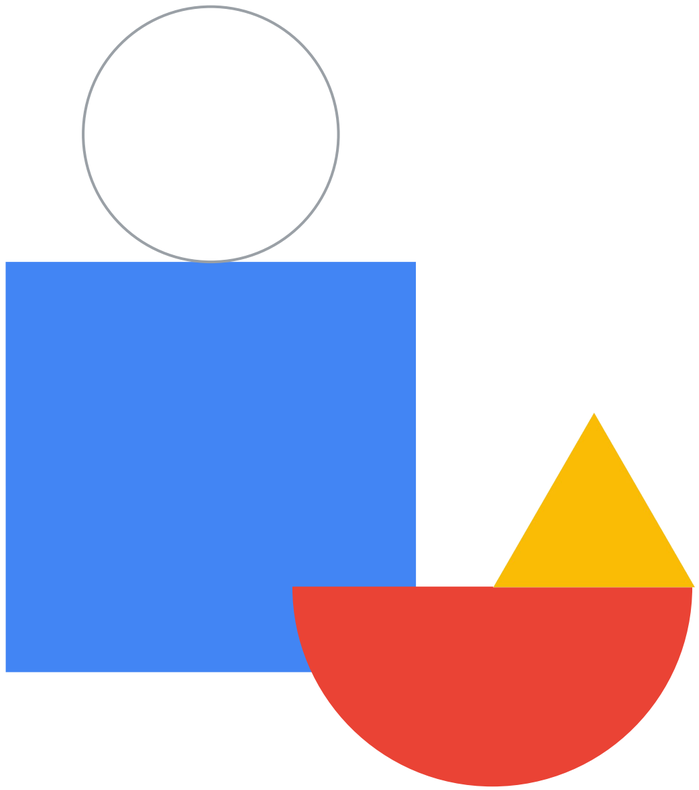
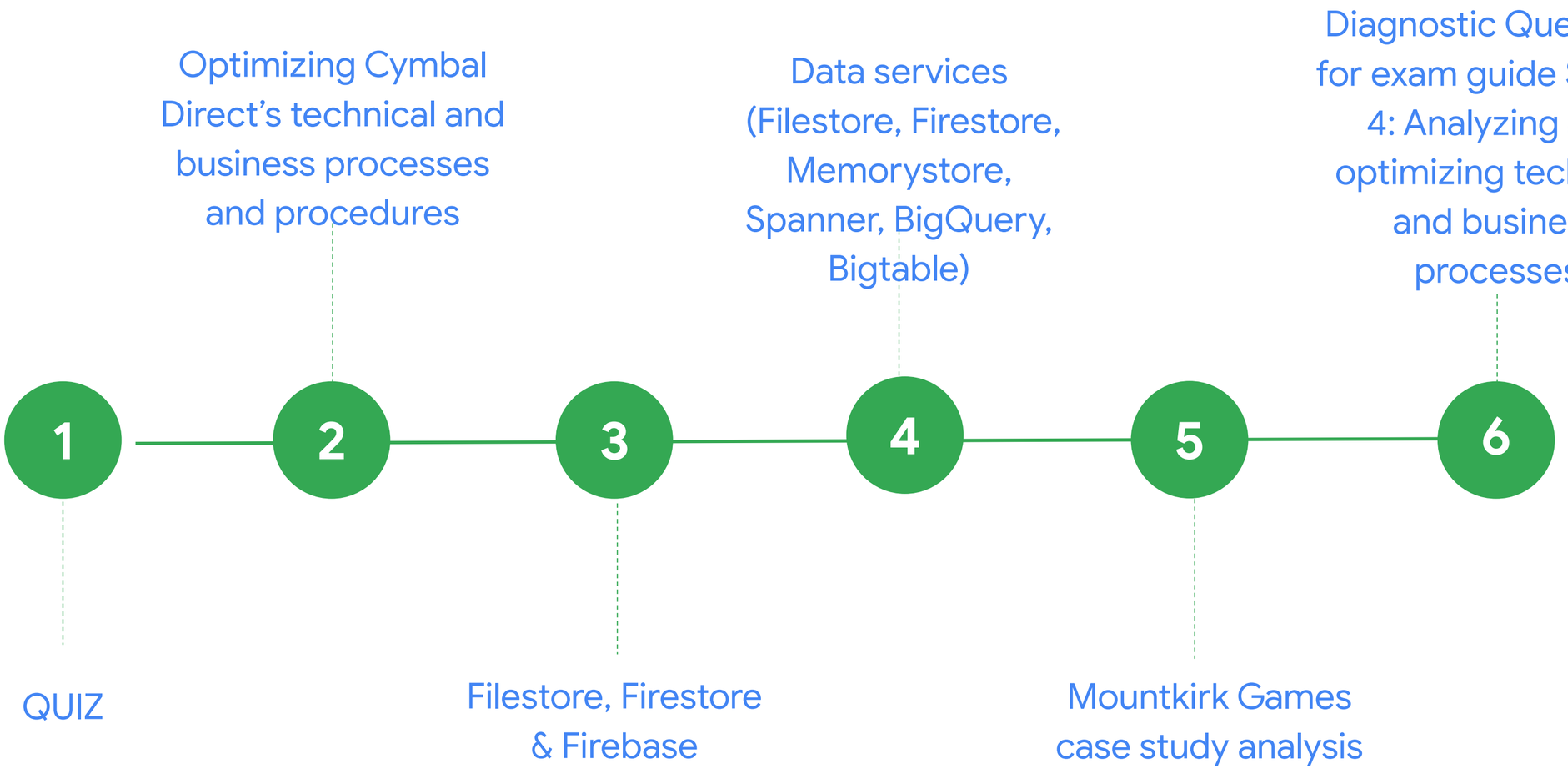
Google Cloud



