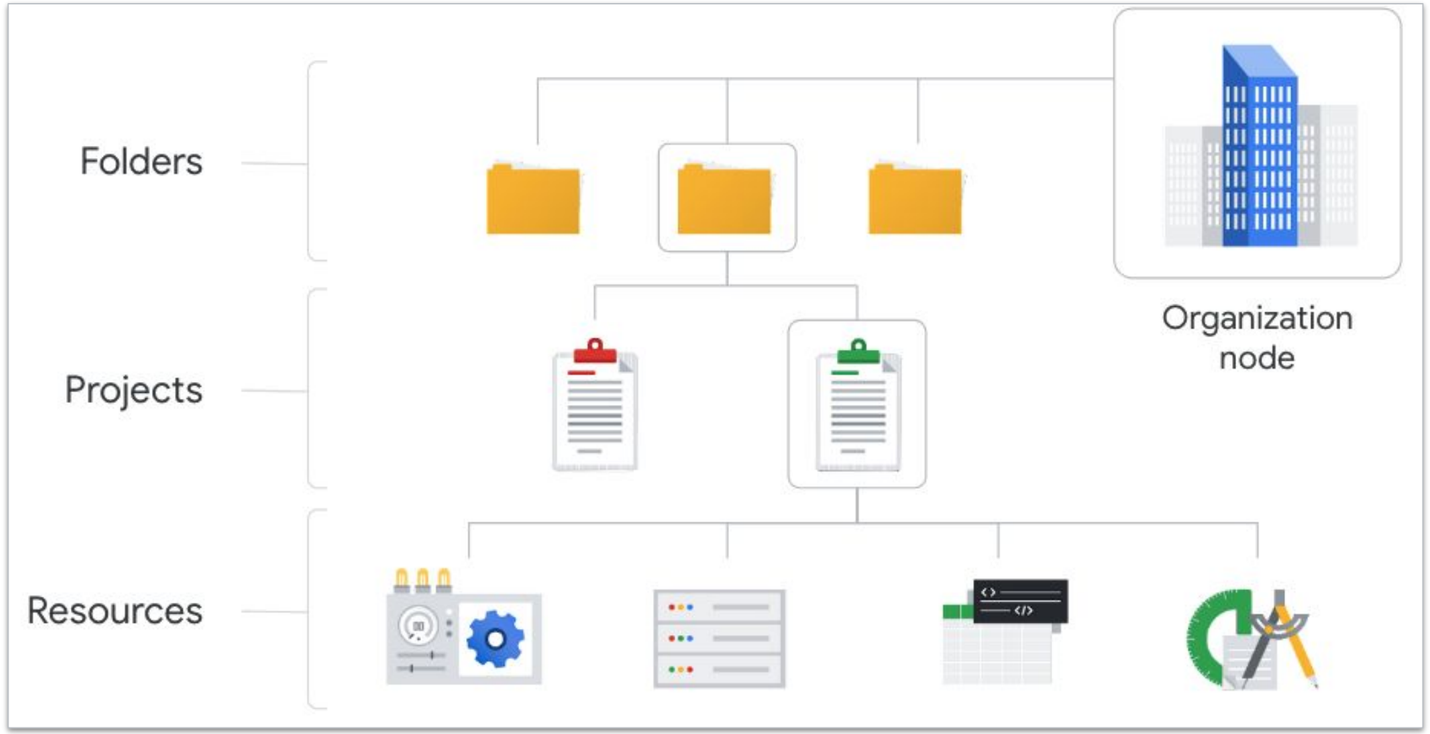
Basic IAM role

Predefined IAM role

Custom IAM role

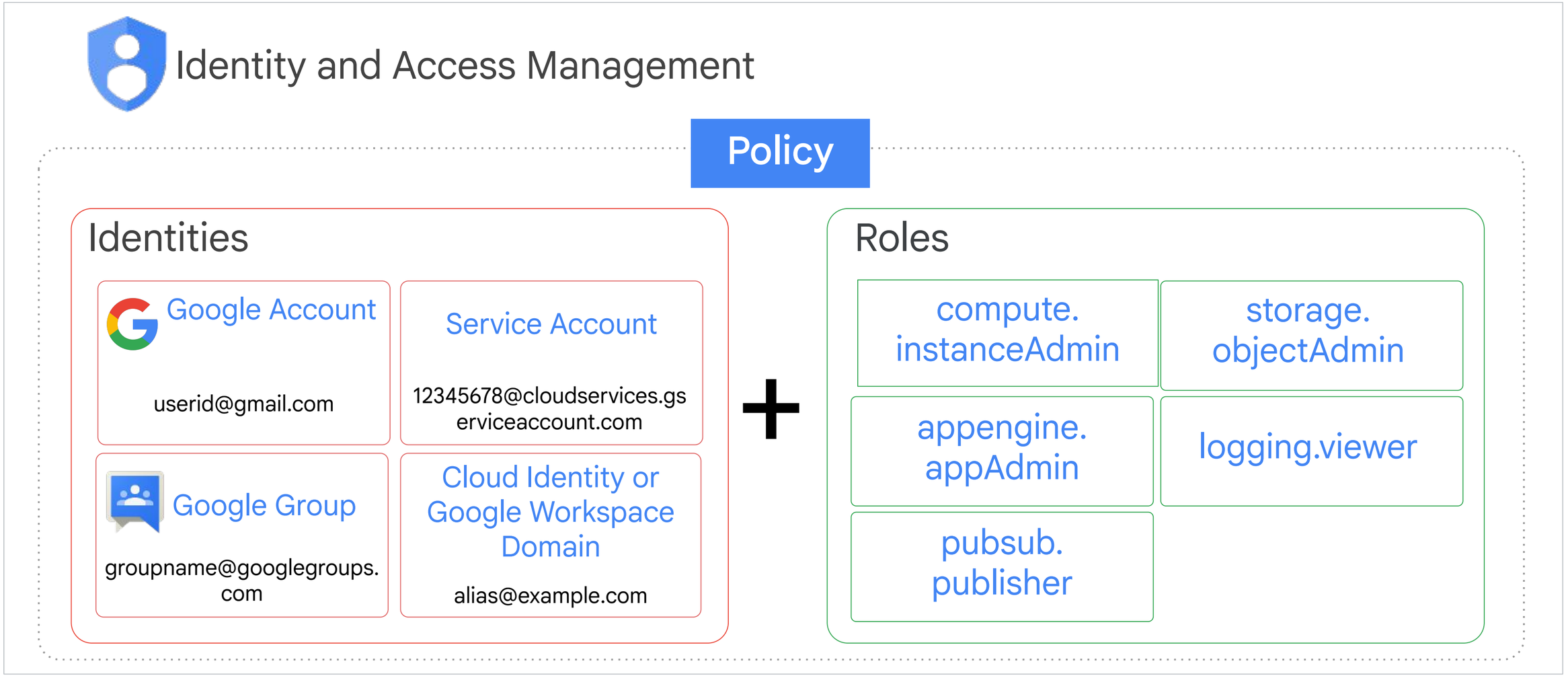
… on which resource?

Can manage IAM at the project level, folder level or organization level*



*Note: Some Cloud Storage IAM is applied at the bucket level

A policy is a combination of principals + assigned roles



Also allUsers and allAuthenticatedUsers

Note: You *don't* use IAM to create or manage your users or groups.

… can do what …?

The “can do what” is defined by an IAM role.

Revisiting this slide

Basic roles discussed next

Three kinds of IAM Roles

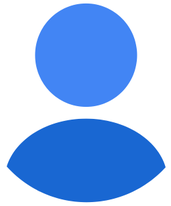
Basic IAM role

Predefined IAM role

Custom IAM role

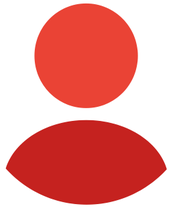
IAM basic roles

Offer fixed, coarse-grained levels of access that are broad in scope



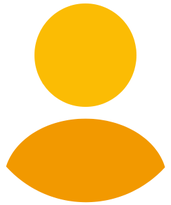
Owner

- Invite members
- Remove members
- Delete projects
- And...



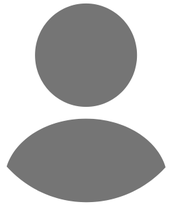
Editor

- Deploy applications
- Modify code
- Configure services
- And...



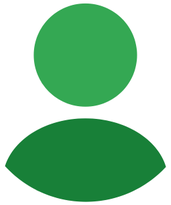
Viewer

- Read-only access



Browser

- View resource hierarchy

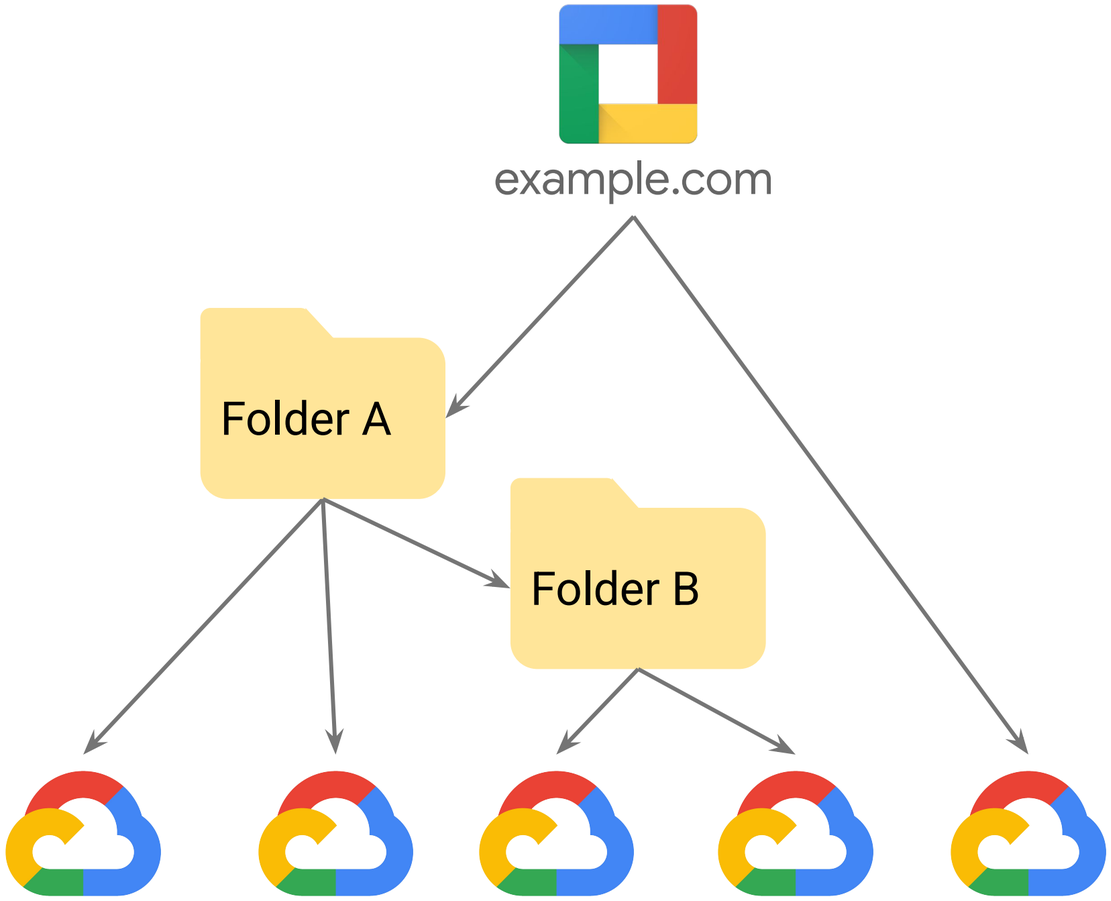


Billing Administrator

- Manage billing
- Add and remove billing administrators

The browser basic role

This role provides read access to browse the hierarchy for a project, including the organization and folders.



… can do what …?

The “can do what” is defined by an IAM role.

