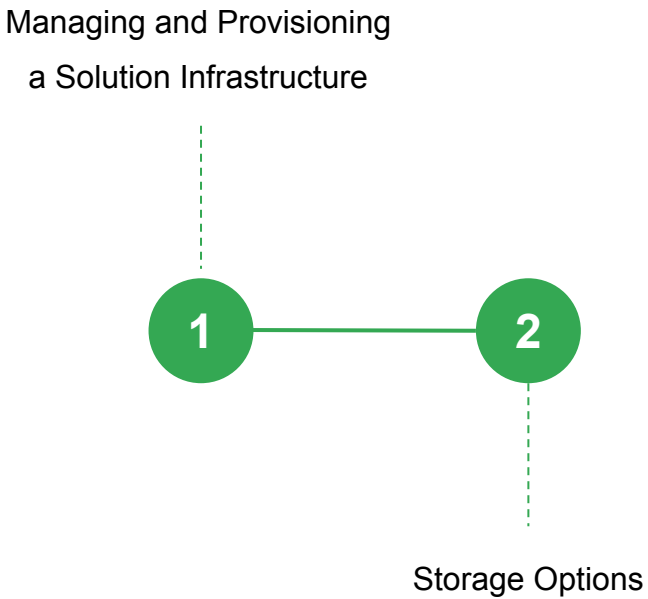
Google Cloud

Professional Cloud Architect

Preparing for Professional Cloud Architect Journey for AWS Professionals

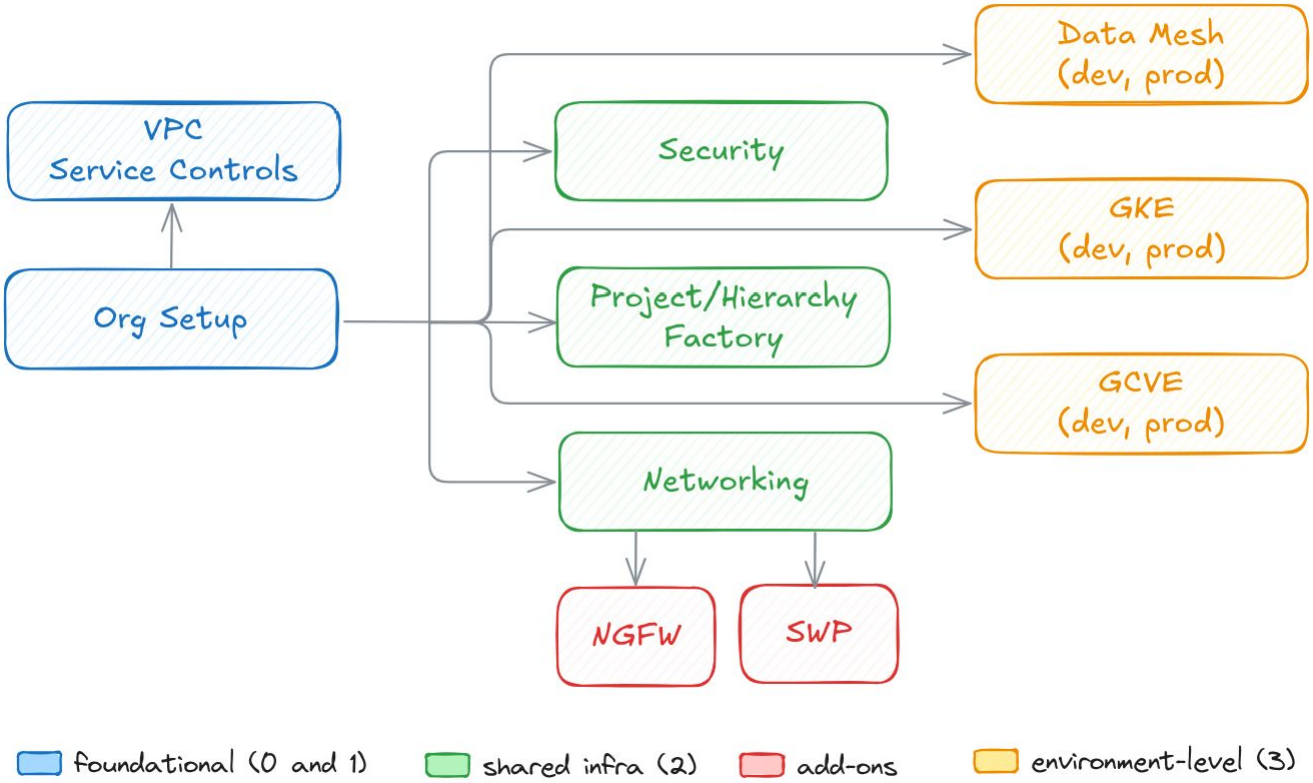
Session 4 topics



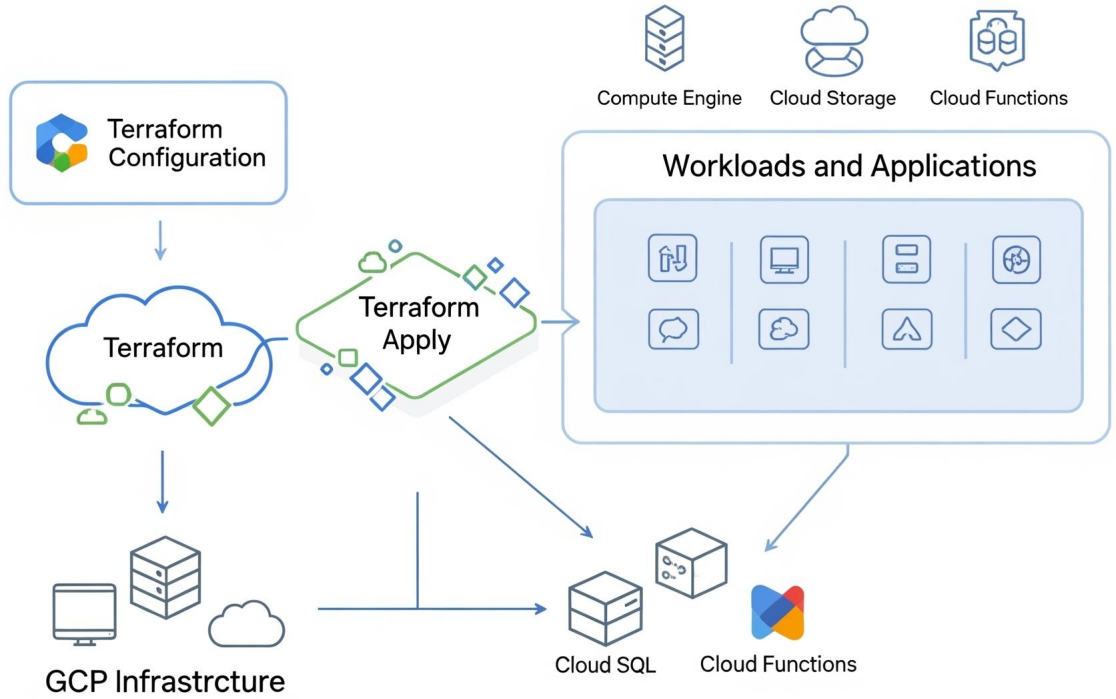
Managing and Provisioning a Solution Infrastructure

Before workload migration starts...

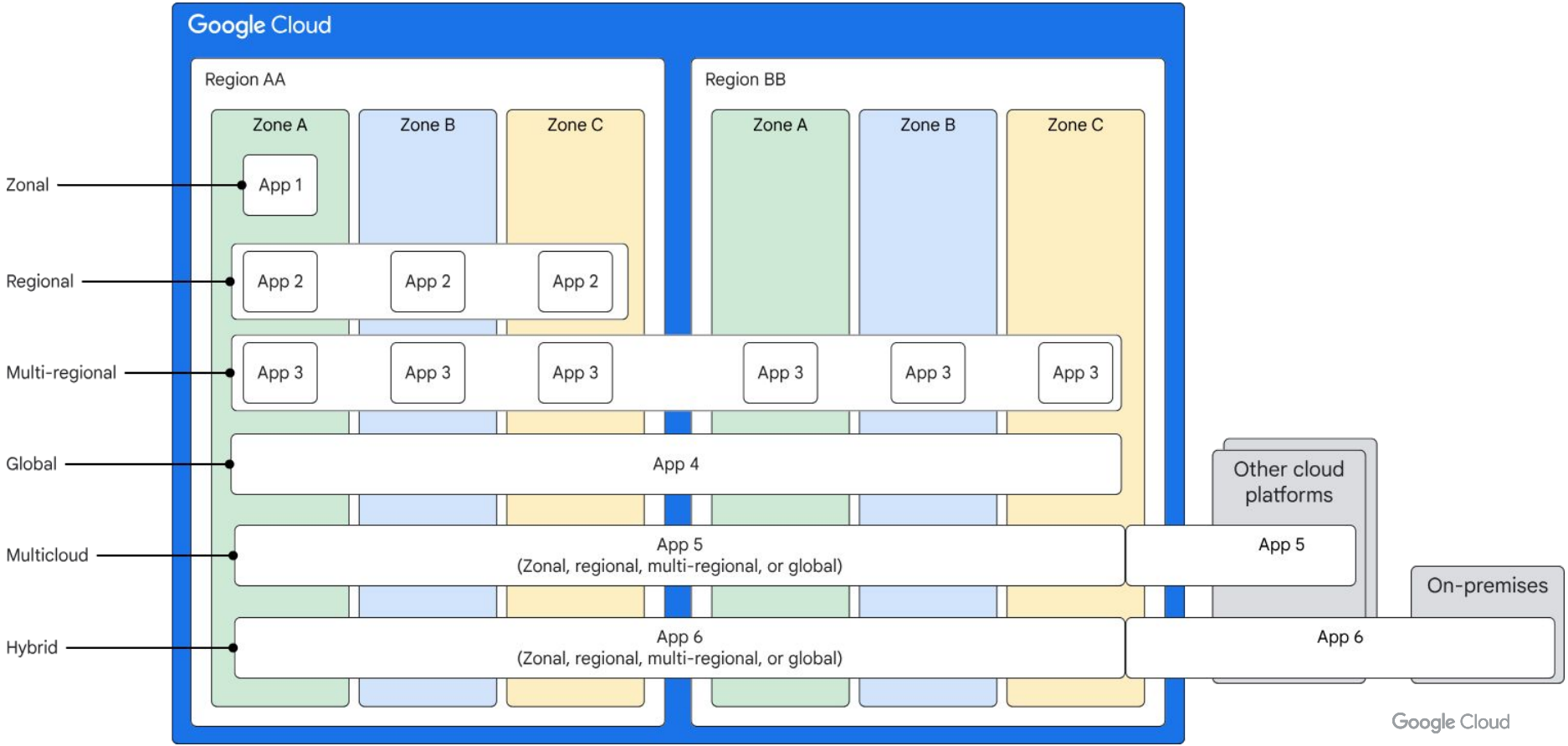
... deploy a Landing Zone



... and only then, focus on your workloads and apps.