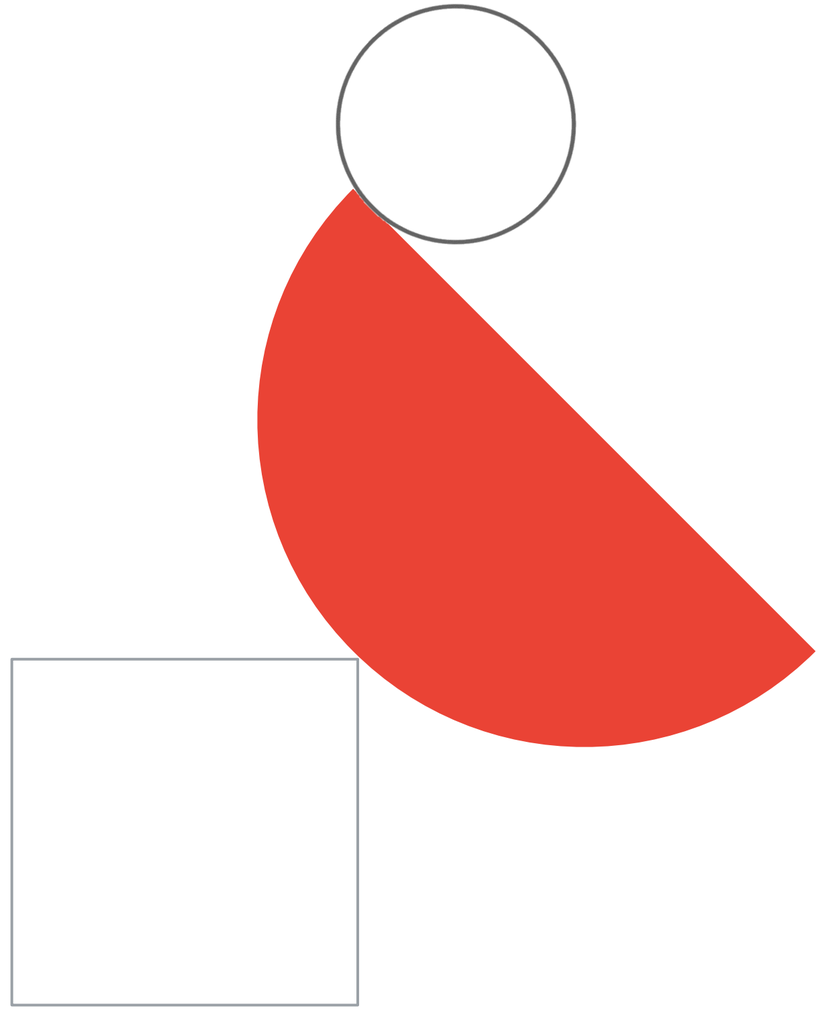
# Preparing for Your Professional Cloud Architect Journey