Revisiting this slide

Three kinds of IAM Roles

Basic IAM role

Predefined IAM role

Custom IAM role

Predefined roles discussed next

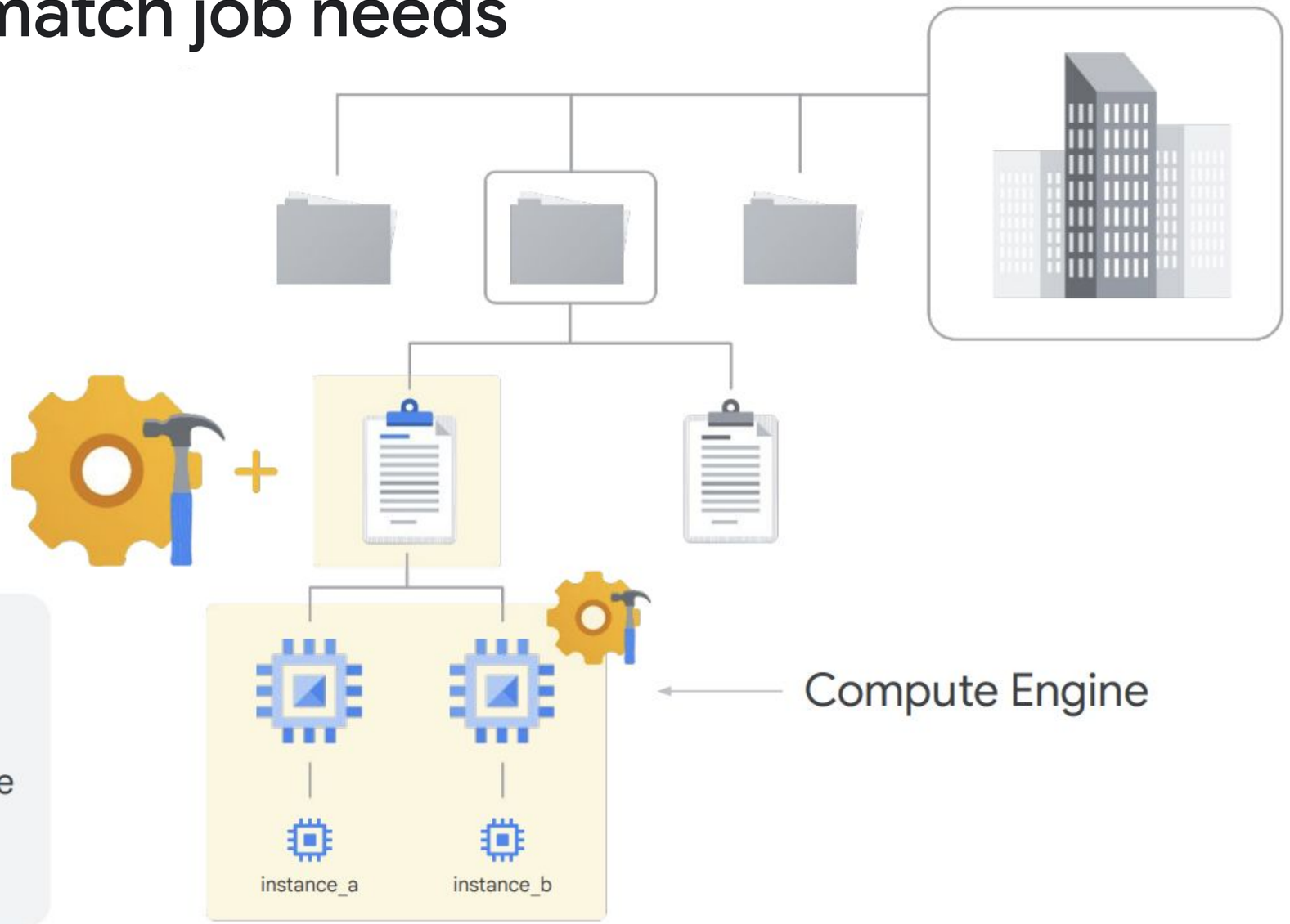
Predefined Roles match job needs

Predefined
IAM role

instanceAdmin

predefined actions

compute.instances.delete
compute.instances.get
compute.instances.list
compute.instances.setMachineType
compute.instances.start
compute.instances.stop



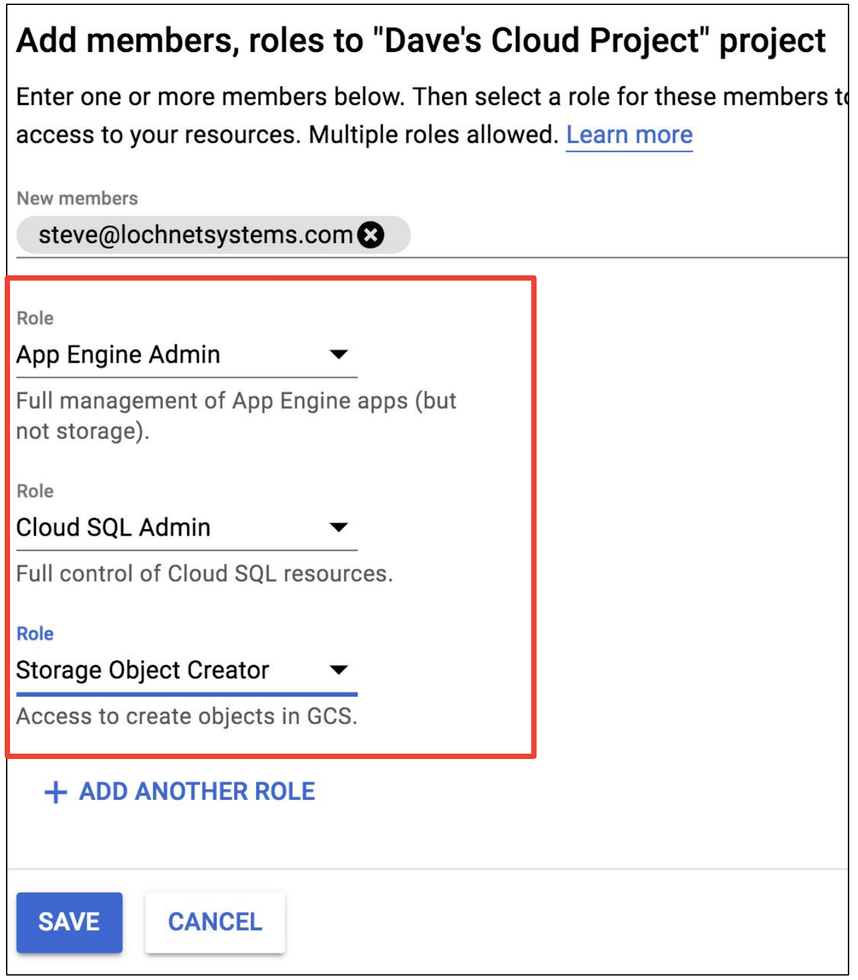
Using predefined roles

Google-created roles for each service

- Permissions defined for different job roles working with each service
- Permissions maintained by Google

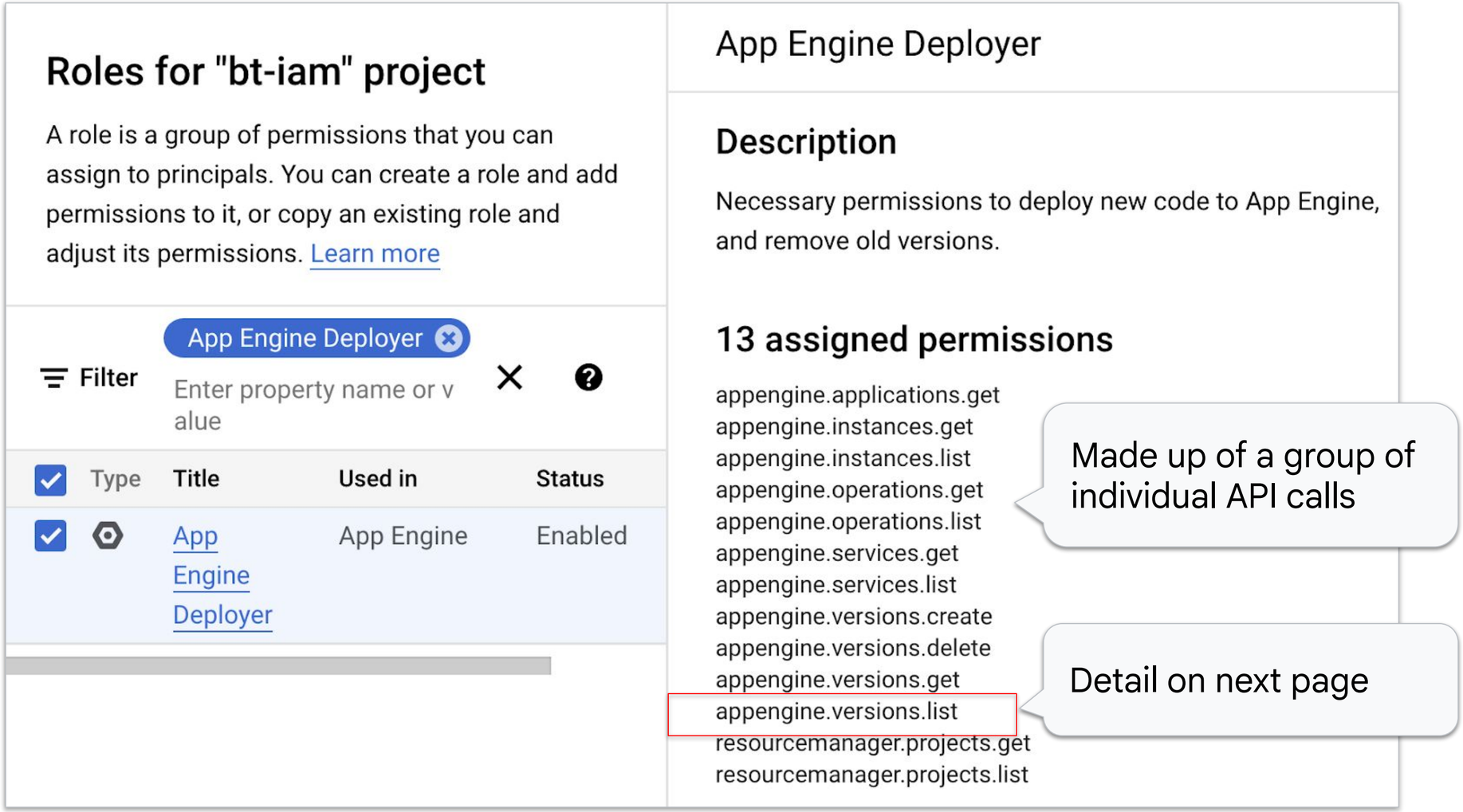
Best practice is to avoid the basic roles and use predefined roles when adding members or service accounts

- Principle of least privilege

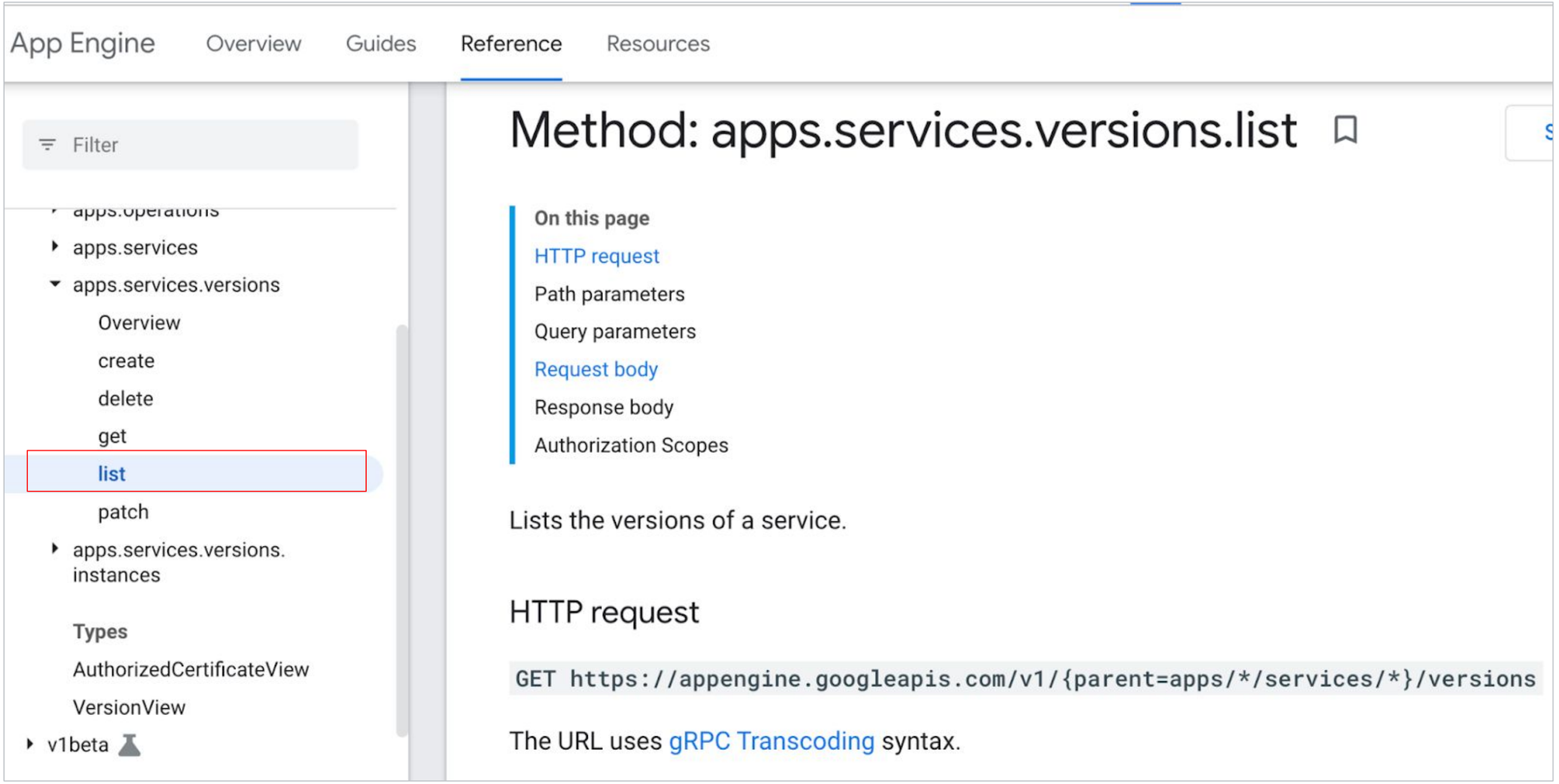


Role permissions

Cloud Console > IAM & Admin > Roles



App Engine API Documentation



[Method: apps.services.versions.list]

… can do what …?