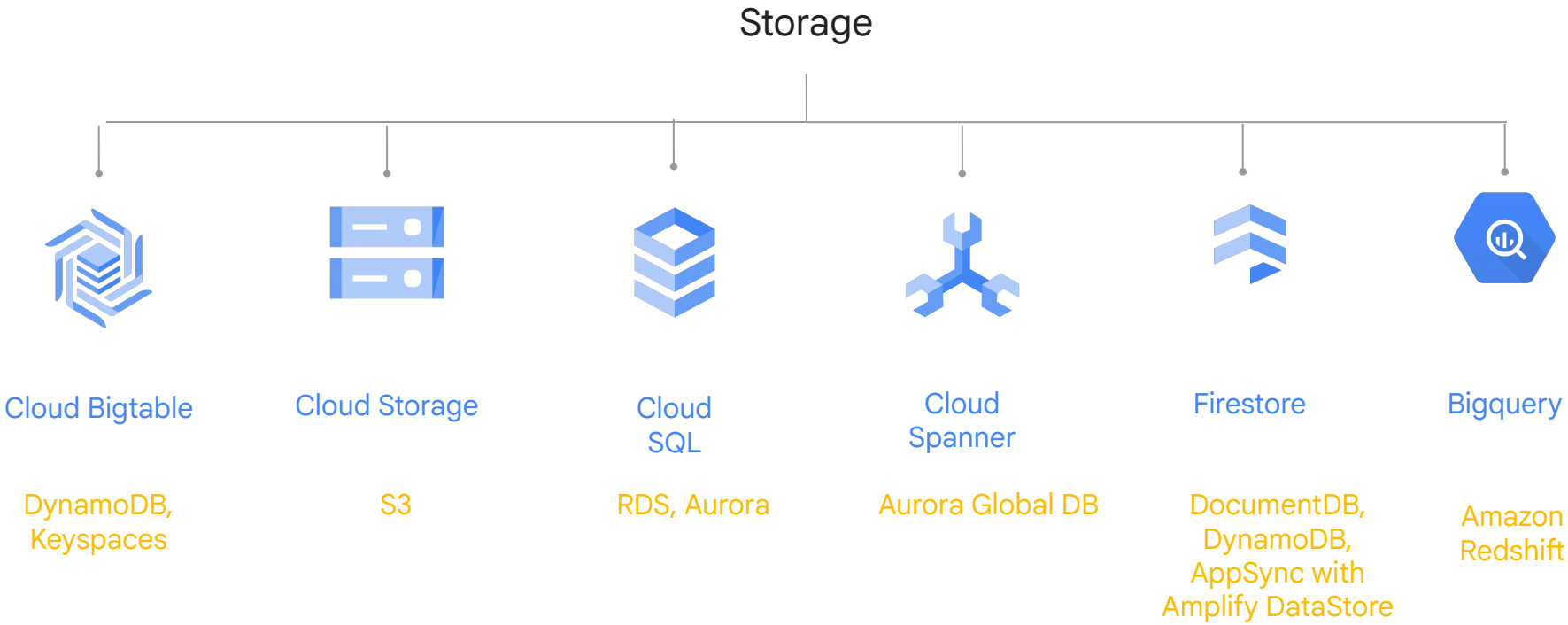


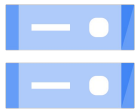
Deployment archetypes



Storage Options

Google Cloud offers a range of storage services





Cloud Storage

S3

Google Cloud Storage and Amazon S3: Similarities

- Use buckets as their units of deployment
- Use objects
- Allow for built-in versioning of all objects in a bucket, and you can configure lifecycle management policies and rules on individual buckets
- Let you enable automatic object versioning
- Use globally unique names as bucket identifiers
- Allow creation of storage buckets using their respective command-line interfaces (CLIs), software development kits (SDKs), and consoles
- Allow triggering of serverless functions through bucket events

Google Cloud Storage and Amazon S3: Differences (⅓)

Topic	Amazon S3	Cloud Storage
DNS compliant naming	Buckets must have DNS compliant names.	Bucket names must only be DNS compliant if you intend to use the bucket name in DNS records.
Deployment locality	Uses regional deployments Additional buckets can be deployed in other regions with object replication.	Deployment localities are multi-regional, dual-regional, and regional.
Object tagging	Lets users tag individual objects with key-value pairs that can be used in analytics, access control, and lifecycle management	Lets you apply tags to buckets Does not support object-level tags

Google Cloud Storage and Amazon S3: Differences (⅔)

Topic	Amazon S3	Cloud Storage
Multi-factor authentication delete	Lets users configure buckets with an additional layer of security, requiring multi-factor authentication on all CLI or API requests to delete an object	Does not offer a similar feature
Event notification	Buckets can be configured to use Amazon Simple Queue Service (SQS) and Amazon Simple Notification Service (SNS) for event notification.	Buckets can be configured to use Google Cloud Pub/Sub for event notification.

Google Cloud Storage and Amazon S3: Differences (3/3)

Topic	Amazon S3	Cloud Storage
Resource hierarchy	1. Account 2. Bucket 3. Object	1. Project 2. Bucket 3. Object
Storage tiers	Standard Intelligent Tiering Standard-Infrequent Access One Zone-Infrequent Access Glacier Instant Retrieval Glacier Flexible Retrieval Glacier Deep Archive	Standard Nearline Coldline Archive

Know these GCS features well!

- Controlling object lifecycle:
  - Retention policy (best for compliance)
  - Object Hold (prevent individual objects from being deleted)
  - Object Versioning (aka “automatic backups with retention policy; be aware of additional costs)
  - Object Lifecycle Management (aka “object TTL” / downgrade class to optimize costs)
- ACLs (read, write, full control on buckets or an object).
- Objects are immutable.
- Location constraints on buckets.
- Pub/Sub notifications for Google Cloud Storage.
- Resumable uploads (can restart from the last successful chunk).
- Strong consistency except for cached objects.
- Storage class set at object level (fine-grained performance/cost control without moving data to different buckets).
- Cloud Storage Triggers to handle events in Cloud Functions.
- Streaming uploads to GCS.
- For data encryption, GCS supports GMEK, CMEK and CSEK (most services do not support CSEK!)

GCS storage classes

	Standard	Nearline (Infrequent Access)	Coldline (Glacier Instant Retrieval)	Archive (Glacier Deep 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 (180) 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

Storage classes in Amazon S3 and Cloud Storage

Topic	Amazon S3	Cloud Storage
Tiers	Three options for longest-term storage • Amazon S3 Glacier Instant Retrieval is the closest to Cloud Storage's Archive storage • Other options have longer wait times	One option for the longest-term storage
Automatic switching between storage classes	Offers an intelligent storage option that automatically switches data between storage classes	Autoclass feature automatically transitions objects in your bucket to appropriate storage classes based on each object's access pattern

Additional resource: Comparison of storage types in Amazon S3 and Google Cloud and their characteristics

Best Practices on Storage Class Selection

Consider retention period and access frequency

Retention Period

Access Frequency	<1 mo	1–3 mo	3–12 mo	>12 mo
>12/yr	Standard	Standard	Standard	Standard
4–12/yr	Standard	Nearline	Nearline	Nearline
1–4/yr	Standard	Nearline	Coldline	Coldline
<1/yr	Standard	Nearline	Coldline	Archive

Exam Tips:

- *Each storage class has so-called "minimum storage duration", so when optimizing costs, you also need to validate if you'll need to keep your objects for at least this amount of time.*

Characteristics applicable to all storage classes

- Unlimited storage with no minimum object size.
- Worldwide accessibility and worldwide storage locations.
- Low latency (time to first byte typically tens of milliseconds for all storage classes, unlike AWS Glacier Flexible Retrieval [minutes+] and Deep Archive [hours]).
- High durability (99.999999999% annual durability).
- Geo-redundancy if the data is stored in a multi-region or dual-region bucket.
- A uniform experience with Cloud Storage features, security, tools, and APIs.

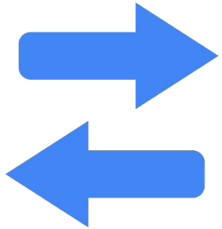
Transferring data into Google Cloud can be challenging

		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

There are several ways to bring data into Cloud Storage

Online transfer

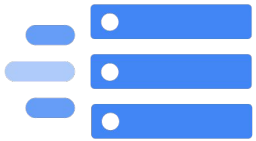
Self-managed copies using command-line tools or drag-and-drop.



Storage Transfer Service

Scheduled, managed batch transfers.

DataSync, Transfer Family



Transfer Appliance

Rackable appliances to securely ship your data.

Snow family (e.g. Snowball)

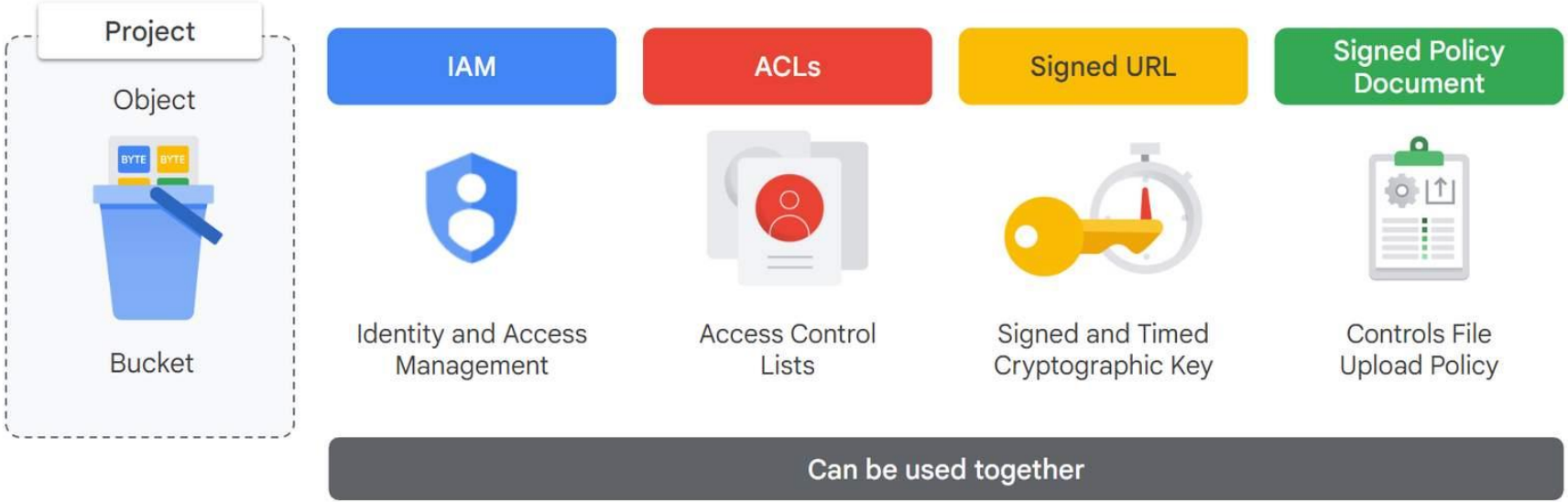
Choose the right tool to move data to GCS...

Where you're moving data from	Scenario	Suggested products
Another cloud provider (for example, Amazon Web Services or Microsoft Azure) to Google Cloud	—	Storage Transfer Service
Cloud Storage to Cloud Storage (two different buckets)	—	Storage Transfer Service
Your private data center to Google Cloud	Enough bandwidth to meet your project deadline for less than 1 TB of data	gcloud storage
Your private data center to Google Cloud	Enough bandwidth to meet your project deadline for more than 1 TB of data	Storage Transfer Service for on-premises data
Your private data center to Google Cloud	Not enough bandwidth to meet your project deadline	Transfer Appliance

Exam Tips:

- *Depending on size and throughput, use gcloud storage / Transfer Service (low cost) / Transfer Appliance*
- ***When using Transfer Appliance, you need to execute so-called "rehydration" process** which will decrypt and uncompress before it's put to a destination bucket.*

Controlling access in Cloud Storage



Diagnostic Question Discussion

Your company has an application that is running on multiple instances of Compute Engine. It generates 1 TB per day of logs. For compliance reasons, the logs need to be kept for at least two years. The logs need to be available for active query for 30 days. After that, they just need to be retained for audit purposes. You want to implement a storage solution that is compliant, minimizes costs, and follows Google-recommended practices.