Module 4: Analyzing and Optimizing Technical and Business Processes

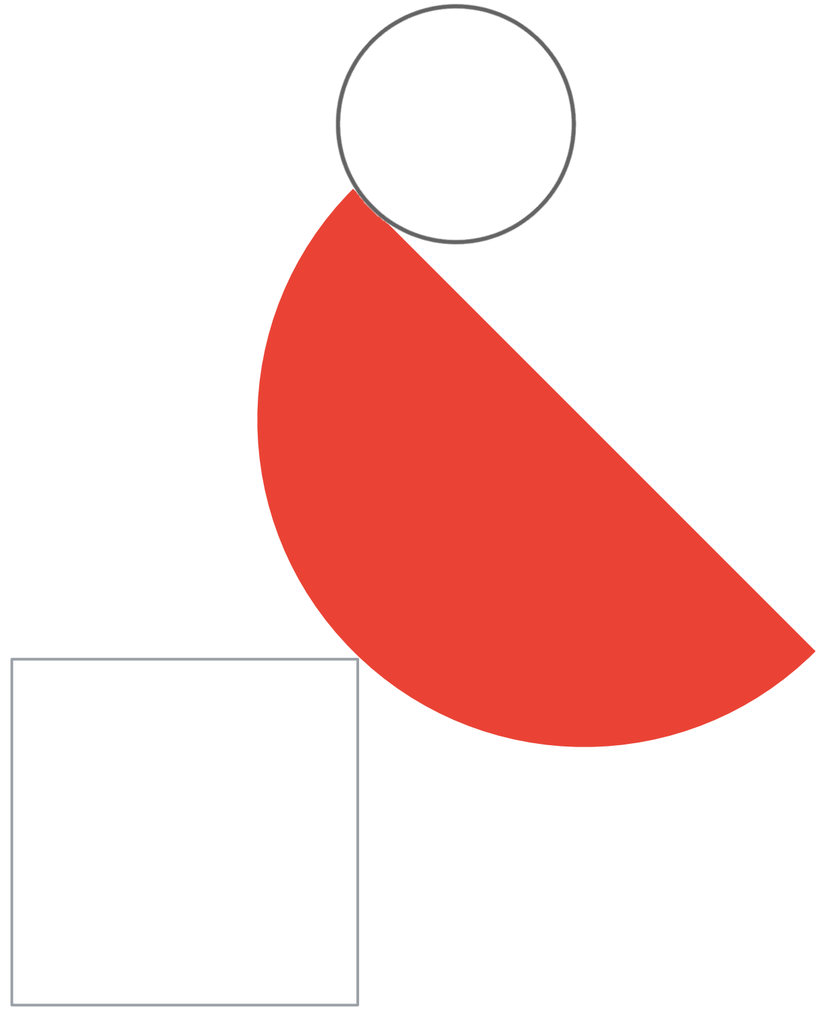
# Week 5 agenda



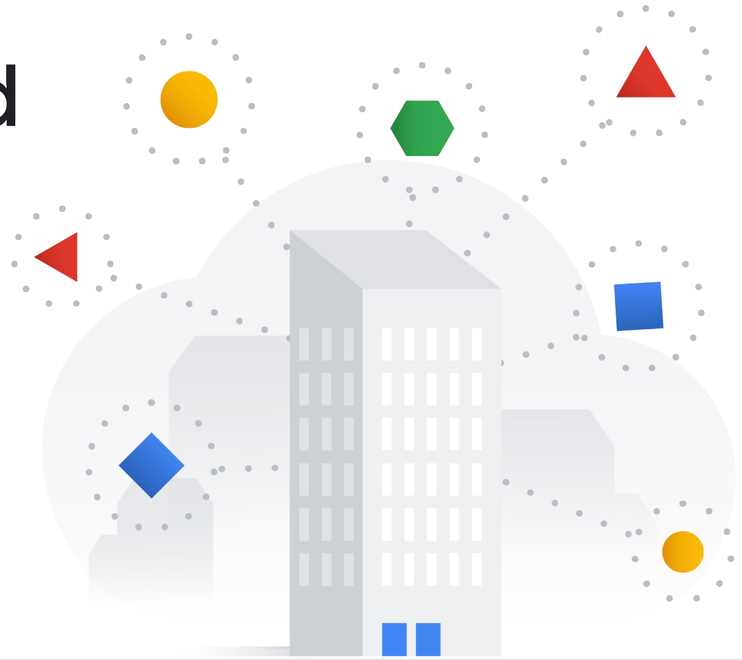
# QUIZ time!



# Optimizing Cymbal Direct's technical and business processes and procedures



# Your role in optimizing business and technical processes



- Analyzing and defining technical processes
- Analyzing and defining business processes
- Developing procedures to ensure reliability of solutions in production

Cymbal Direct

# Business Requirements

- Cymbal Direct’s management wants to make sure that they can easily scale to handle additional demand when needed, so they can feel comfortable with expanding to more test markets.
- Streamline development for application modernization and new features/products.
- Ensure that developers spend as much time on core business functionality as possible, and not have to worry about scalability wherever possible.