The “can do what” is defined by an IAM role.

Revisiting this slide

Three kinds of IAM Roles

Basic IAM role

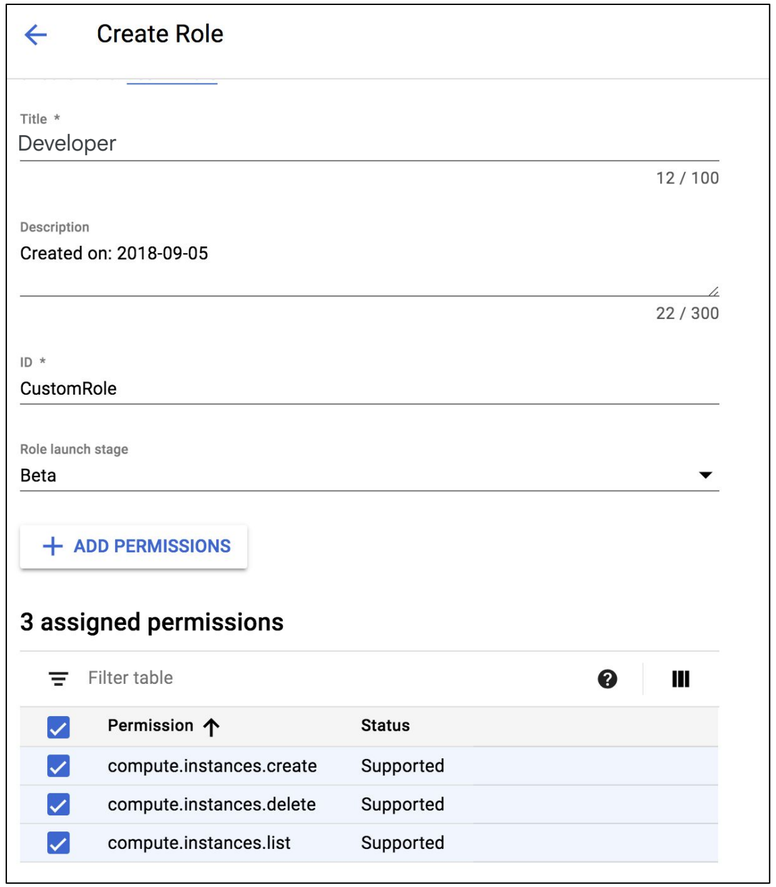
Predefined IAM role

Custom IAM role

Custom roles discussed next

Custom Roles

- Roles that you create
  - Fine-grained control over permissions
- Can add any permissions you like
- Can create custom roles based on predefined roles and add /remove permissions
- Custom roles add operational overhead
  - You must maintain the permissions
- Applied at the project or organization level



Deny Policies

- Introduced in 2022
- Inherited through the resource hierarchy just like IAM allow policies
- Attached to project, folder or organization
- Denies override grants further down the hierarchy
- Currently, must be created via command line

First create a deny policy and store it in a file

People in the dev@example.com group are not allowed to create or delete service account keys

```
{
  "deniedPrincipals": [
    "principalSet://goog/group/dev@example.com"
  ],
  "deniedPermissions": [
    "iam.googleapis.com/serviceAccountKeys.create",
    "Iam.googleapis.com/serviceAccountKeys.delete"
  ]
}
```

Next apply the deny policy

```
gcloud iam policies create POLICY_ID \
    --attachment-point=[proj-id|folder-id|org-id] \
    --kind=denypolicies \
    --policy-file=POLICY_FILE
```

Who ...?

Saw this slide earlier

Next topic: Service accounts

The “who” part of an IAM policy can be a

- Google account
- Google group
- **Service account**
- Google Workspace or Cloud Identity dom

Also

- allAuthenticatedUsers
- allUsers

There are three types of Service Accounts

Google-managed SAs

- Created and managed by Google
- Used to allow services to access resources
  - E.g., You create a Pub/Sub *push* message to trigger a Cloud Run service
  - Google create a SA with `run.invoker` IAM
- Do not modify these

Default SAs

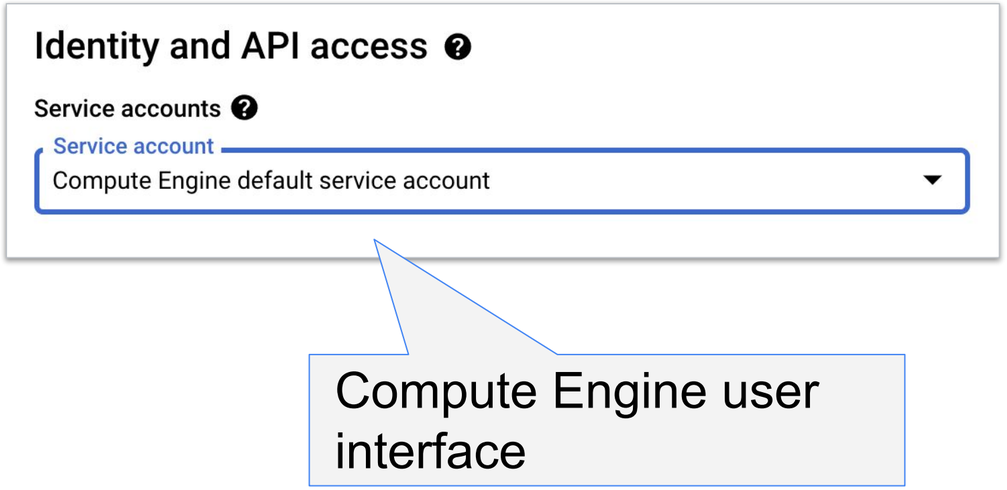
- Google-created but user-managed SAs
- Created when certain APIs are enabled, e.g., Compute Engine, Cloud Run

User-managed SAs

- Created and managed by someone with **Service Account Admin** or **Owner** IAM
  - Are given various IAM roles depending on the use case
- Attached to services

Service accounts are identities for Google Cloud services

- Special type of account intended to represent a non-human user that needs to authenticate and be authorized to access data
  - Are created *within* Google Cloud, unlike other Principals
- Attached to VMs or other services
  - Applications will only be allowed to perform actions allowed by the roles given to the service account



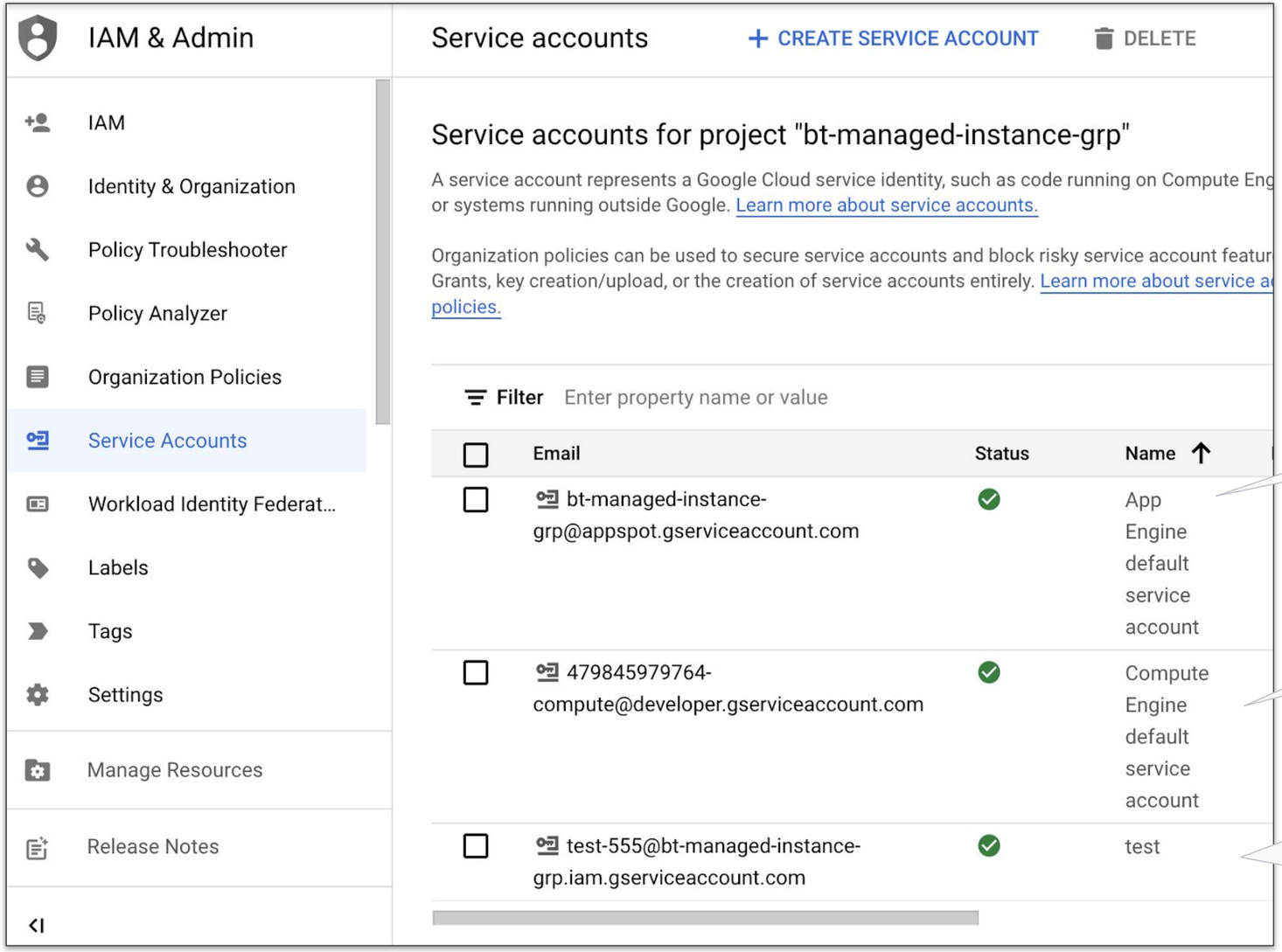
[Understanding Service Accounts]

Service account use cases

Typically, service accounts are used in scenarios such as:

- Running workloads on virtual machines (VMs)
  - E.g., Create a service account with permissions to query BigQuery
    - Attach it to a VM
    - Deploy an application onto the VM that submits SQL commands to BigQuery
- Running workloads on on-premises workstations or data centers that call Google APIs.
  - E.g., Same example as above, but now the application is running on-premise
- Running workloads which are not tied to the lifecycle of a human user.
  - E.g., Batch jobs that are scheduled to run periodically