What should you do?

A. 1. Write a daily cron job, running on all instances, that uploads logs into a Cloud Storage bucket. 2. Create a sink to export logs into a regional Cloud Storage bucket. 3. Create an Object Lifecycle rule to move files into a Coldline Cloud Storage bucket after one month

B. 1. Install a Cloud Logging agent on all instances. 2. Create a sink to export logs into a partitioned BigQuery table. 3. Set a time_partitioning_expiration of 30 days.

C. 1. Install a Cloud Logging agent on all instances. 2. Create a sink to export logs into a regional Cloud Storage bucket. 3. Create an Object Lifecycle rule to move files into a Coldline Cloud Storage bucket after one month. 4. Configure a retention policy at the bucket level using bucket lock.

D. 1. Create a daily cron job, running on all instances, that uploads logs into a partitioned BigQuery table. 2. Set a time_partitioning_expiration of 30 days.

Diagnostic Question Discussion

Your company has an application that is running on multiple instances of Compute Engine. It generates 1 TB per day of logs. For compliance reasons, the logs need to be kept for at least two years. The logs need to be available for active query for 30 days. After that, they just need to be retained for audit purposes. You want to implement a storage solution that is compliant, minimizes costs, and follows Google-recommended practices.

What should you do?

A. 1. Write a daily cron job, running on all instances, that uploads logs into a Cloud Storage bucket. 2. Create a sink to export logs into a regional Cloud Storage bucket. 3. Create an Object Lifecycle rule to move files into a Coldline Cloud Storage bucket after one month

B. 1. Install a Cloud Logging agent on all instances. 2. Create a sink to export logs into a partitioned BigQuery table. 3. Set a time_partitioning_expiration of 30 days.

C. 1. Install a Cloud Logging agent on all instances. 2. Create a sink to export logs into a regional Cloud Storage bucket. 3. Create an Object Lifecycle rule to move files into a Coldline Cloud Storage bucket after one month. 4. Configure a retention policy at the bucket level using bucket lock.

D. 1. Create a daily cron job, running on all instances, that uploads logs into a partitioned BigQuery table. 2. Set a time_partitioning_expiration of 30 days.

Diagnostic Question Discussion

Your operations team currently stores 10 TB of data in an object storage service from a third-party provider. They want to move this data to a Cloud Storage bucket as quickly as possible, following Google-recommended practices. They want to minimize the cost of this data migration.

Which approach should they use?

A. Use the “gcloud storage mv” command to move the data.

B. Use the Storage Transfer Service to move the data.

C. Download the data to a Transfer Appliance, and ship it to Google.

D. Download the data to the on-premises data center, and upload it to the Cloud Storage bucket.

Diagnostic Question Discussion

Your operations team currently stores 10 TB of data in an object storage service from a third-party provider. They want to move this data to a Cloud Storage bucket as quickly as possible, following Google-recommended practices. They want to minimize the cost of this data migration.

Which approach should they use?

A. Use the “gcloud storage mv” command to move the data.

B. Use the Storage Transfer Service to move the data.

C. Download the data to a Transfer Appliance, and ship it to Google.

D. Download the data to the on-premises data center, and upload it to the Cloud Storage bucket.

Diagnostic Question Discussion

Your company has created an application that uploads a report to a Cloud Storage bucket. When the report is uploaded to the bucket, you want to publish a message to a Cloud Pub/Sub topic. You want to implement a solution that will take a small amount to effort to implement.

What should you do?

A. Create an App Engine application to receive the file; when it is received, publish a message to the Cloud Pub/Sub topic.

B. Configure the Cloud Storage bucket to trigger Cloud Pub/Sub notifications when objects are modified.

C. Create a Cloud Function that is triggered by the Cloud Storage bucket. In the Cloud Function, publish a message to the Cloud Pub/Sub topic.

D. Create an application deployed in a Google Kubernetes Engine cluster to receive the file; when it is received, publish a message to the Cloud Pub/Sub topic.

Diagnostic Question Discussion

Your company has created an application that uploads a report to a Cloud Storage bucket. When the report is uploaded to the bucket, you want to publish a message to a Cloud Pub/Sub topic. You want to implement a solution that will take a small amount to effort to implement.

What should you do?

A. Create an App Engine application to receive the file; when it is received, publish a message to the Cloud Pub/Sub topic.

B. Configure the Cloud Storage bucket to trigger Cloud Pub/Sub notifications when objects are modified.

C. Create a Cloud Function that is triggered by the Cloud Storage bucket. In the Cloud Function, publish a message to the Cloud Pub/Sub topic.

D. Create an application deployed in a Google Kubernetes Engine cluster to receive the file; when it is received, publish a message to the Cloud Pub/Sub topic.

https://cloud.google.com/storage/docs/pubsub-notifications#other_notification_options

Storage options:

noSQL databases

Cloud Bigtable

DynamoDB, Keyspaces

Cloud Bigtable (DynamoDB / Keyspaces) is managed NoSQL

- Fully managed NoSQL, wide-column database service for terabyte applications.
  - Wide column database: Keyspaces
  - General NoSQL, scalable database: DynamoDB
- Integrated
  - Accessed using HBase API
  - Native compatibility with big data, Hadoop ecosystems

Cloud Bigtable vs Amazon DynamoDB - key differences

Feature	Cloud Bigtable	Amazon DynamoDB
Read options	Single row Row key prefix Range	Single row Row key (short)
Provisioning	Managed instances	Serverless
Autoscaling	User-assigned triggers	Automatic
Maximum item size	100 MB	400 KB
Secondary indices	Not supported	Supported
Backup	Integrated in the cluster	Point-in-time restore

Why choose Cloud Bigtable?

- Replicated storage.
- Data encryption in-flight and at rest.
- Scales Linearly
- Consistent sub-10ms latency
- Role-based ACLs.
- Drives major applications such as Google Analytics and Gmail (DynamoDB is used extensively on amazon.com).

What is Bigtable good for?

Use Case Examples

- **Time-series** data, such as CPU and memory usage over time for multiple servers.
- **Marketing data**, such as purchase histories and customer preferences.
- **Financial data**, such as transaction histories, stock prices, and currency exchange rates.
- **Internet of Things** data, such as usage reports from energy meters and home appliances.
- **Graph data**, such as information about how users are connected to one another.

Applications that need...

- Very high throughput
- Scalability
- Non-Structured key/value data where each value is no larger than 10MB

Storage Engine

- Batch MapReduce
- Stream Processing/Analytics
- ML applications

Exam Tip: *types of apps where you'd consider using Bigtable: recommendation engines, personalizing user experience, Internet of Things, real-time analytics, fraud detection, migrating from HBase or Cassandra, Fintech, gaming, high-throughput data streaming for creating / improving ML models.*

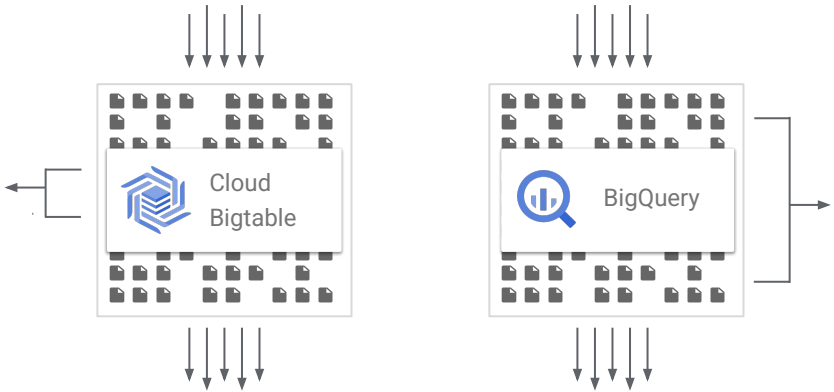
Bigtable for analytics... ?

Bigtable vs BigQuery

NoSQL wide column database

Low latency per-entry access

Heavy read/write events



Enterprise data warehouse for relational structured data

Large scale, ad-hoc SQL-based OLAP analysis

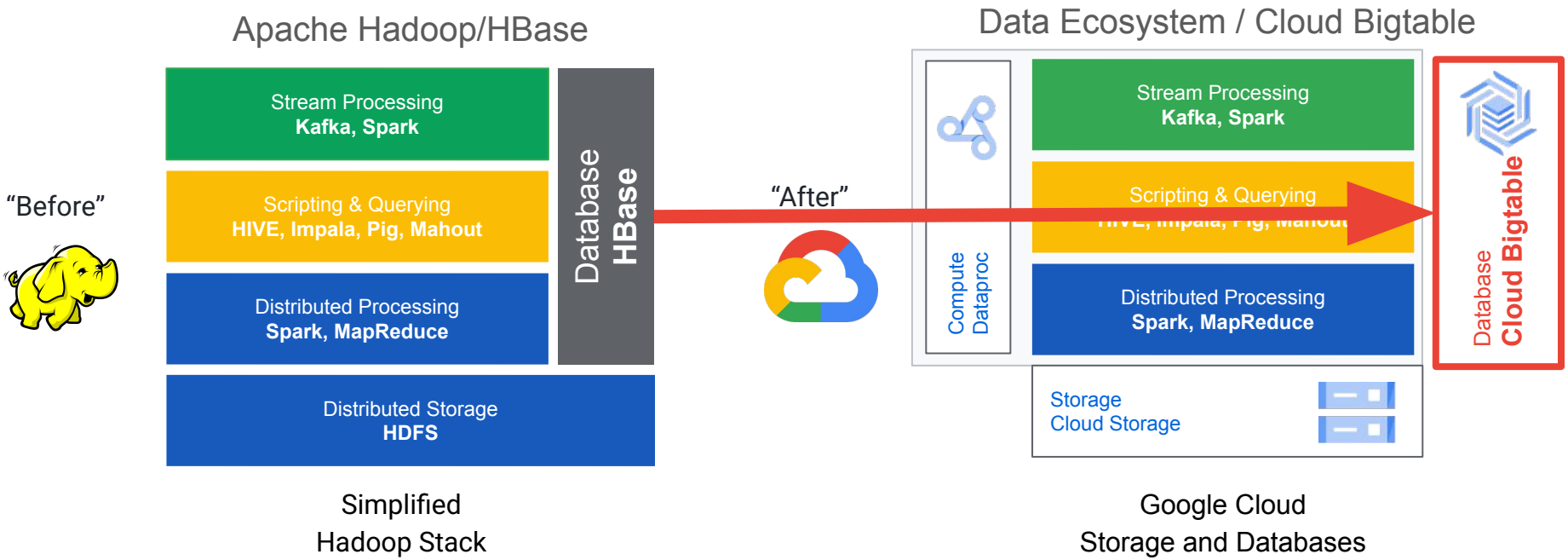
Organizational insights

Analyze data from Cloud Bigtable database

Optimized for	**Point read/write**	**Ad-hoc analysis and reporting**
Typical target	**User/entity level**	**Cohort /population level**

Exam Tip: *BigTable might be optimal for "real-time analytics", when you need to make decisions on events as they're happening.*

Bigtable: Hadoop migration and modernization



Exam Tip: *Main goal: decoupling of storage & compute. As a consequence, you can treat Dataproc clusters as job-specific / ephemeral*

What is Bigtable not good for?