- Allow for partners to order directly via API
- Get a production version of the social media highlighting service up and running, and ensure no inappropriate content

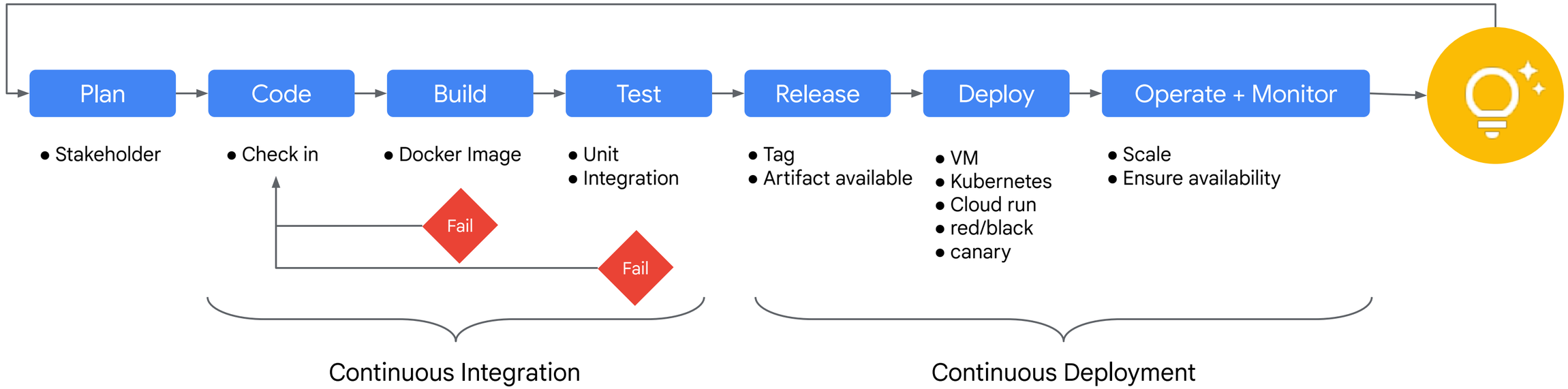
# Technical Requirements

- Move to managed services wherever possible
- Ensure that developers can deploy container based workloads to testing and production environments in a highly scalable environment.
- Standardize on containers where possible, but also allow for existing virtualization infrastructure to run as-is without a re-write, so it can be slowly refactored over-time
- Securely allow partner integration
- Allow for streaming of IoT data from drones

# Process optimization

The current build process at Cymbal Direct is:

- Package monolithic application with its dependencies
- Check it in and notify the QA team they need to test it
- Stress test the application to ensure it performs well
- Build a VM image for deployment