Viewing Service Accounts in the Console

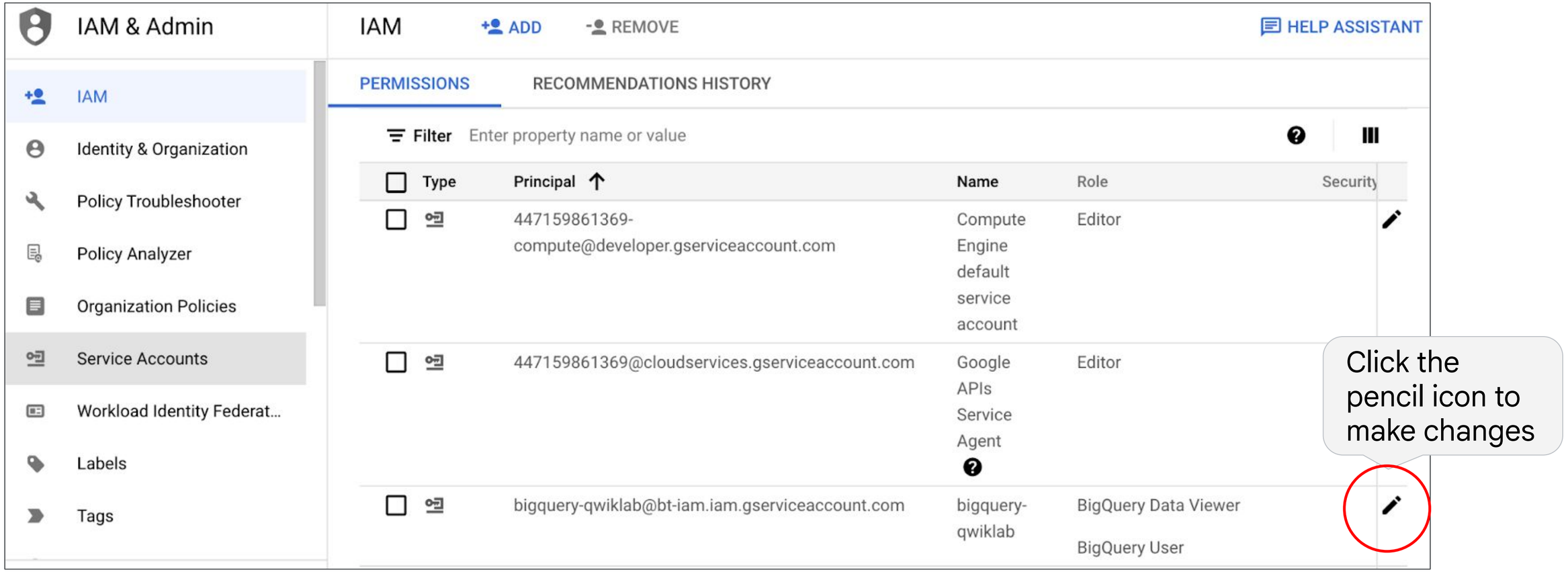


Created by Google when App Engine API is enabled

Created by Google when Compute Engine API is enabled

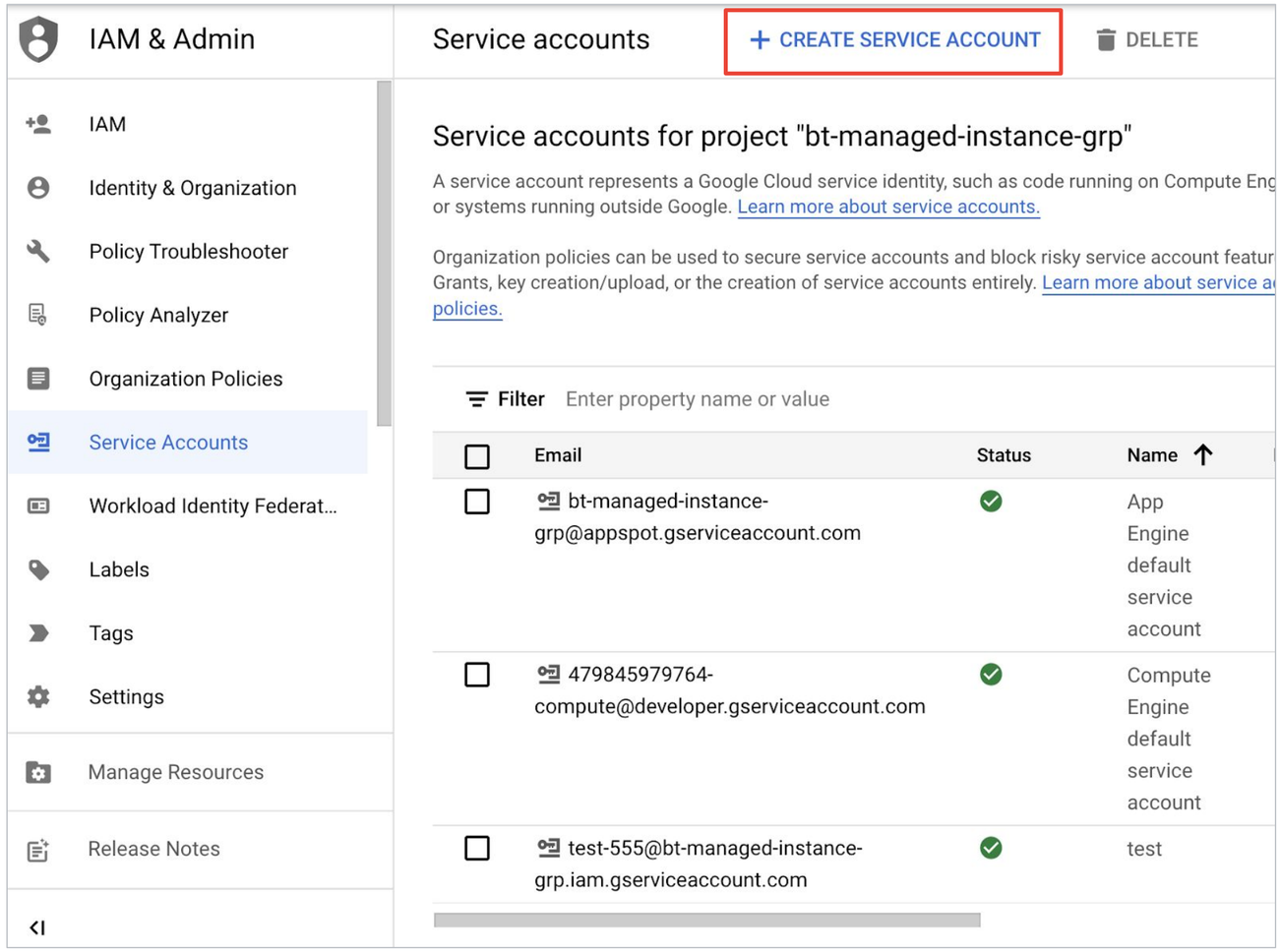
Custom service account. Individuals need Service Account Admin or Editor role to create them.

Viewing Service Accounts IAM in the Console

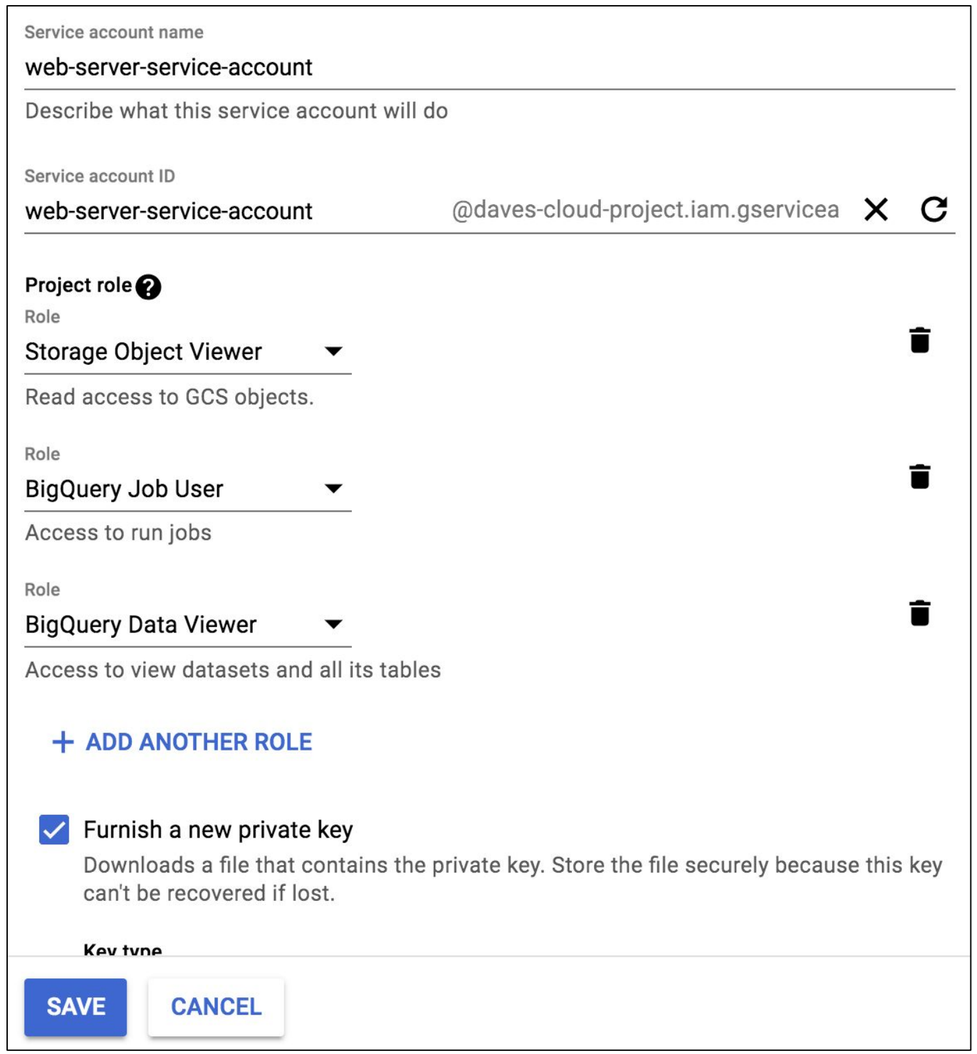


Creating Service Accounts in the Console

IAM & Admin > Service Accounts> Create Service account



Creating Service Accounts (continued)



Service account name

Add one or more roles

Attaching service accounts to resources

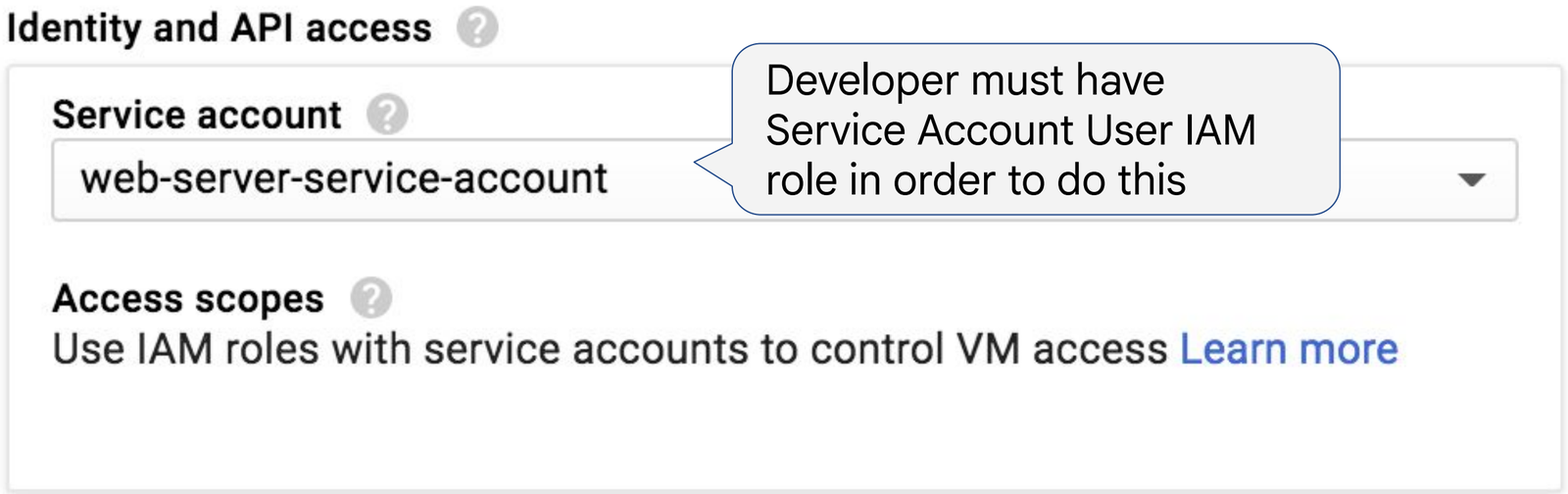
Service accounts can be attached to multiple services in Google Cloud, including

- Compute Engine (and Kubernetes Engine)
  - Default service account is created by Google when API is enabled
- App Engine
  - Default service account is created by Google when API is enabled
- Cloud Run
  - Uses the Compute Engine default service account
- Cloud Functions
  - Uses the App Engine default service account

Best practice: Create custom service accounts for all services

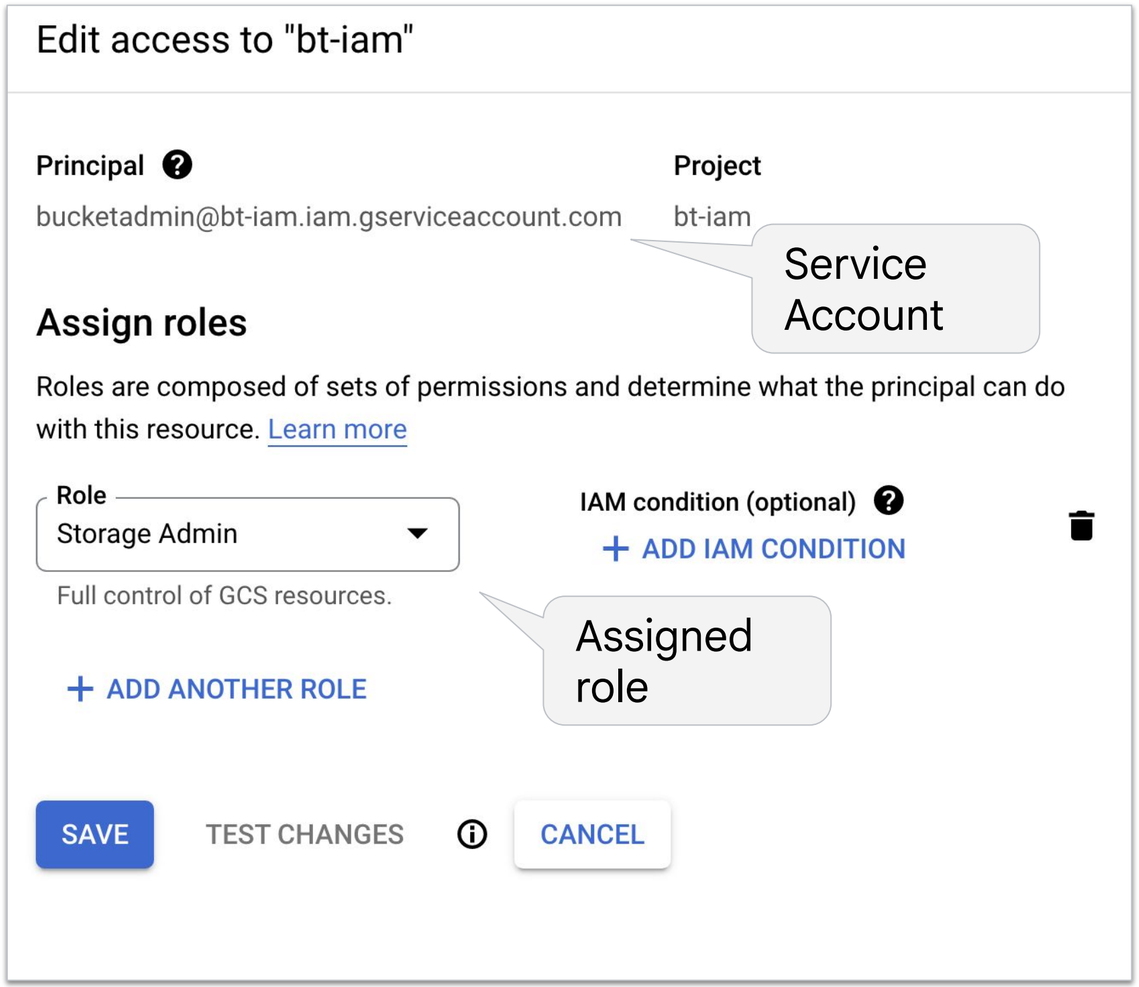
Example: Assigning Service Accounts to VMs