Not good for…

- Not a relational database
- No SQL Queries or Joins
- No Multi-Row Transactions

Considerations

- You need full SQL support for OLTP
  → consider Spanner or CloudSQL
- Interactive querying for OLAP
  → consider BigQuery
- Need to store immutable blobs larger than 10MB (e.g. movies, images)
  → consider Cloud Storage

Diagnostic Question Discussion

You want to optimize the performance of an accurate, real-time, weather-charting application. The data comes from 50,000 sensors sending 10 readings a second, in the format of a timestamp and sensor reading.

Where should you store the data?

A. Google BigQuery

B. Google Cloud SQL

C. Google Cloud Bigtable

D. Google Cloud Storage

Diagnostic Question Discussion

You want to optimize the performance of an accurate, real-time, weather-charting application. The data comes from 50,000 sensors sending 10 readings a second, in the format of a timestamp and sensor reading.

Where should you store the data?

A. Google BigQuery

B. Google Cloud SQL

C. Google Cloud Bigtable

D. Google Cloud Storage

Diagnostic Question Discussion

Your company has an application running on Google Cloud that is collecting data from thousands of physical devices that are globally distributed. Data is published to Pub/Sub and streamed in real time into an SSD Cloud Bigtable cluster via a Dataflow pipeline. The operations team informs you that your Cloud
Bigtable cluster has a hotspot, and queries are taking longer than expected. You need to resolve the problem and prevent it from happening in the future.

What should you do?

A. Advise your clients to use HBase APIs instead of NodeJS APIs.

B. Delete records older than 30 days.

C. Review your RowKey strategy and ensure that keys are evenly spread across the alphabet.

D. Double the number of nodes you currently have.

Diagnostic Question Discussion

Your company has an application running on Google Cloud that is collecting data from thousands of physical devices that are globally distributed. Data is published to Pub/Sub and streamed in real time into an SSD Cloud Bigtable cluster via a Dataflow pipeline. The operations team informs you that your Cloud
Bigtable cluster has a hotspot, and queries are taking longer than expected. You need to resolve the problem and prevent it from happening in the future.

What should you do?

A. Advise your clients to use HBase APIs instead of NodeJS APIs.

B. Delete records older than 30 days.

C. Review your RowKey strategy and ensure that keys are evenly spread across the alphabet.

D. Double the number of nodes you currently have.

Firestore

DocumentDB, DynamoDB, AppSync with Amplify DataStore

Firestore (Amazon DocumentDB / DynamoDB)

Flexible, horizontally scalable NoSQL cloud database to store and sync data

Key capabilities:

- Flexibility in querying
- Supports ACID Transactions
- Expressive querying
- Realtime updates (via AppSync)
- Offline support (via Amplify DataStore and AppSync)
- Designed to scale
- Data is indexed by default (manually indexed)

Firestore sample 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:

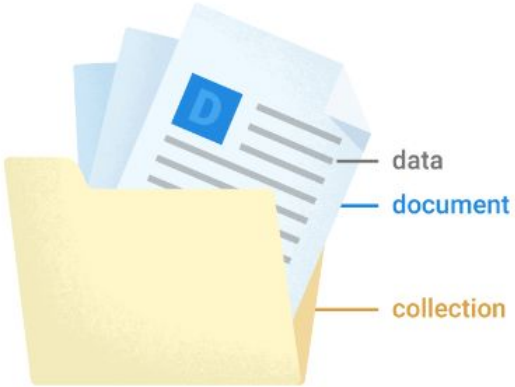
users

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: When to use?

Firestore is ideal for applications that rely on **highly available structured data** at scale.

Ideal Use Cases:

- Product catalogs that provide real-time inventory and product details for a retailer.
- User profiles that deliver a customized experience based on the user's past activities and preferences.
- Transactions based on ACID properties

Non-Ideal Use Cases:

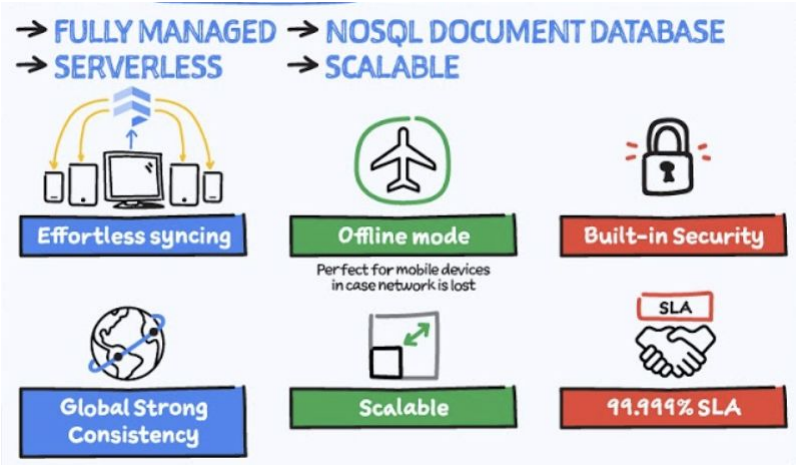
- OLTP relational database with full SQL support. *Consider:* Cloud SQL
- Data isn't highly structured or no need for ACID transactions. *Consider:* Cloud Bigtable
- Interactive querying in an online analytical processing (OLAP) system. *Consider:* BigQuery
- Unstructured data such as images or movies, *Consider:* Cloud Storage

Firestore: Datastore mode vs Firestore (native) mode

	Both	Native Mode (only)	Datastore Mode (only)
Data model	Strong consistency	Documents and collections	Entities, kinds, ancestor queries/results
Performance limits	No read limits	10K writes/sec 500 documents/txn	
API		Firestore (Documents)	Datastore (Entities)
Security	IAM	Firebase Rules	
Offline data persistence		**Yes**	
Real-time updates		Yes	

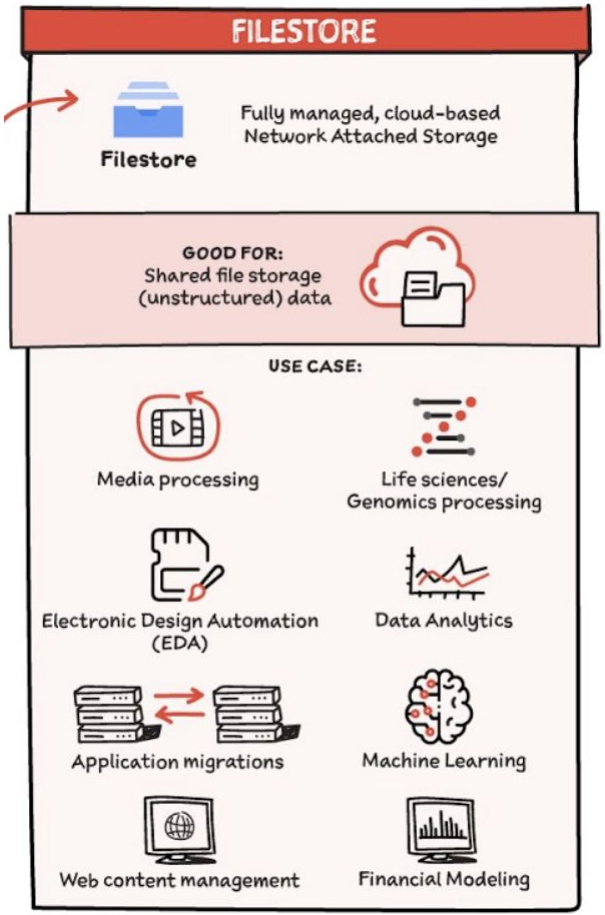
Firestore or Datastore - comparison

Firestore vs Filestore



… vs Firebase

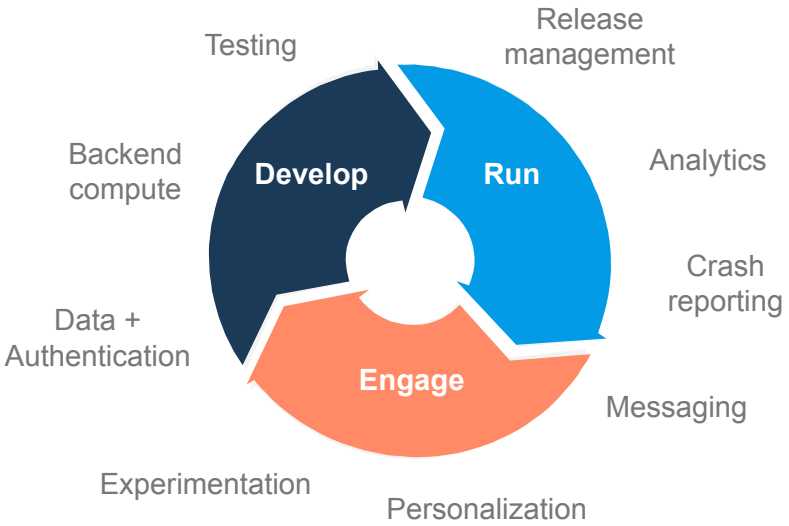
Exam Tip: *Firestore is a NoSQL Database, but Firebase is a development platform with a ton of additional features that uses Firestore. Make sure to differentiate between them!*



Firebase (AWS Amplify)
complete app development platform

Complete = it provides different products to:

- Build apps
- Test apps
- Implement authentication (Firebase Authentication can be a part of PCA exam on very high-level!)
- Run apps
- Run analytics
- Personalize apps
- And more…



Exam Tip: *Firestore is usually a part of Firebase-based app (for storing and syncing data)*

Diagnostic Question Discussion

Your development team has created a mobile game app. You want to test the new mobile app on Android and iOS devices with a variety of configurations. You need to ensure that testing is efficient and cost-effective.

What should you do?

A. Create Android and iOS VMs on Google Cloud, install the mobile app on the VMs, and test the mobile app.

B. Upload your mobile app to the Firebase Test Lab, and test the mobile app on Android and iOS devices.

C. Create Android and iOS containers on Google Kubernetes Engine (GKE), install the mobile app on the containers, and test the mobile app.

D. Upload your mobile app with different configurations to Firebase Hosting and test each configuration.

Diagnostic Question Discussion

Your development team has created a mobile game app. You want to test the new mobile app on Android and iOS devices with a variety of configurations. You need to ensure that testing is efficient and cost-effective.

What should you do?

A. Create Android and iOS VMs on Google Cloud, install the mobile app on the VMs, and test the mobile app.

B. Upload your mobile app to the Firebase Test Lab, and test the mobile app on Android and iOS devices.

C. Create Android and iOS containers on Google Kubernetes Engine (GKE), install the mobile app on the containers, and test the mobile app.