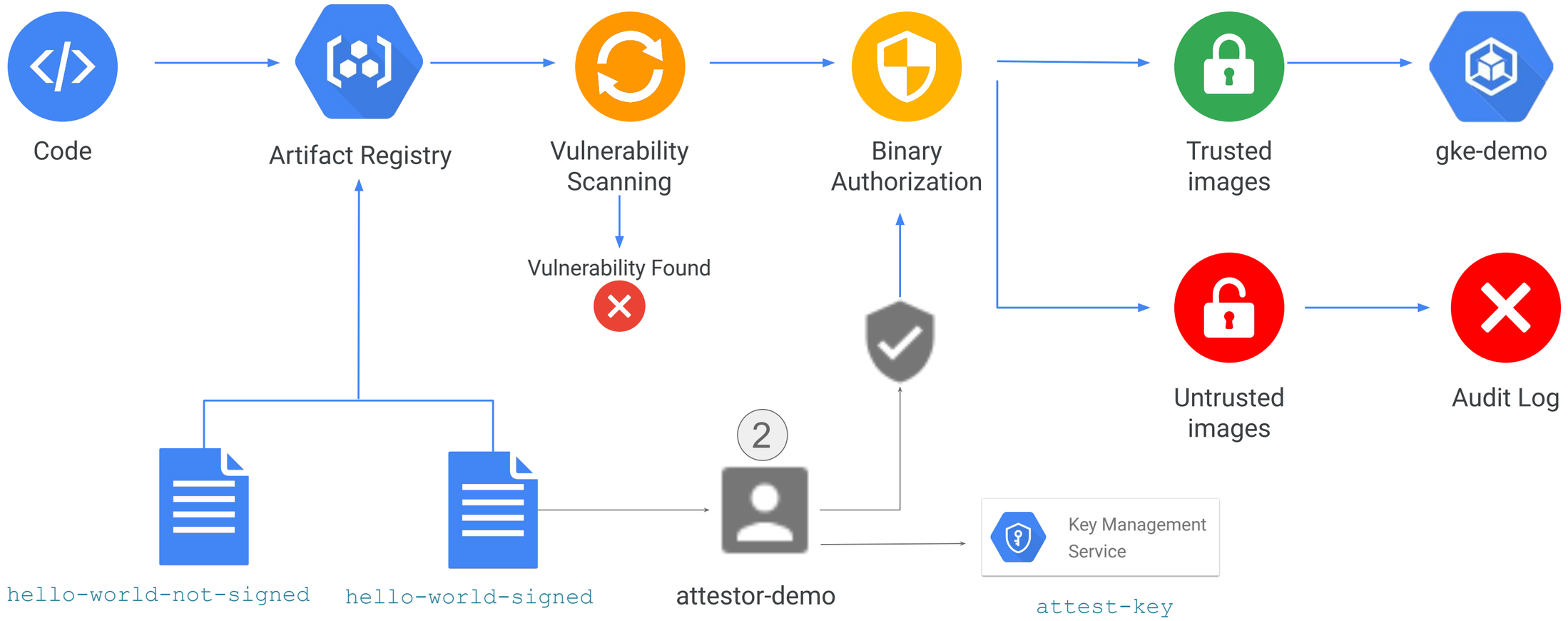


# Process optimization

Requirements not met:

- development is streamlined
- developers focus on core business functionality
- Move to managed services wherever possible
- deploy container based workloads

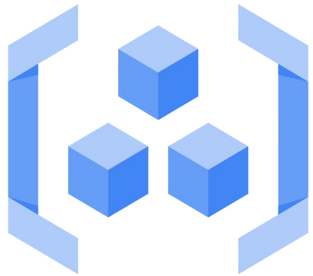
# Example of end-to-end CI/CD pipeline




Google Cloud

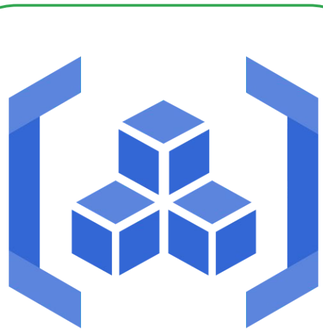
# Container Registry vs Artifact Registry