- Default service account is a project editor which requires scopes to control what the machine can do
- Best practice is to create a custom service account



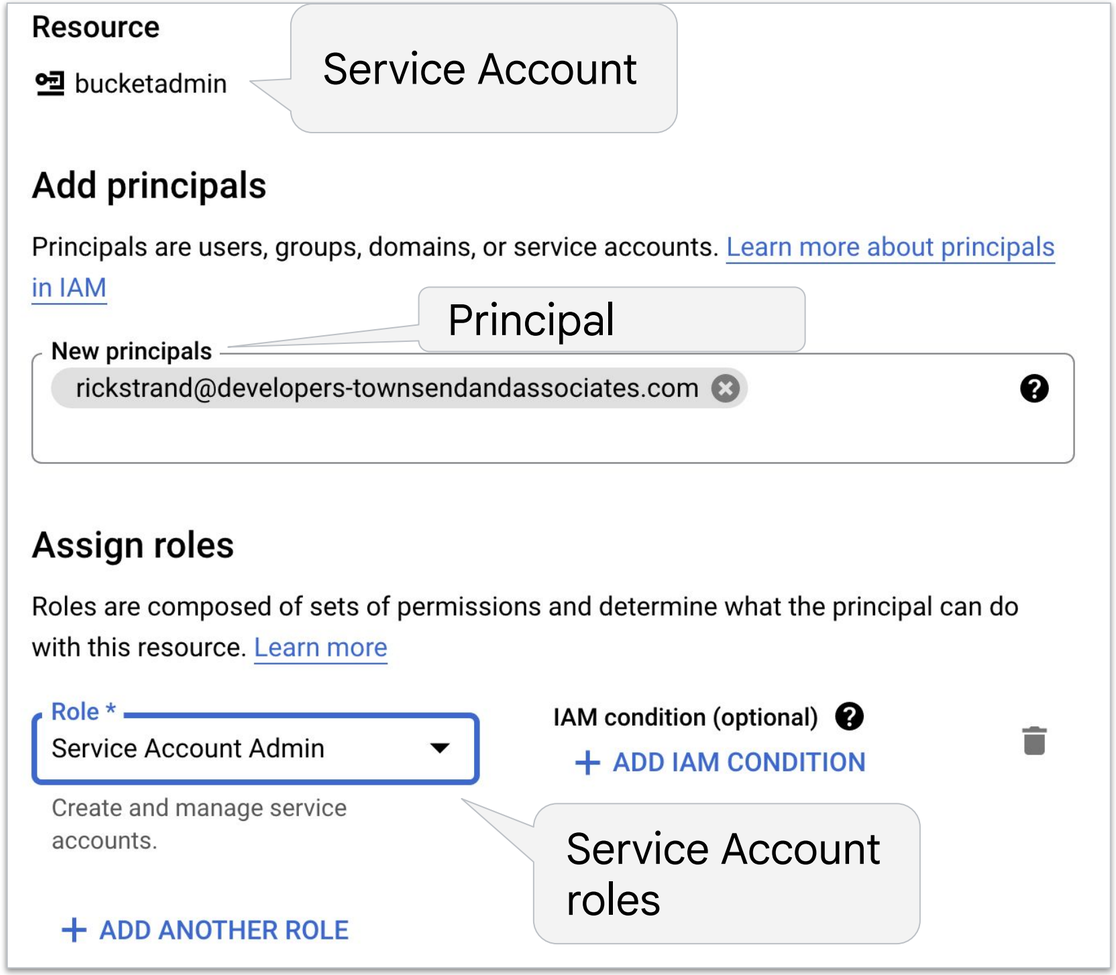
Service accounts are both principals and resources

Is a principal when roles are assigned to it



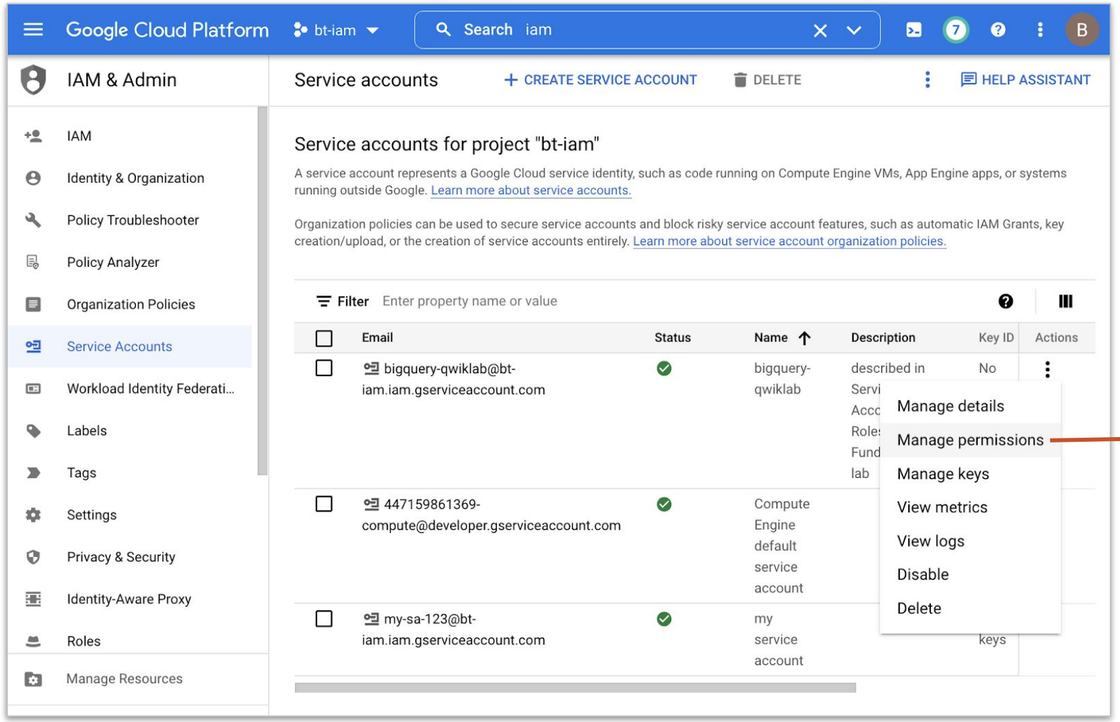
Service accounts are both principals and resources

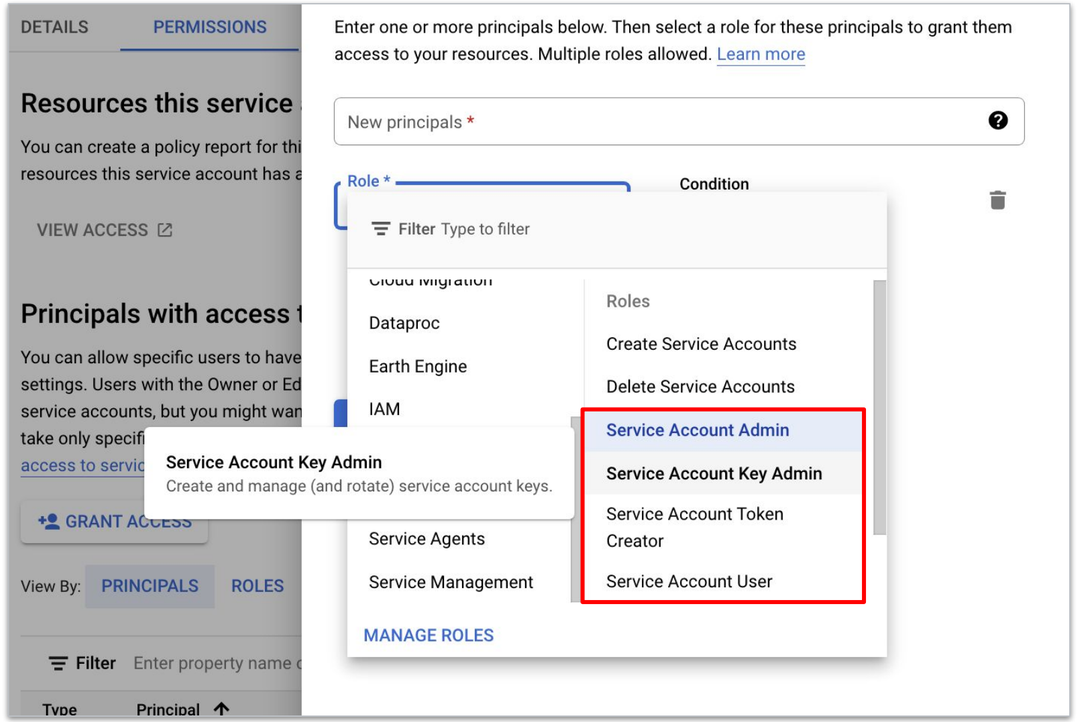
Is a resource when users (principals) are given Service Account IAM roles to manage the service account in some way



Managing service account role assignments

Console > IAM & Admin > Service Accounts > Manage Permissions





Service Account predefined roles (not a complete list)

Assigned to principles

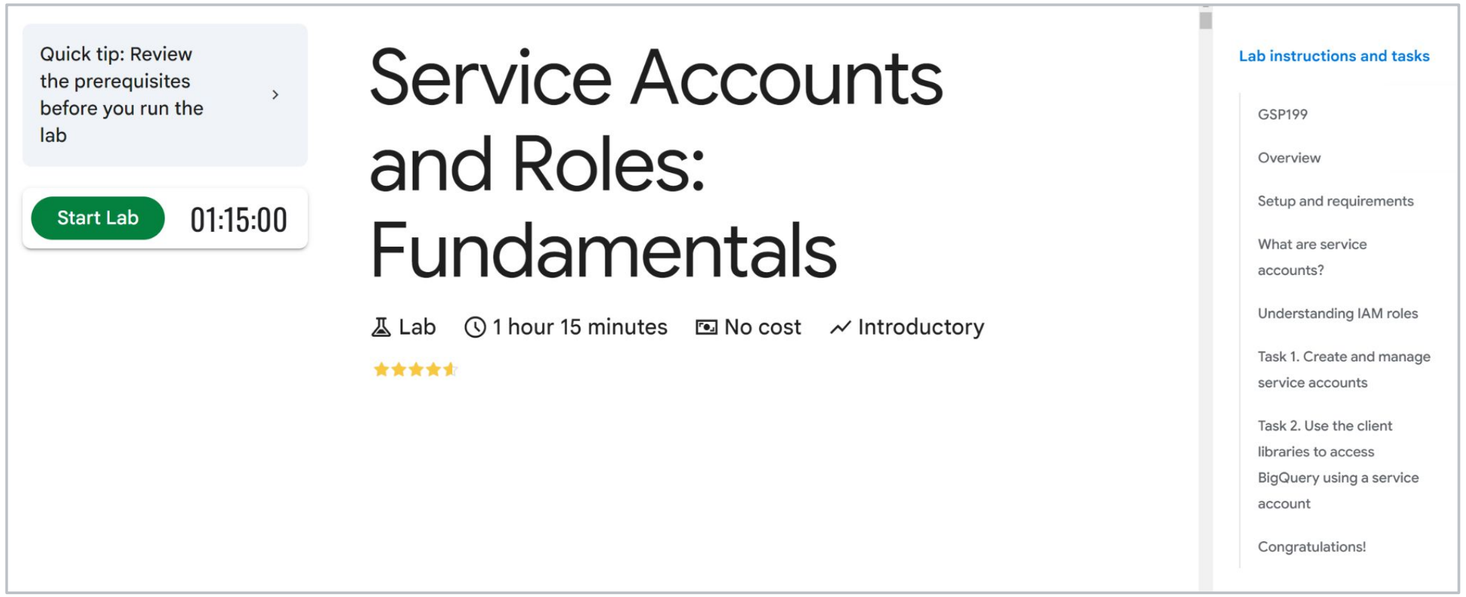
- Service Account Admin
  - Create and manage service accounts
- Service Account User
  - Can attach service account to resources (e.g., Compute Engine)
  - Can “impersonate” the service account and perform the tasks allowed by IAM given to the service account
- Service Account Key Admin
  - Create and manage (and rotate) service account keys
  - Keys are used by applications external to Google Cloud
- Service Account Token Creator
  - Short lived credentials represented as OAuth 2.0 access tokens, OpenID Connect ID tokens, self-signed JSON Web Tokens (JWTs), and self-signed binary objects (blobs)

For those taking notes using the downloaded PDFs:
I’m not discussing pages 145-154. Those are more appropriate for the ACE exam and will be removed in a future revision.