D. Upload your mobile app with different configurations to Firebase Hosting and test each configuration.

https://firebase.google.com/docs/test-lab

NoSQL database comparison

Type	Bigtable	Firestore	DynamoDB	Keyspaces	DocumentDB
Data Model	Columnar	Document	Document and Key-Value	Columnar	Document
Access Type	APIs	Document APIs	Document APIs	APIs	Document APIs
Scaling	Linear	Adaptive	Preprovisioned or Adaptive	Preprovisioned or Adaptive	Adaptive
Scalability	PBs	TBs	PBs	PBs	64 TB
Ideal For	Real Time, Adtech	Lookups	Session Stores, Lookups	Real Time, Cassandra compatibility	Content management, user profiles
Latency	Milliseconds	Seconds	Milliseconds	Milliseconds	Seconds
Architecture	Cluster based	Serverless	Serverless	Serverless	Cluster based

Storage options:

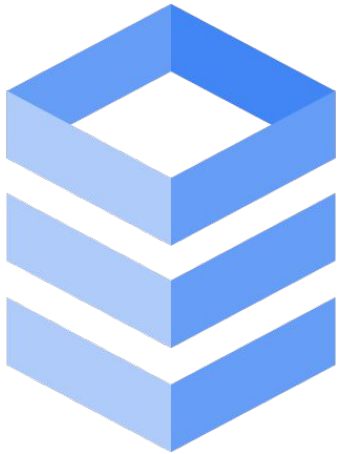
SQL (aka ‘relational’) databases

Cloud SQL

RDS, Aurora

Cloud SQL (Amazon RDS) is a managed RDBMS

- Offers MySQL, PostgreSQL, and SQL Server (as well as Oracle, MariaDB, and Aurora [MySQL or Postgres]) database platforms as a service.
- Automatic replication
- Managed backups
- Point in Time Recovery
- Vertical scaling (read and write) (Aurora can be serverless as well)
- Horizontal scaling (read)
- Google security

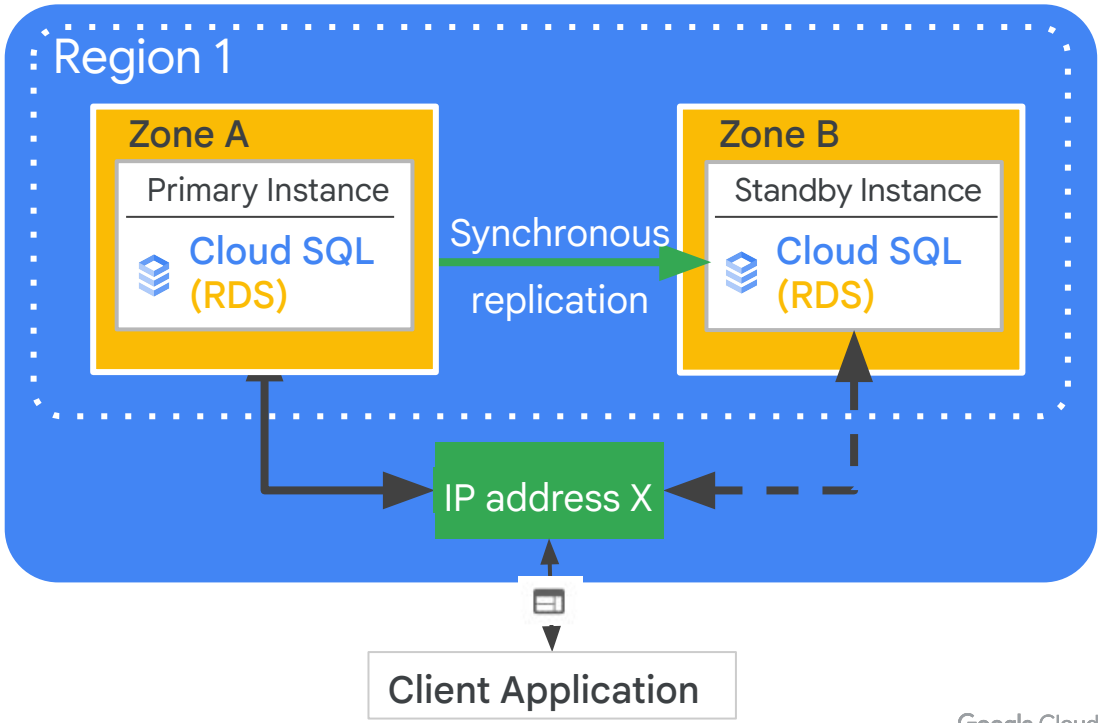


Cloud SQL vs Amazon RDS - key differences

Feature	Cloud SQL	Amazon RDS
Network and security	Use VPC firewalls, Cloud SQL Proxy, or allowlist IP addresses.	Use VPC security groups to control access from IP addresses and EC2 instances.
Encryption for external connections	Use Cloud SQL Proxy or SSL.	Use SSL.
Encryption at rest	Data is automatically encrypted at rest and in Google Cloud's networks.	You can enable encryption.
Read replica	You can create a read replica, external instance replica, and internal replica from external primary instances.	You can create a read replica, and leverage Amazon RDS Proxy for pooling RDS connections across replicas.
Backups	You can enable automatic backups.	Automatic backups are enabled by default.

Cloud SQL details

- HA configuration
- Backup service
- Import/export
- Scaling
  - Out: Read replicas
  - Up: Machine capacity

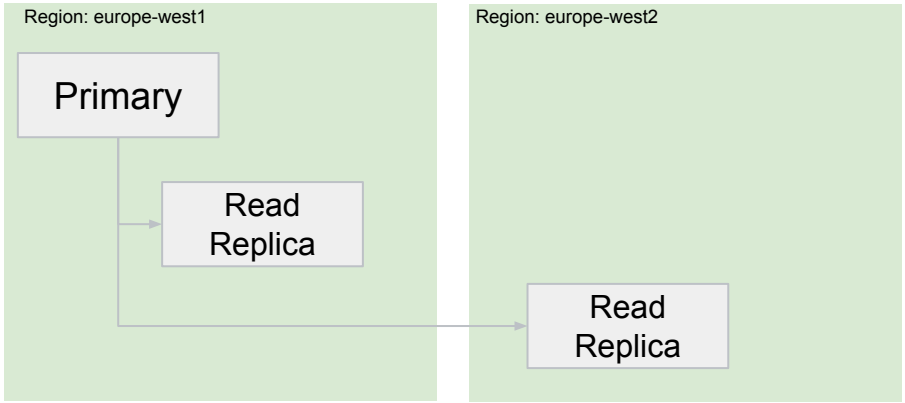


Cloud SQL: Read Replicas

Exam Tip: Docs on replicating TO external server

Use cases: Disaster Recovery / offload analytics workloads / migrate between platforms or regions

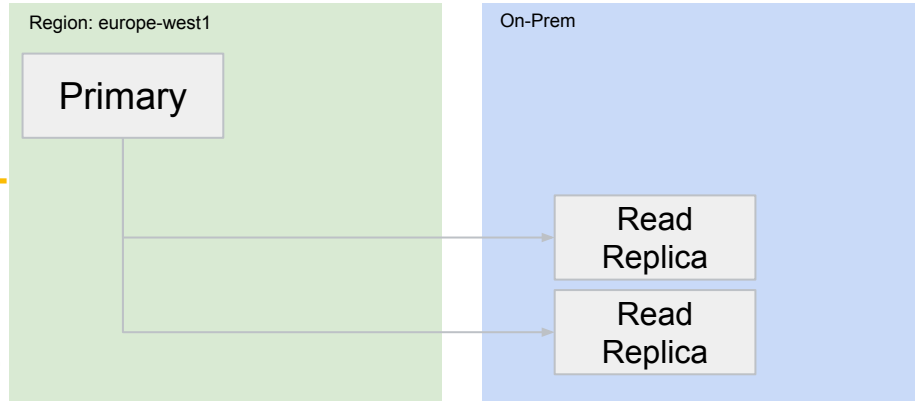
Read Replica GCP -> GCP



Benefits & Use Cases

- Additional Read capacity (read only)
- Analytics target (adding secondary indexes)
- Read replicas can be different machine types than primary (never less vcpus for postgres)
- Settings of primary are propagated to replicas incl. root pwd &user table changes
- No load balancing between replicas
- Mysql Parallel replication (read replica side)

External Read Replica GCP -> on-premises



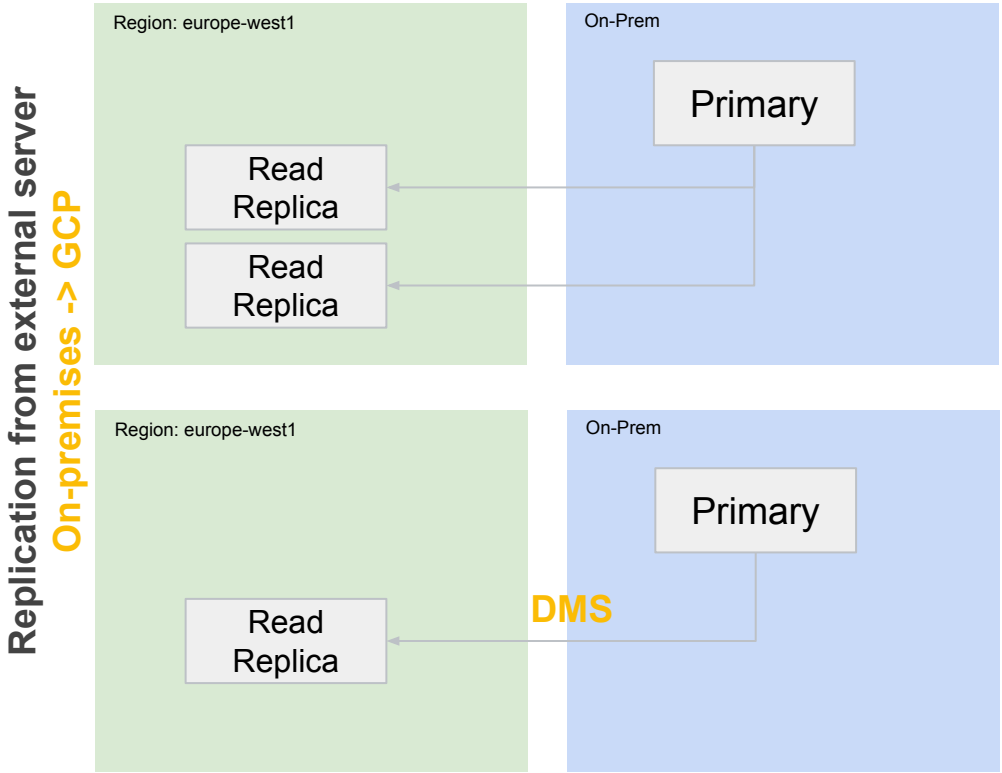
Benefits & Use Cases

- Reduce latency for external connection
- Analytics target
- Migration path to other platforms
- In case of e.g. network outage on-prem the replication lag might be too large and replicas need to be recreated

Cloud SQL: Read Replicas

Exam Tip: *Docs on replicating FROM external server*

Use cases: Disaster Recovery / offload analytics workloads / migrate between platforms or regions



MYSQL Benefits & Use Cases

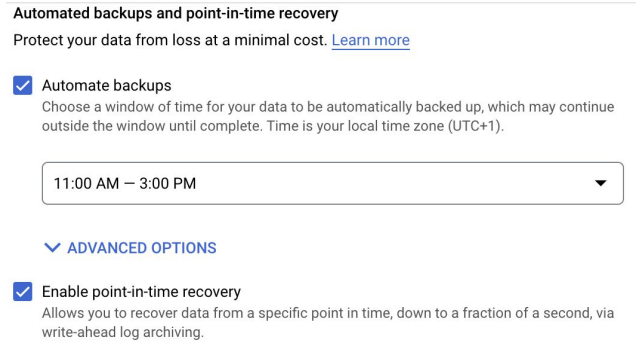
- Migration path to Cloud SQL with minimum downtime
- Data replication to GCP
- Offloading admin overhead of replicas to GCP
- Analytics target
- Parallel replication (read replica)