- Container Registry is currently in [maintenance mode](#) Although it is still available and supported as a Google Enterprise API, it won't see new any features
- **Artifact Registry** is the successor and the recommended solution
- Artifact Registry covers all use cases of container registry and can be used for additional packages like maven, npm, python, etc.
- See more info on how to migrate here https://cloud.google.com/artifact-registry/docs/transition/transition-from-gcr



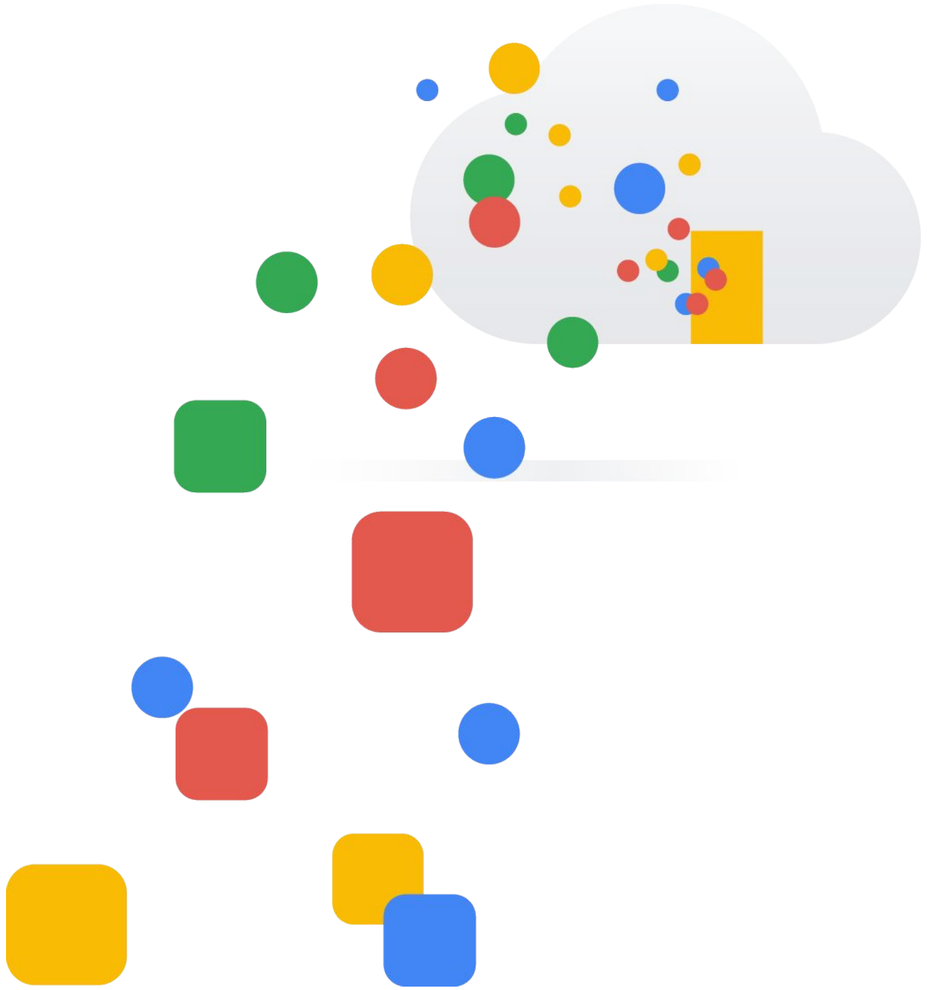
Container Registry

**vs**



Artifact Registry

prefer this for new projects

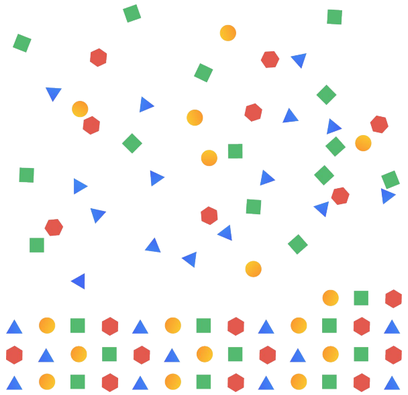


# Process optimization

New Process