Suggested lab: Service Accounts and Roles: Fundamentals

In this lab you will

- Create a service account
  - IAM - BigQuery Data Viewer and BigQueryUser
- Attach it to a VM
- Create and run Python app to run a SQL query in BigQuery



https://partner.cloudskillsboost.google/catalog_lab/956

VPC Firewall rules (revisiting the topic)

Creating Firewall Rules

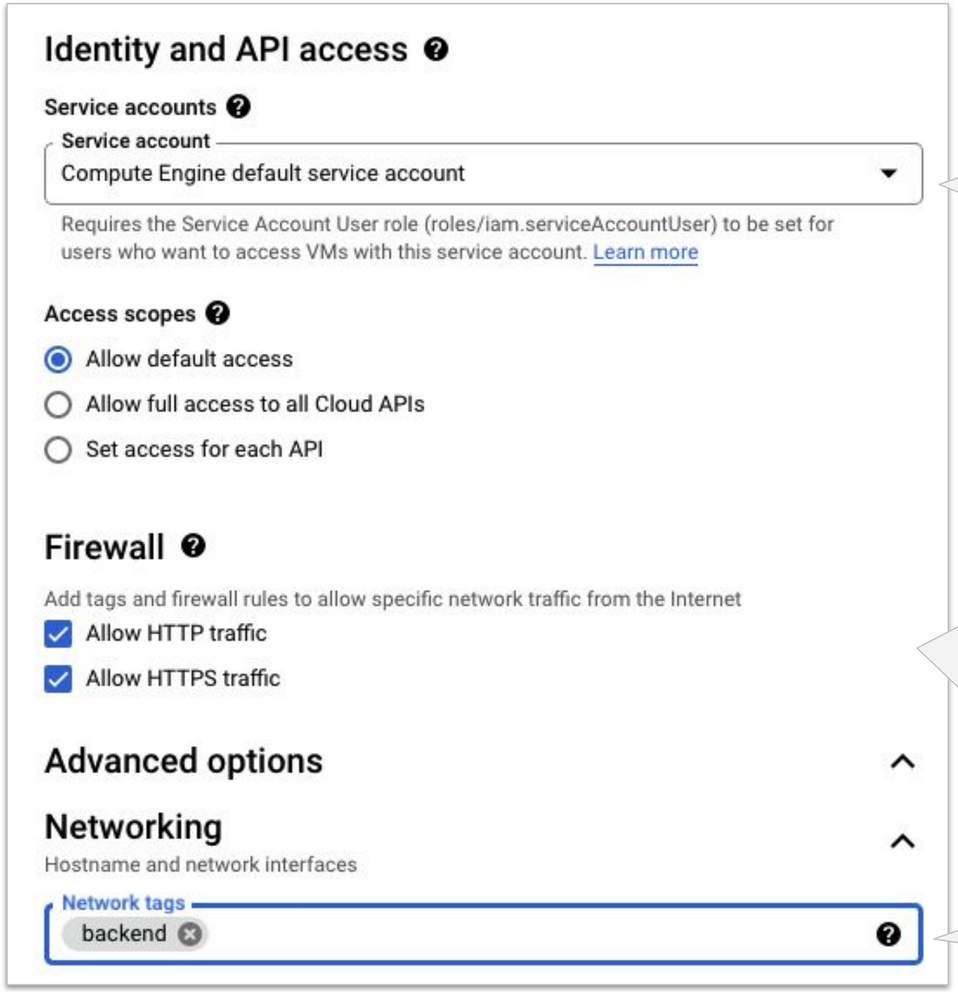
Saw this slide earlier

- When creating rules, specify
  - Source
    - Could be the internet (0.0.0.0/0 IP range)
    - Individual or ranges of IPv4 or IPv6 addresses
    - Could be VMs with specific network tags or service accou
  - Target - Defines which VMs the rule applies to
    - All instances in the network
    - VMs with specific network tags
    - VM's with service accounts

Will revisit the last 2 after service account discussion

Console

Creating a VM with network tags and a service account



Set the Service Account here

Developer needs a specific IAM role to attach a service account to a resource

Clicking one (or both) of these boxes results in automatically generated firewall rule(s) that apply to VMs with the network tags:

- allow-http
- allow-https

These tags are automatically added to the VM upon creation.

Can manually add network tags. One VM can have multiple tags

Creating firewall rules - tags vs service accounts

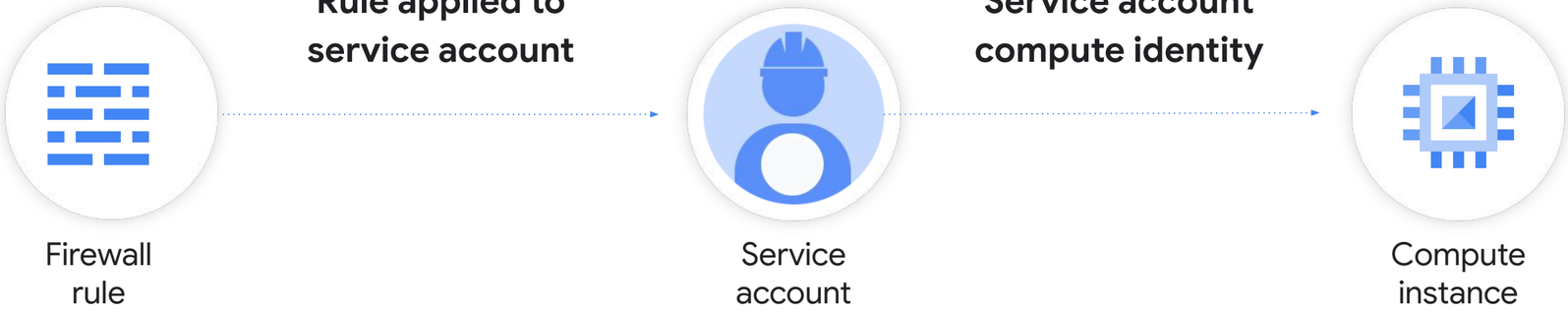
Network Tags

- Multiple tags applied to one VM (64 max)
- Firewall rule may target multiple tags
- May update tags to live VM

Best practice

Service accounts

- May restrict who uses
- Must shut down VM to change service account



Network Tags vs Service Accounts

Network Tags

Are very flexible

- Tags can be updated while a VM is running, while a service account cannot
- A VM can have multiple tags, while a VM can have one service account only
- FW rules match if >= 1 tag matchesTags

Potential con:

- Not possible to control who can apply a specific network tag.
- Example: A HTTP ingress allow rule exists that applies to instances with ‘webapp-http-allow’ network tag
- Anyone with Compute Engine Admin IAM role can create VM’s with this network tag.

```
gcloud compute firewall-rules create web-logdata \
    --network logging-network \
    --allow TCP:443 \
    --source-tags web-production \
    --target-tags log-data
```

Network Tags vs Service Accounts

Network Tags

Are very flexible

- Tags can be updated while a VM is running, while a service account cannot
- A VM can have multiple tags, while a VM can have one service account only
- FW rules match if >= 1 tag matchesTags

Potential con:

- Not possible to control who can apply a specific network tag.
- Example: A HTTP ingress allow rule exists that applies to instances with ‘webapp-http-allow’ network tag
- Anyone with Compute Engine Admin IAM role can create VM’s with this network tag.

Service accounts

Best practice from a security viewpoint

- Can control who can assign a given service account to instances
- Allows strict control over the instances a given rule will apply to.
- Goes hand in hand with IAM best practice - creating service accounts with minimal privileges for different applications and their components.

Network Tags vs Service Accounts

```
gcloud compute firewall-rules create frontend-to-backend \
--direction=INGRESS --network=default --allow TCP:443 \
--source-service-accounts=frontend@proj.iam.gserviceaccount.com \
--target-service-accounts=backend@proj.iam.gserviceaccount.com
```

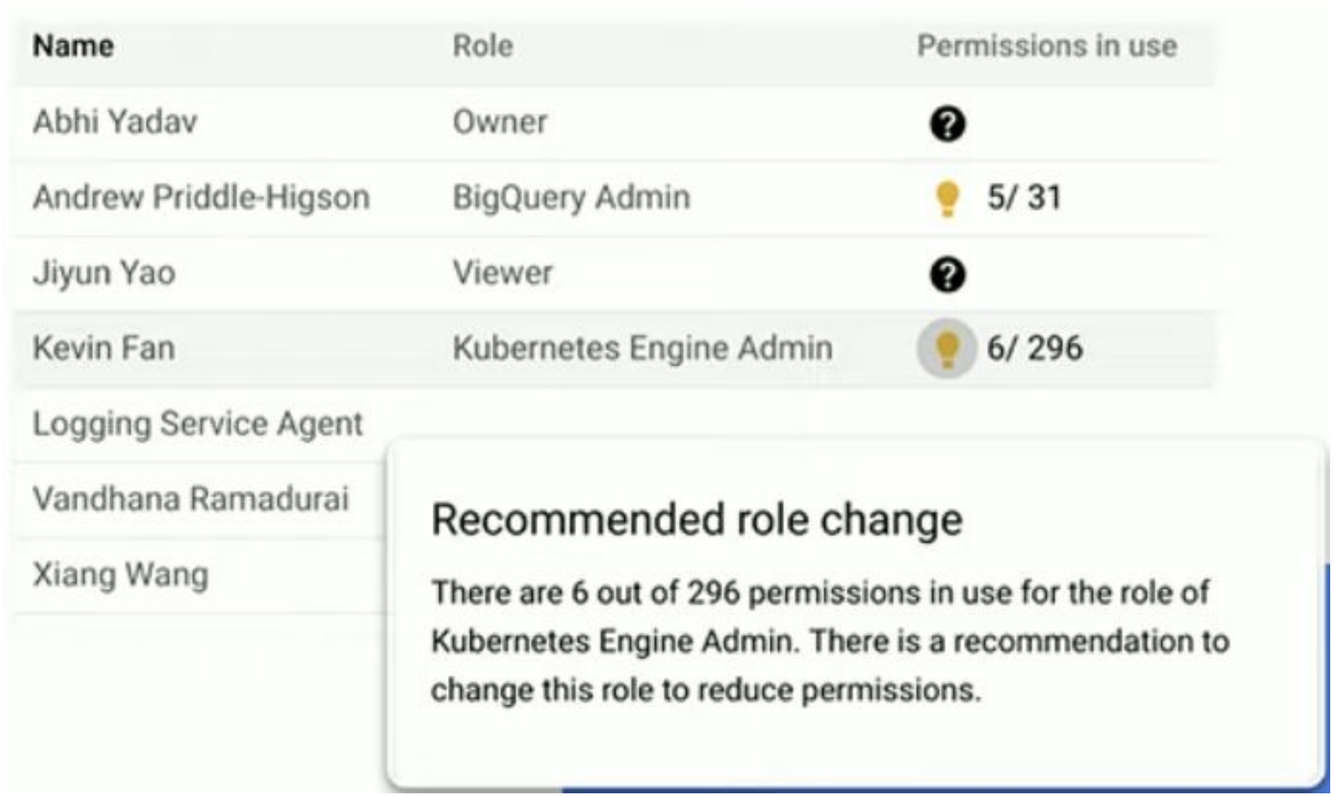
IAM Best Practices

- Separation of duties – use projects to control access to resources based on access requirements
  - Developer project allows the programmers access
  - Production project allows the operations people access
- Principle of least privilege – use roles to control what users can do
  - Avoid making everyone a Project Owner just because it's easy
- Use more restrictive rights at higher levels and provide access to lower levels and resources when required