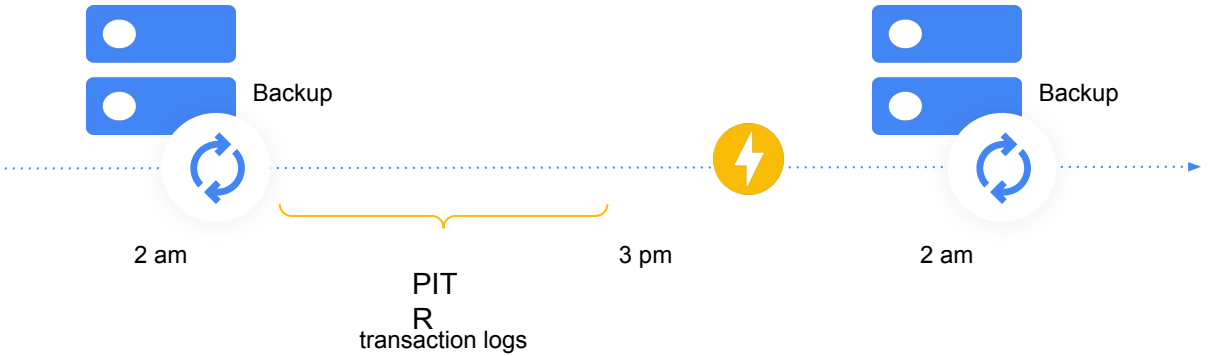
POSTGRES - DMS

- Use DMS to replicate from an external DB Server to a Cloud SQL Read replica (One-off Migration or Continuous cdc replication)
- DB Source can be self-managed DB (on-prem or IaaS), Aws Rds, Aurora, Cloud SQL

Cloud SQL: Point-in-time recovery

recover an instance to a specific point in time





Database Migration Service (DMS) makes migrations to Cloud SQL easier and faster

Serverless, secure, and offered at no additional charge

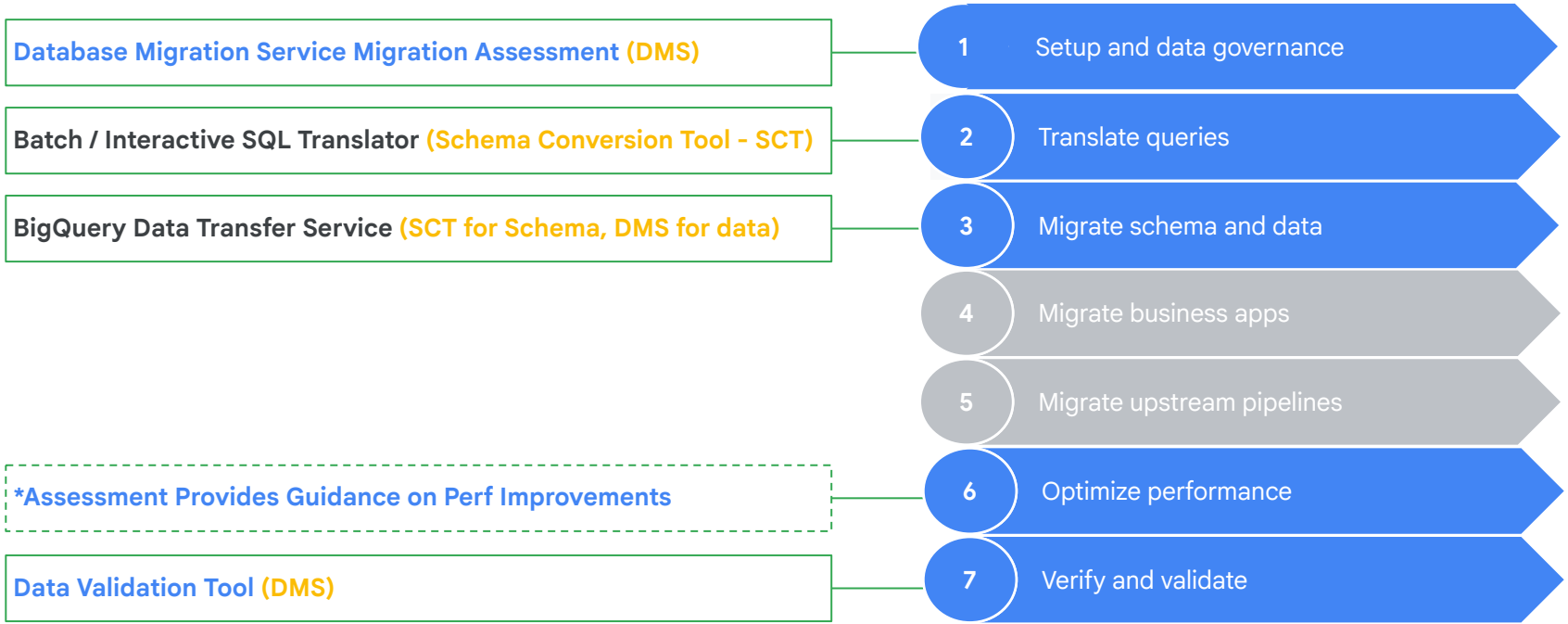
Supports migrations of MySQL PostgreSQL and SQL Server (Oracle, MariaDB, MongoDB, Db2, etc.) databases to Cloud SQL (EC2, RDS, Redshift, Redshift, DynamoDB, etc.)

Utilizes native replication (homogeneous) for higher fidelity and greater reliability (or native SCT and DMS capabilities [heterogeneous])

More than 85% of migrations are underway in less than an hour

A majority of customers are migrating databases from other clouds

Database Migration Service overview



Diagnostic Question Discussion

You need to set up Microsoft SQL Server on GCP. Management requires that there's no downtime in case of a data center outage in any of the zones within a GCP region.

What should you do?

A. Configure a Cloud Spanner instance with a regional instance configuration.

B. Set up SQL Server on Compute Engine, using Always On Availability Groups using Windows Failover Clustering. Place nodes in different subnets.

C. Configure a Cloud SQL instance with high availability enabled.

D. Set up SQL Server Always On Availability Groups using Windows Failover Clustering. Place nodes in different zones.

Diagnostic Question Discussion

You need to set up Microsoft SQL Server on GCP. Management requires that there's no downtime in case of a data center outage in any of the zones within a GCP region.

What should you do?

A. Configure a Cloud Spanner instance with a regional instance configuration.

B. Set up SQL Server on Compute Engine, using Always On Availability Groups using Windows Failover Clustering. Place nodes in different subnets.

C. Configure a Cloud SQL instance with high availability enabled.

D. Set up SQL Server Always On Availability Groups using Windows Failover Clustering. Place nodes in different zones.

Diagnostic Question Discussion

During a high traffic portion of the day, one of your relational databases crashes, but the replica is never promoted to a master. You want to avoid this in the future.

What should you do?

A. Use a different database

B. Choose larger instances for your database

C. Create snapshots of your database more regularly

D. Implement routinely scheduled failovers of your databases

Diagnostic Question Discussion

During a high traffic portion of the day, one of your relational databases crashes, but the replica is never promoted to a master. You want to avoid this in the future.

What should you do?

A. Use a different database

B. Choose larger instances for your database

C. Create snapshots of your database more regularly

D. Implement routinely scheduled failovers of your databases

https://cloud.google.com/solutions/cloud-sql-mysql-disaster-recovery-complete-failover-fallback

Diagnostic Question Discussion

You are implementing a single Cloud SQL MySQL second-generation database that contains business-critical transaction data. You want to ensure that the minimum amount of data is lost in case of logical data inconsistency.

Which two features should you implement? (Choose two.)

A. Sharding

B. Read replicas

C. Binary logging

D. Automated backups

E. Semisynchronous replication

Diagnostic Question Discussion

You are implementing a single Cloud SQL MySQL second-generation database that contains business-critical transaction data. You want to ensure that the minimum amount of data is lost in case of logical data inconsistency.

Which two features should you implement? (Choose two.)

A. Sharding

B. Read replicas

C. Binary logging

D. Automated backups

E. Semisynchronous replication

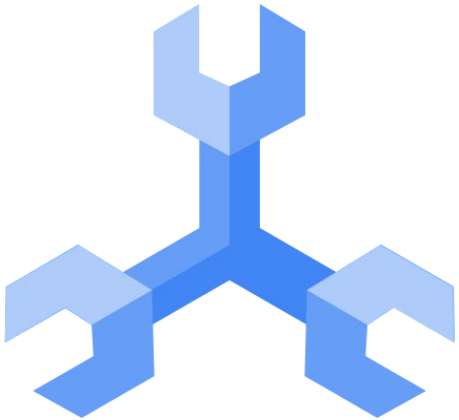
Cloud Spanner

Aurora Global DB

Cloud Spanner (Amazon Aurora) is a horizontally scalable RDBMS

Cloud Spanner (Amazon Aurora) supports:

- Automatic replication.
- Multi (single) region writes and multi region reads.
  - Strong global consistency.
- Managed instances with high availability.
- SQL (ANSI 2011 with extensions or MySQL or Postgres compatibility).



Cloud Spanner vs Amazon Aurora - key differences

Feature	Google Cloud Spanner	Amazon Aurora
Write method	Multi-master write	Single-master write
Column	Supports regular and wide columns	Supports regular columns
Serverless?	Fully serverless	Offers a serverless option
Postgre complaint?	For reading only	For reading and writing

Cloud Spanner, Cloud SQL vs Amazon RDS, Aurora

	RDS	Aurora	Cloud SQL	Cloud Spanner
Consistency	Strong Regional	Strong Regional, Eventual Global (~ 1 sec)	Strong Regional	Strong Global
Availability	Failover	Failover	Failover	High
Scalability	Vertical	Vertical and Horizontal and Serverless	Vertical	Horizontal
Replication	Configurable	Configurable	Configurable	Automatic
Ideal for	Few TBs	Few TBs	Few TBs	PBs

Diagnostic Question Discussion

Your organization is developing an application that will manage payments and online bank accounts located around the world. The most critical requirement for your database is that each transaction is handled consistently. Your organization anticipates almost unlimited growth in the amount of data stored.

Which Google Cloud product should your organization choose?

A. Cloud SQL

B. Cloud Storage

C. Cloud Firestore

D. Cloud Spanner

Diagnostic Question Discussion

Your organization is developing an application that will manage payments and online bank accounts located around the world. The most critical requirement for your database is that each transaction is handled consistently. Your organization anticipates almost unlimited growth in the amount of data stored.

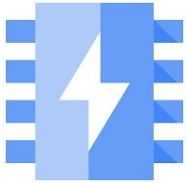
Which Google Cloud product should your organization choose?