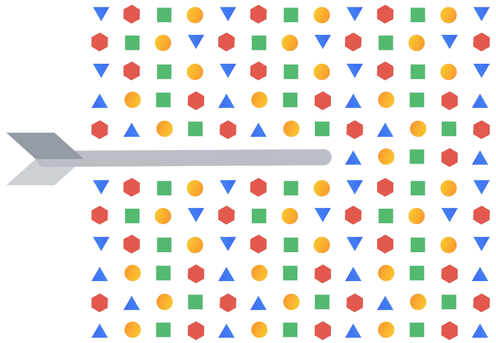
- New features are implemented as microservices in Docker containers
- Code check-in triggers CI/CD pipeline w/ automatic test & release
- Code is deployed to Cloud Run

# Developing Cymbal Direct's procedures to ensure solution reliability



## Chaos Engineering

- Creates a culture of reliability
- Crashes systems intentionally to build resiliency
- **Service Mesh can help you here!**

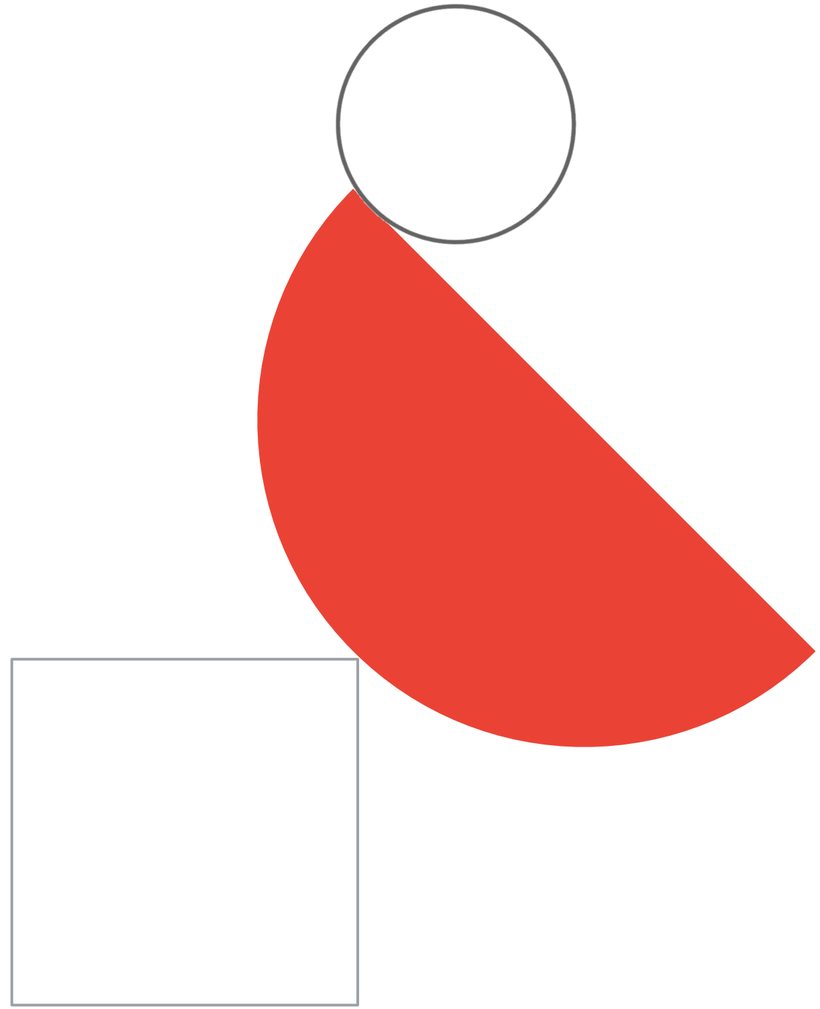


## Penetration testing

- Mimics the behavior of hackers to attack your own environment

"If you plan to evaluate the security of your Cloud Platform infrastructure with penetration testing, you are not required to contact us."