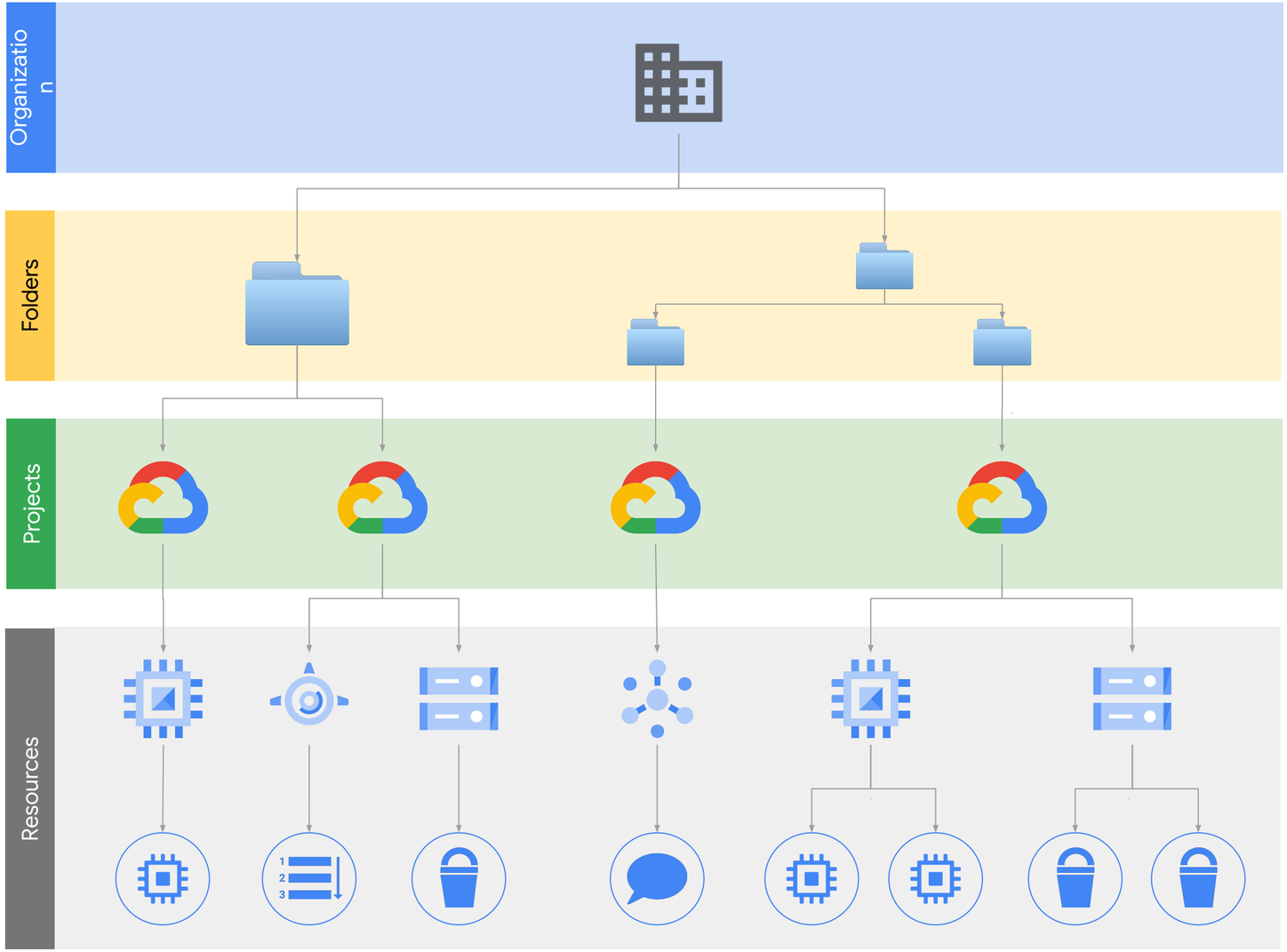
Cloud IAM Recommender

IAM Recommender

Automatically detect overly permissive access and suggests changes **based on permissions used in the last 90 days, similar users in the organization and their access patterns**



Resource Hierarchy

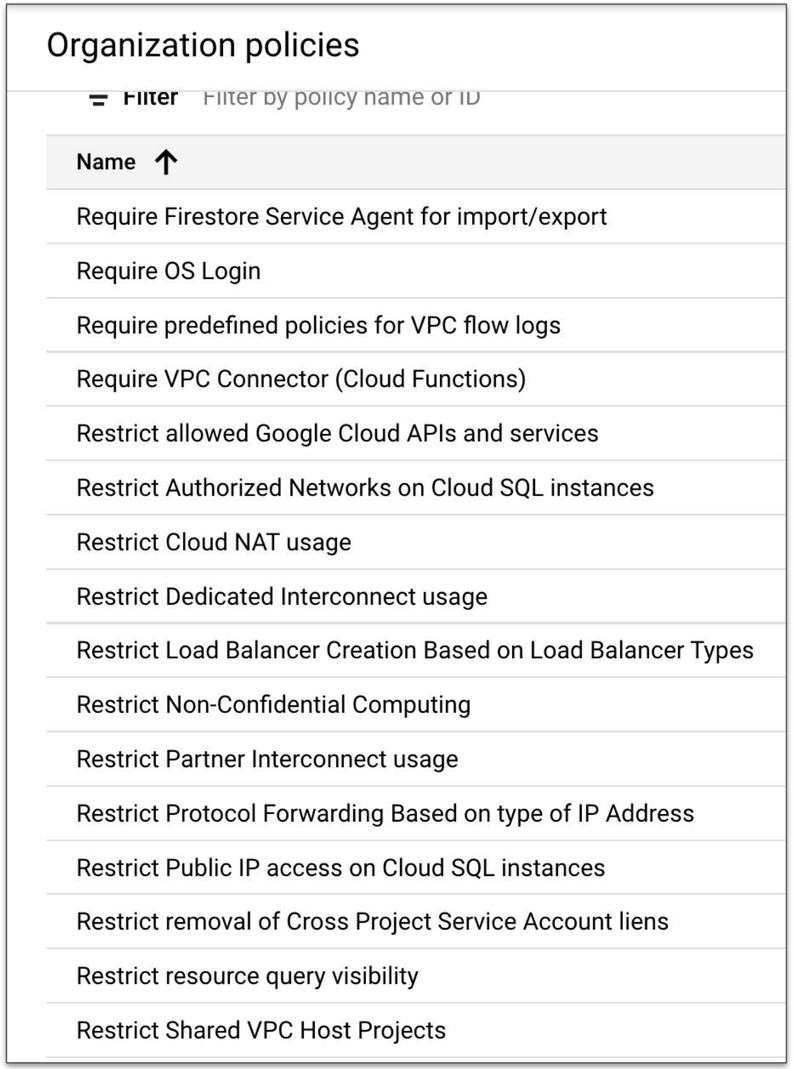


Next discussion: Organization Policies

Organizational policies

- An organization policy is a **restriction** or constraint that you can set over the use of a service
- For example
  - Restrict what type of VMs developers can create
  - Restrict what Google regions resources can be created in
  - Restrict the use of public IPs to some VMs (or none)
- Are set at the organization level, folder level or project level
- Define and establish guardrails for your development teams to stay within compliance boundaries.

[Introduction to the Organization Policy Service]

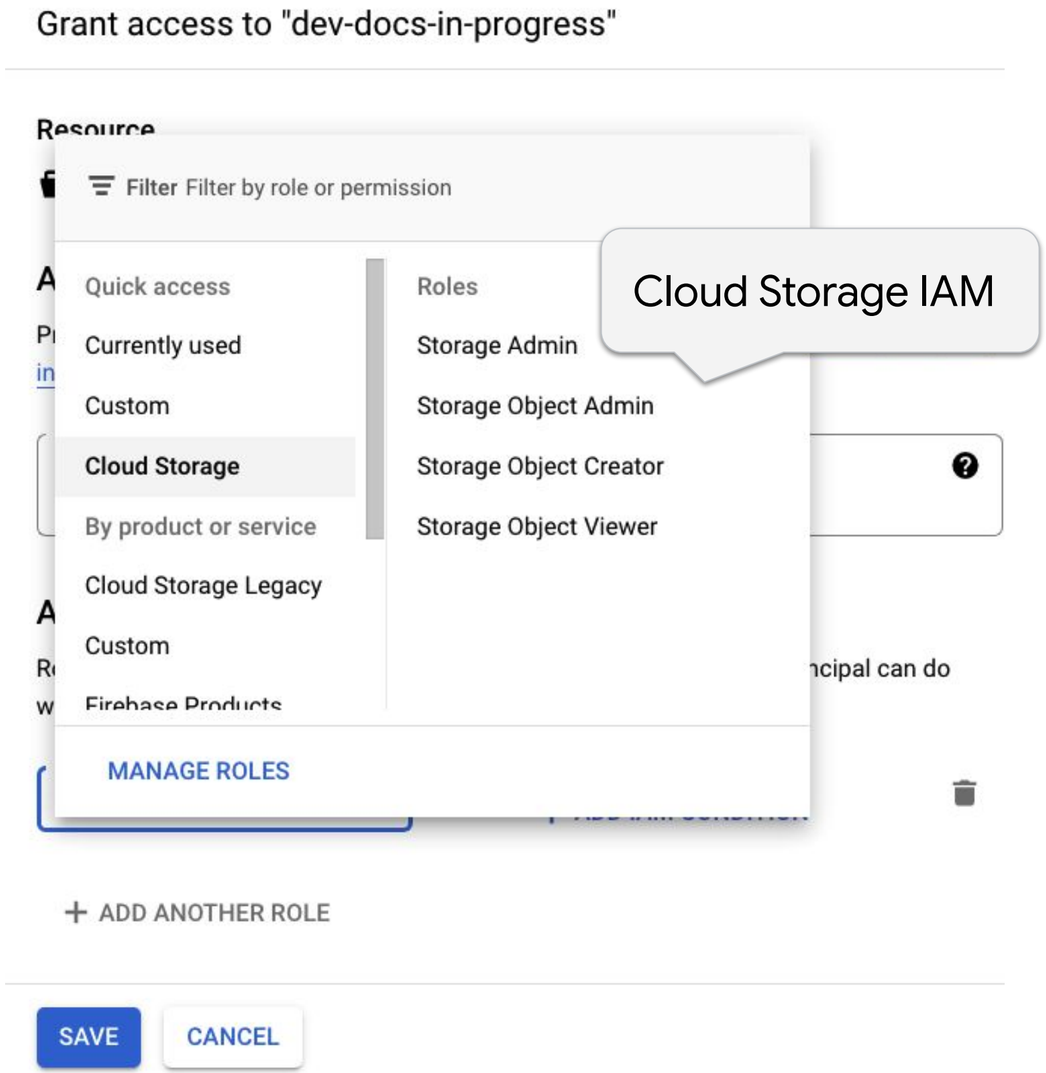


Cloud Storage security

Bucket Security

Permissions can be added to control access to buckets and objects

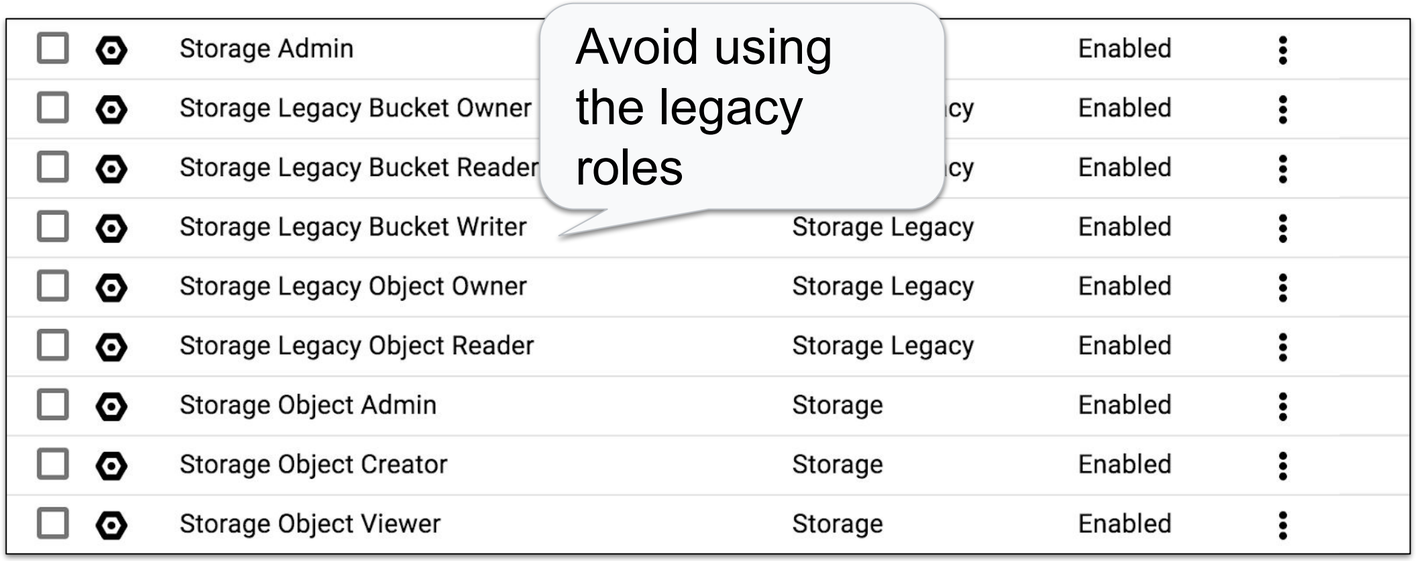
- Use IAM users and groups



Predefined IAM Storage Roles

Roles can be addedPredefined IAM Storage Roles to a principle or service account at the organization, folder, project or bucket level