A. Cloud SQL

B. Cloud Storage

C. Cloud Firestore

D. Cloud Spanner



Memorystore

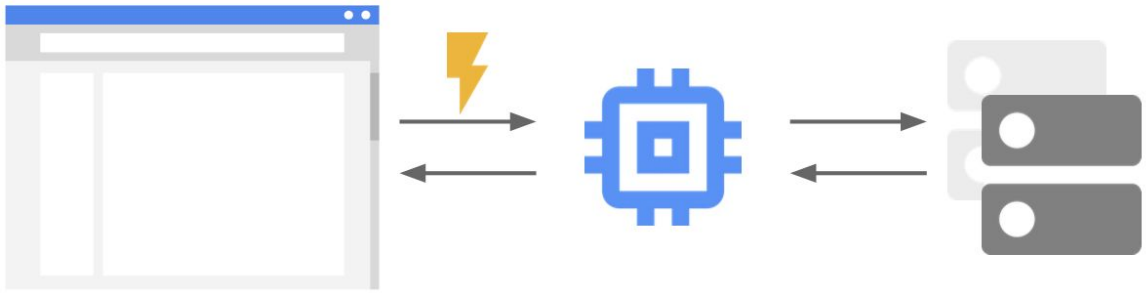
Amazon ElastiCache

Memorystore (Amazon ElastiCache)

- 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 (30) Gbps
- Use cases:
  - Lift and shift of Redis, Memcached
  - Anytime need a managed service for cached data

Memorystore

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
```

Storage options:

Data Warehouse

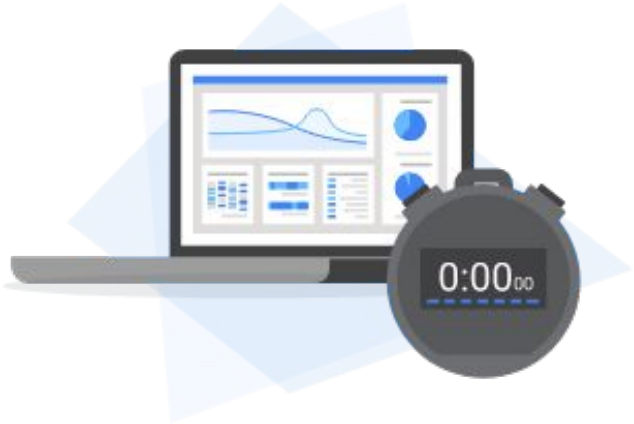
Bigquery

Amazon Redshift

BigQuery (Redshift) is serverless, highly scalable, and cost-effective cloud data warehouse

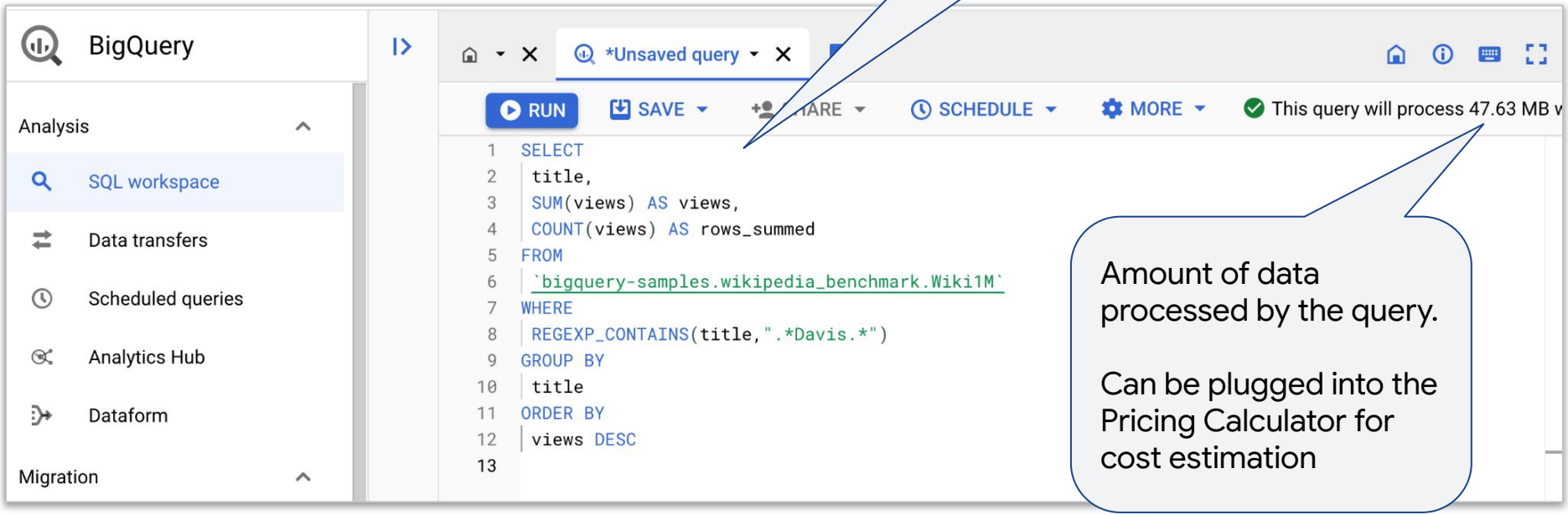
BigQuery = Redshift Spectrum or Redshift Serverless (both serverless) or standard Redshift (cluster based)

- Fully (semi) managed
- Petabyte scale
- SQL interface
- Very fast (separated compute and storage with Redshift Spectrum [data in S3] and Redshift Serverless [native database storage])



BigQuery

BigQuery User Interface



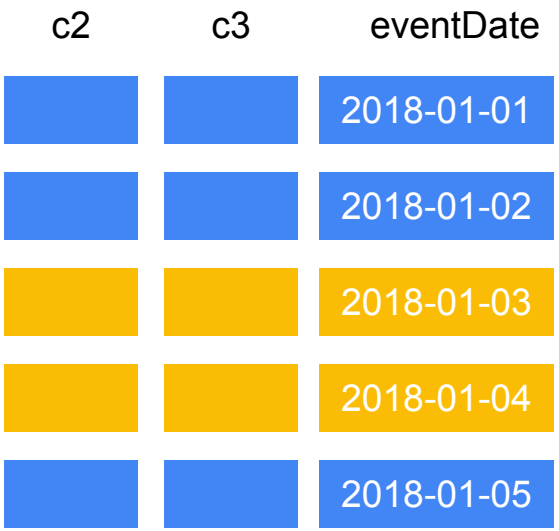
BigQuery: Table Partitioning

Partitioning versus sharding:

- Table sharding is the practice of storing data in multiple tables, using a naming prefix such as [PREFIX]_YYYYMMDD. Partitioning is recommended over table sharding, because partitioned tables perform better.

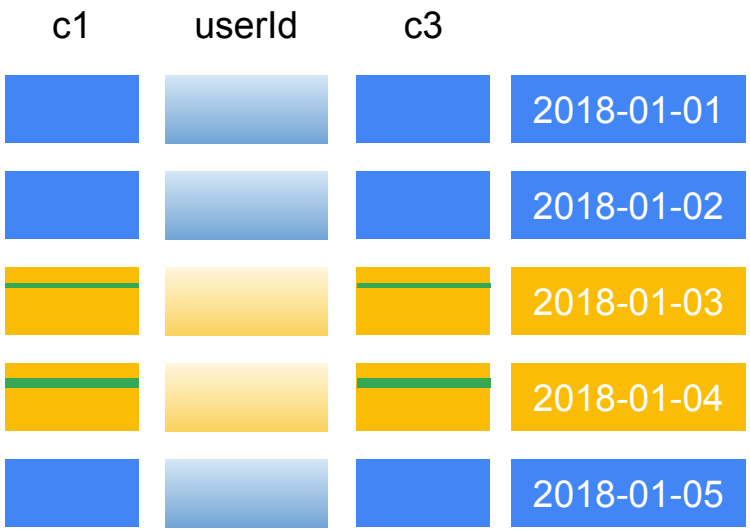
You can partition BigQuery tables by:

- Time-unit column: Tables are partitioned based on a TIMESTAMP, DATE, or DATETIME column in the table.
- Ingestion time: Tables are partitioned based on the timestamp when BigQuery ingests the data.
- Integer range: Tables are partitioned based on an integer column.