- Bucket level - applies to that specific bucket only



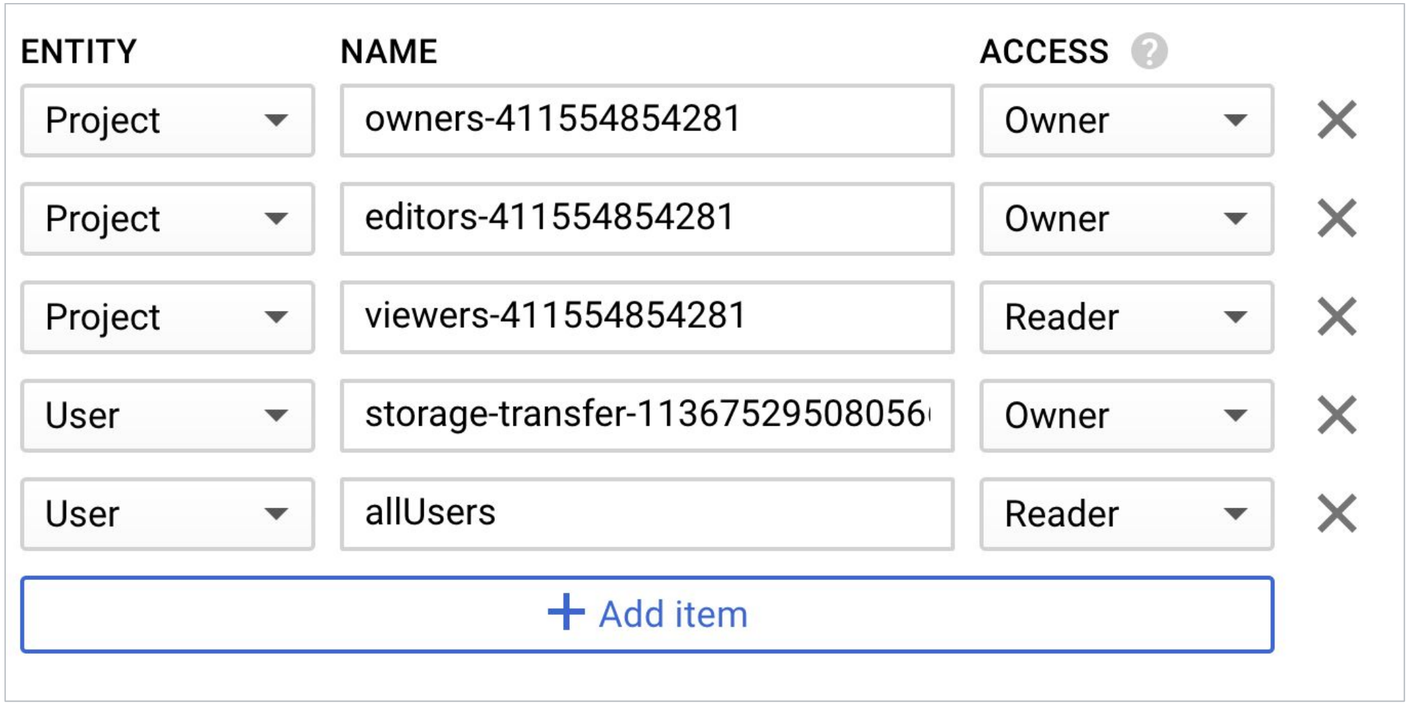
Storage Role Permissions

Storage Object Admin	Storage Object Creator	Storage Object Viewer
Description Full control of storage objects. 9 assigned permissions: ● resourcemanager.projects.get ● resourcemanager.projects.list ● storage.objects.create ● storage.objects.delete ● storage.objects.get ● storage.objects.getIamPolicy ● storage.objects.list ● storage.objects.setIamPolicy ● storage.objects.update	Description Access to create objects in storage. 3 assigned permissions: ● resourcemanager.projects.get ● resourcemanager.projects.list ● storage.objects.create	Description Read access to storage objects. 4 assigned permissions: ● resourcemanager.projects.get ● resourcemanager.projects.list ● storage.objects.get ● storage.objects.list

Storage Object ACLs

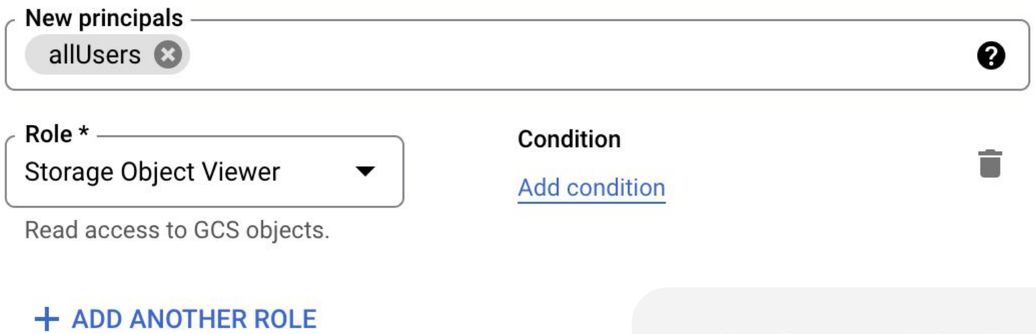
Access Control Lists (ACLs) can be used to grant access to objects in buckets

Not best practice



Making buckets public

To make a bucket public, grant allUsers the Storage Object Viewer role.



To make an object public, grant allUsers Reader access

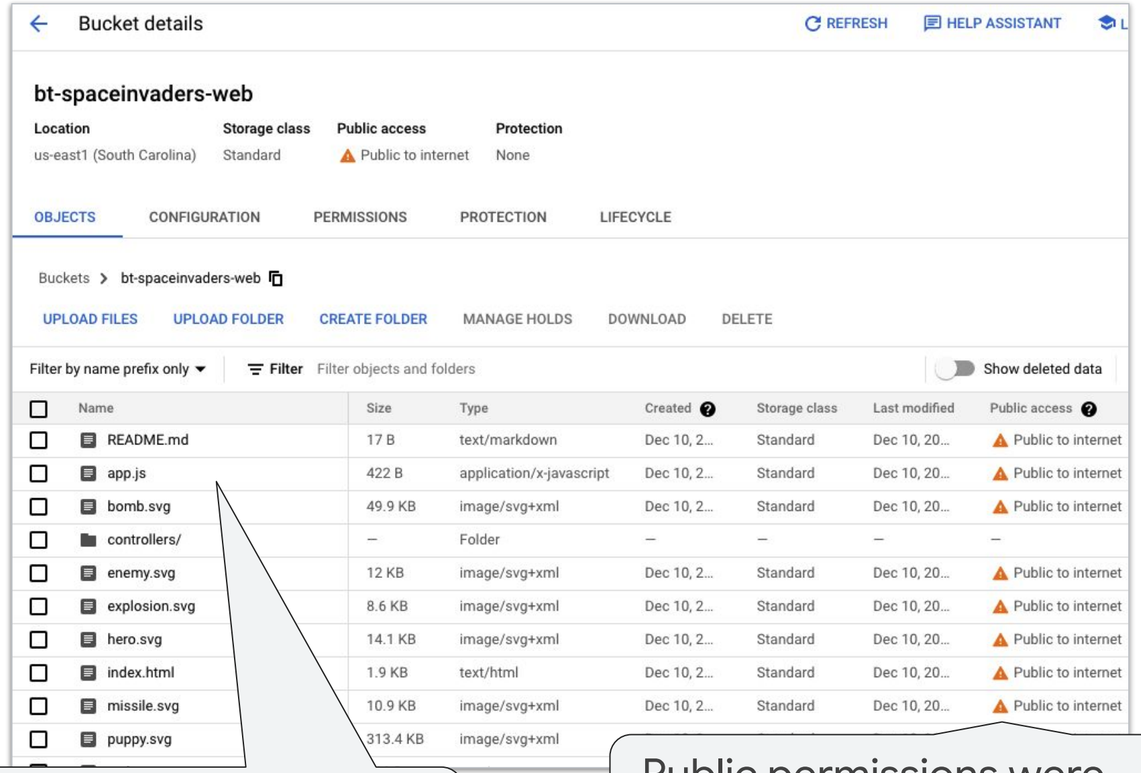
Only for publicly accessible web content:

Use with caution!

Note: There is also an allAuthenticatedUsers role.
This represents the principals who have identities within your Google Cloud domain

Delivering Static Web Content Worldwide

- Use buckets to host web applications written with JavaScript frameworks
  - Angular, React, etc.
- Upload all static files to a bucket (.css, .js, .html, .jpb, .png, etc.)
- Make the files public to the internet
  - Give `allUsers` read access
- Objects that are public include a public link



Signed URLs

Provide temporary access to buckets

- Create a service account with rights to storage
- Create a service account key
- Use signurl command to create a URL that allows access to the resource
  - -d parameter is used to specify duration

```
gcloud iam service-accounts keys create ~/key.json --iam-account
storage-admin-sa@doug-demo-project.iam.gserviceaccount.com


gsutil signurl -d 10m ~/key.json gs://super-secure-bucket/noir.jpg
```

Signed URL Example Output

```
me@doug-demo-project:~$ gsutil signurl -d 10m ~/key.json gs://super-secure-bucket/noir.jpg
URL     HTTP Method     Expiration      Signed URL
gs://super-secure-bucket/noir.jpg       GET     2018-08-31 16:29:25     https://storage.google
apis.com/super-secure-bucket/noir.jpg?x-goog-signature=107d26e38f5c962296c26f4153a1cbeb61a84ac
a905009752e849f8f890de1f9a80e482da3bae562c7796389e12a8657a70c87860700149c4b2218c81ad3d57730cd3
5ced850b266cdffd84de01898ee8c807d742a85136e56f46d83c29ceb792bdd3a22adbe2e540ba27b0f565bbf8f31a
ee6ae61d6ae20968021d5a47c8d0aada43f2d32407f2977a4c7b4c66ef64ddd68bd6f6135936f847ace3530a968d72
63ff5e70f9fc39bf16fabbd472f63584a8d8c6b24b1f81859f1c5176b8e97580a6b4a7613ad76bfcdd403e6afc9a70
90a3e1b4cf95c7fb68142416af86ef5ef6bfab93c00492b307233180df9b3dfeefe1bb9a5bf81cb441f879ecc2e57c
def&x-goog-algorithm=GOOG4-RSA-SHA256&x-goog-credential=storage-admin-sa%40doug-demo-project.i
am.gserviceaccount.com%2F20180831%2Fus%2Fstorage%2Fgoog4_request&x-goog-date=20180831T201925Z&
x-goog-expires=600&x-goog-signedheaders=host
me@doug-demo-project:~$
```

Creating a Signed URL with Python example

```python
import datetime

from google.cloud import storage


def generate_download_signed_url_v4(bucket_name, blob_name):
    """Generates a v4 signed URL for downloading a blob.
    Note that this method requires a service account key file.
    """
    # bucket_name = 'your-bucket-name'
    # blob_name = 'your-object-name'

    storage_client = storage.Client()
    bucket = storage_client.bucket(bucket_name)
    blob = bucket.blob(blob_name)

    url = blob.generate_signed_url(
        version="v4",
        # This URL is valid for 15 minutes
        expiration=datetime.timedelta(minutes=15),
        # Allow GET requests using this URL.
        method="GET",
    )

    print("Generated GET signed URL:")
    print(url)
    print("You can use this URL with any user agent, for example:")
    print(f"curl '{url}'")
    return url

generate_download_signed_url_v4("bt-acme-test-data", "fish2.png")
```

Call the method passing in a bucket name and file name

Example output

https://storage.googleapis.com/bt-acme-test-data/fish2.png?X-Goog-Algorithm=GOOG4-RSA-SHA256&X-Goog-Credential=bucket-admin%40bt-dev-general.iam.gserviceaccount.com%2F20230906%2Fauto%2Fstorage%2Fgoog4_request&X-Goog-Date=20230906T154204Z&X-Goog-Expires=900&X-Goog-SignedHeaders=host&X-Goog-Signature=5aca5d9163c0c2d9152d524c8ae048549cdd3a6ca5a3464d883b6b5b1c71ac86fdd17d628ce1bb5dc90dd96f823ec4672c47c97a1f954f49ff7c2059731217538e44c48f87bafbe1e48b49cbdcf195ce5bde09d8e906387dabc6fafc12b973d993fbbad2601febd45ee309946c69f8b129db8eebdde9d84b6ffc84c3be9438f1c683add09161528bcc38119d7b74e4bf0f68c9fe7ed8d1884b77a1d8e4ef141743f7c794be0ddf6053232633984a49290b6a080d974f9078fddf7db03ae76e3e1fe1f41d8361bda7e734665a86b4d35d4ee9114ad11b20cc65ba8a6f68f4cc974bd715a329b7016ed5662f564942c7d540b149fdb4c6c7a667d874a3f0a1e367