```
SELECT * FROM ...
WHERE eventDate BETWEEN
“2018-01-03” AND “2018-01-04”
```

BigQuery: Table Clustering



SELECT c1, c3 FROM ... WHERE userId BETWEEN 52 and 63
AND eventDate BETWEEN “2018-01-03” AND “2018-01-04”

BigQuery: table partitioning vs clustering

Decision making

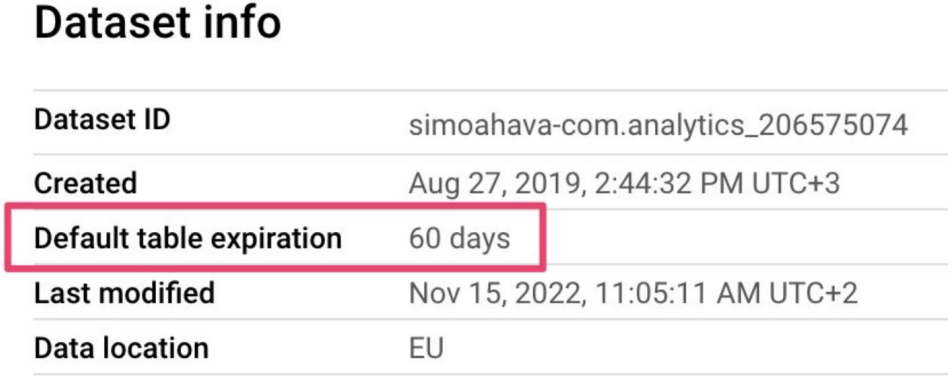
- Clustering gives you more granularity than partitioning alone allows
- Use clustering if your queries commonly use filters or aggregation against multiple particular columns.

Use case	Recommendation
You're using on-demand pricing and require strict cost guarantees before running queries.	Partitioned tables
Your segment size is less than 1 GB after partitioning the table.	Clustered tables
You require a large number of partitions beyond the BigQuery limits	Clustered tables
Frequent mutations in your data modify a large number of partitions.	Clustered tables
You frequently run queries to filter data on certain fixed columns.	Partitions plus clustering

BigQuery: table/partition (automatic) data expiration

Can be set for dataset / table / partition

Best practice for data lifecycle management. Expiration in BigQuery automatically implements retention policy.



- Dataset expiration
  - = “default table expiration time” for a dataset
- Table expiration
  - If Dataset expiration is set, each table inherits this setting by default
- Partition expiration:
  - The setting applies to all partitions in the table, but is calculated independently for each partition based on the partition time.
  - At any point after a table is created, you can update the table's partition expiration

BigQuery: Controlling access to datasets

You can grant access at the following BigQuery resource levels:

- organization or Google Cloud project level
- dataset level
- table or view level
  a. Authorized Views
- You can also restrict access to data on more granular level by using the following methods:
  a. column-level access control
  b. dynamic data masking (aka “some columns may be hidden, depending on privileges”)
     i. Works together with column-level security.
     ii. no need to modify existing queries by excluding the columns that the user cannot access
  c. row-level security (aka “some rows may be hidden, depending on privileges”)
     i. One table can have multiple row-level access policies. Row-level access policies can coexist on a table with column-level security as well as dataset-level, table-level, and project-level access controls.

BigQuery: Controlling access to datasets

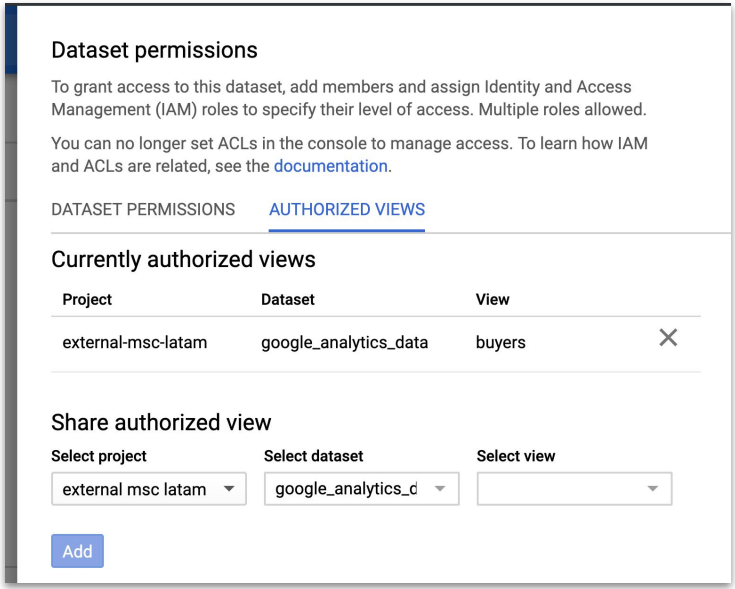
Authorized Views

1. **View:** View is a virtual table defined by a SQL query. When you create a view, you query it in the same way you query a table

2. **Query:** When a user queries the view, the query results contain data only from the tables and fields specified in the query that defines the view.

3. **Authorized Views:** An authorized view allows you to share query results with particular users and groups without giving them access to the underlying tables.

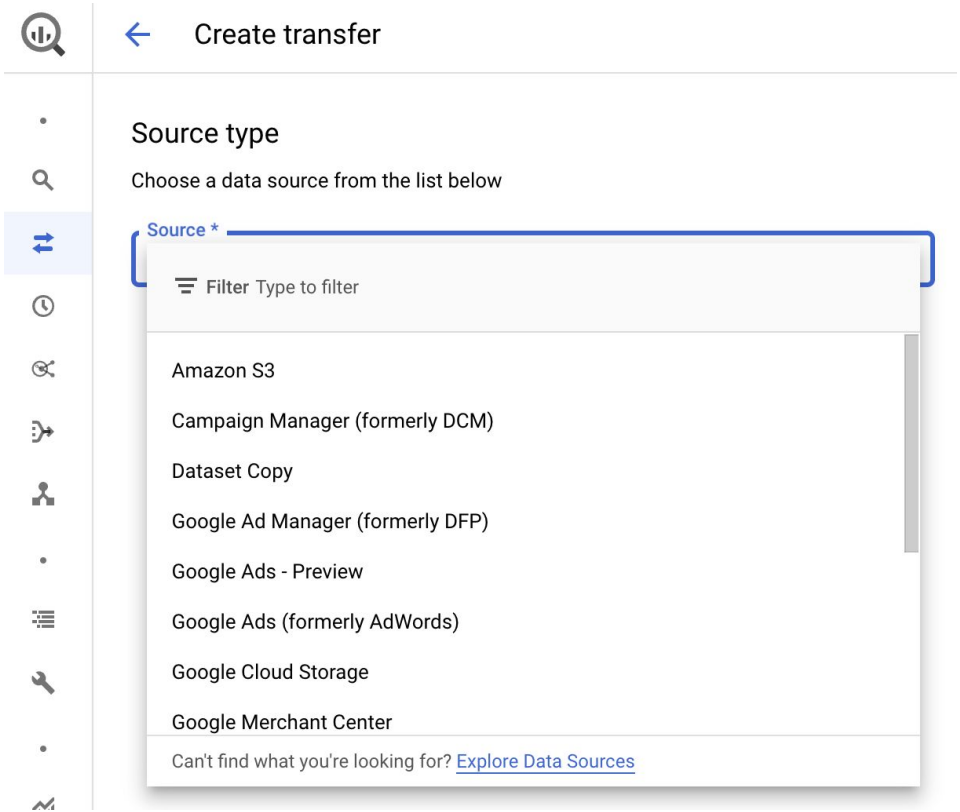
Exam Tip: *Authorized Views were especially useful when there were no table/column-level permissions. However, they're still often-used way to selectively share access to datasets (and they pop up on the exam!). MAKE SURE TO UNDERSTAND HOW TO CREATE AND SHARE SUCH A VIEW.*



BigQuery: Data Transfer Service

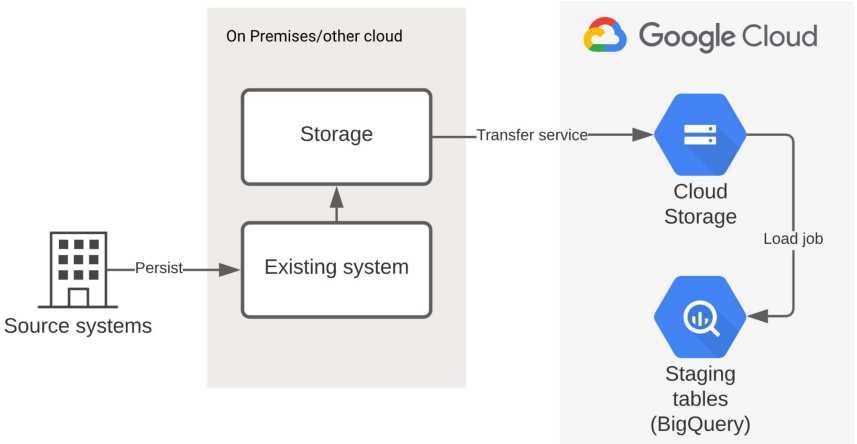
Mostly useful for regular data transfers to BigQuery

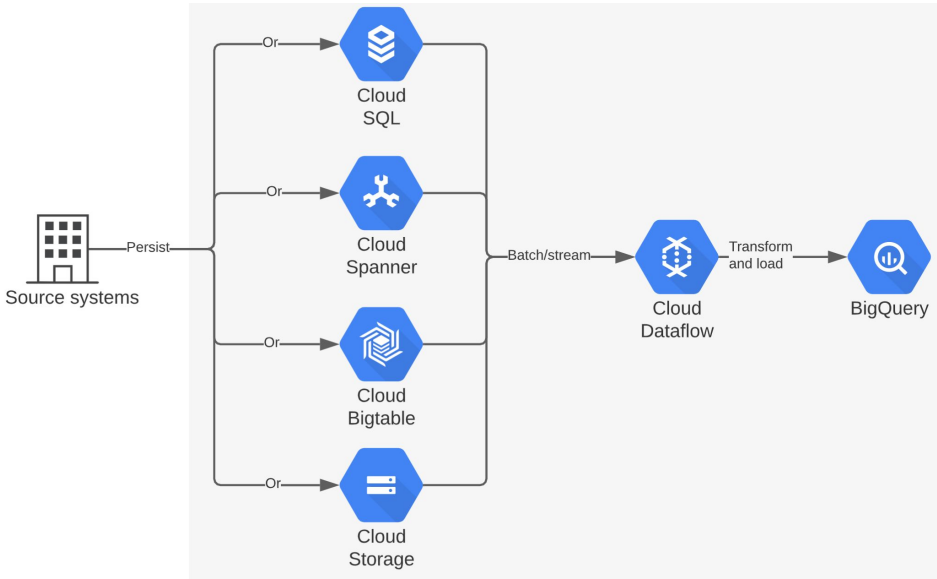
- BigQuery Data Transfer Service automates data movement from various sources into BigQuery on a scheduled, managed basis.
- You can initiate data backfills to recover from any outages or gaps.



BigQuery: Batch vs Streaming inserts

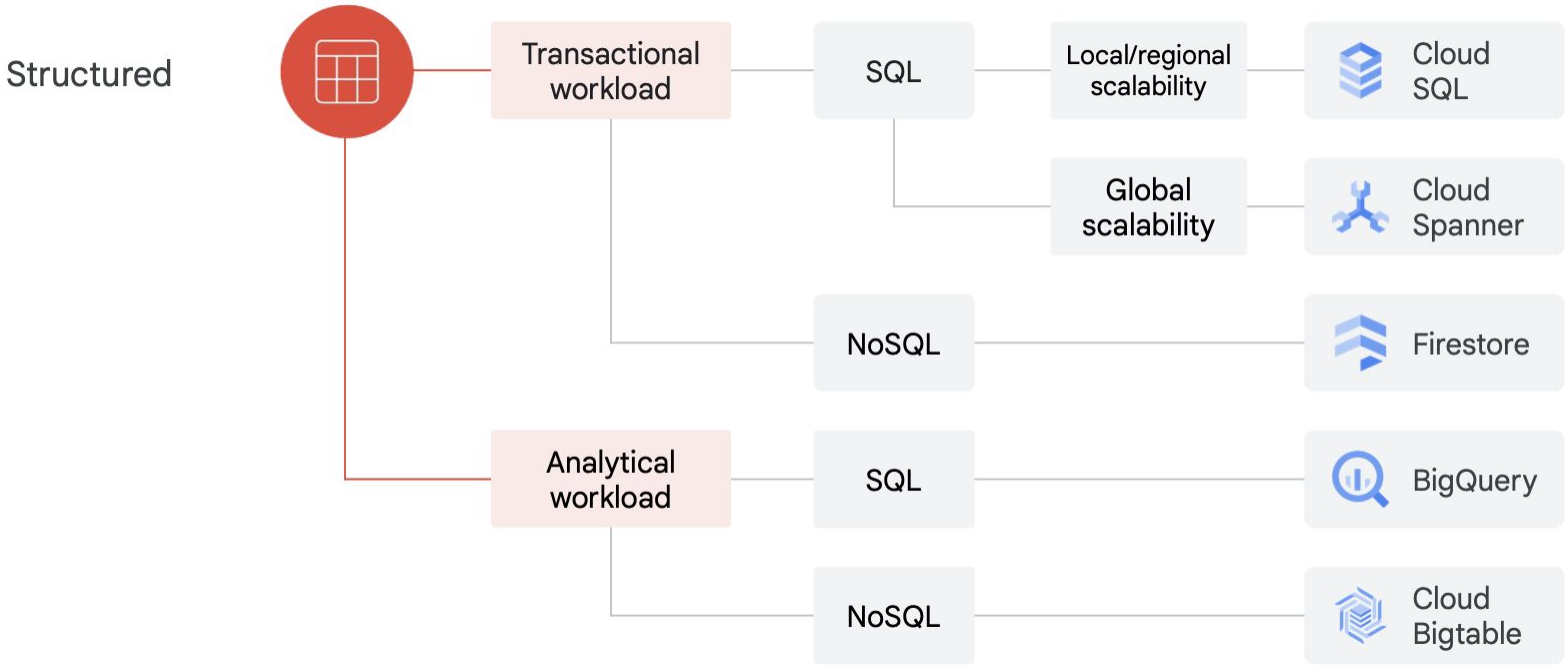
Most common architectures





Exam Tip: *There is additional cost for streaming (both inserts and reads) in BigQuery.*

GCP: storage service decision tree



Diagnostic Question Discussion

To be compliant with European GDPR regulation, Cymbal Bank is required to delete data generated from its European customers after a period of 36 months when it contains personal data. In the new architecture, this data will be stored in both Cloud Storage and BigQuery.

What should you do?

A. Create a BigQuery table for the European data, and set the table retention period to 36 months. For Cloud Storage, use gsutil to enable lifecycle management using a DELETE action with an Age condition of 36 months.

B. Create a BigQuery table for the European data, and set the table retention period to 36 months. For Cloud Storage, use gsutil to create a SetStorageClass to NONE action when with an Age condition of 36 months.

C. Create a BigQuery time-partitioned table for the European data, and set the partition expiration period to 36 months. For Cloud Storage, use gsutil to enable lifecycle management using a DELETE action with an Age condition of 36 months.

D. Create a BigQuery time-partitioned table for the European data, and set the partition expiration period to 36 months. For Cloud Storage, use gsutil to create a SetStorageClass to NONE action with an Age condition of 36 months.

Diagnostic Question Discussion

To be compliant with European GDPR regulation, Cymbal Bank is required to delete data generated from its European customers after a period of 36 months when it contains personal data. In the new architecture, this data will be stored in both Cloud Storage and BigQuery.

What should you do?

A. Create a BigQuery table for the European data, and set the table retention period to 36 months. For Cloud Storage, use gsutil to enable lifecycle management using a DELETE action with an Age condition of 36 months.

B. Create a BigQuery table for the European data, and set the table retention period to 36 months. For Cloud Storage, use gsutil to create a SetStorageClass to NONE action when with an Age condition of 36 months.

C. Create a BigQuery time-partitioned table for the European data, and set the partition expiration period to 36 months. For Cloud Storage, use gsutil to enable lifecycle management using a DELETE action with an Age condition of 36 months.

D. Create a BigQuery time-partitioned table for the European data, and set the partition expiration period to 36 months. For Cloud Storage, use gsutil to create a SetStorageClass to NONE action with an Age condition of 36 months.

Make sure to...

Enjoy the journey as much as the destination!