# Filestore

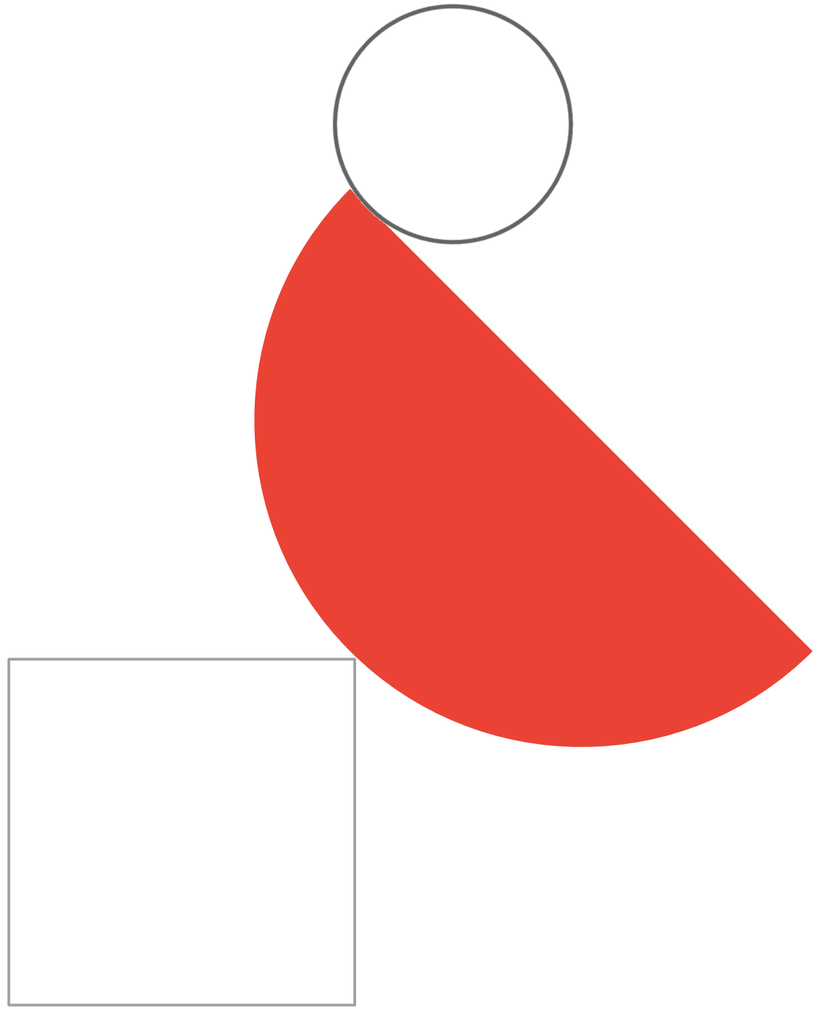


# Filestore

## Managed NFS, NOT a database

| | Filestore Basic (GA) | Filestore High Scale (Public Preview) | Filestore Enterprise (GA) |
|---|---|---|---|
| **Workloads** | File sharing, Software Dev, and Web Hosting | HPC, Financial Modeling, Pharma, and Analytics | SAP, GKE, and ‘Lift & Shift Apps” |
| **Capacity** | 1 - 64 TiB | 10 - 100 TiB | 1 - 10 TiB |
| **Scale** | Scale-up | Scale-out | Scale-out |
| **Capacity Management** | Grow | **Grow & Shrink** | **Grow & Shrink** |
| **Max Performance (Throughput \| IOPS)** | 1.2GiB/s \| 60k | 26GiB/s \| 920k | 1.2GiB/s \| 120k |
| **Data Protection** | **Backups** | None | **Snapshots** |
| **Availability SLA** | 99.9% | 99.9% | **99.99%** |

# Firestore



# 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<br>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 #GCPSketchnote

@PVERGADIA THECLOUDGIRL.DEV

06.22.2021

What's the secret large companies use to build data-driven apps quickly?

Yes! Build robust apps in half the time with...

Firestore!

## What is Firestore?

→ FULLY MANAGED → NOSQL DOCUMENT DATABASE
→ SERVERLESS → SCALABLE

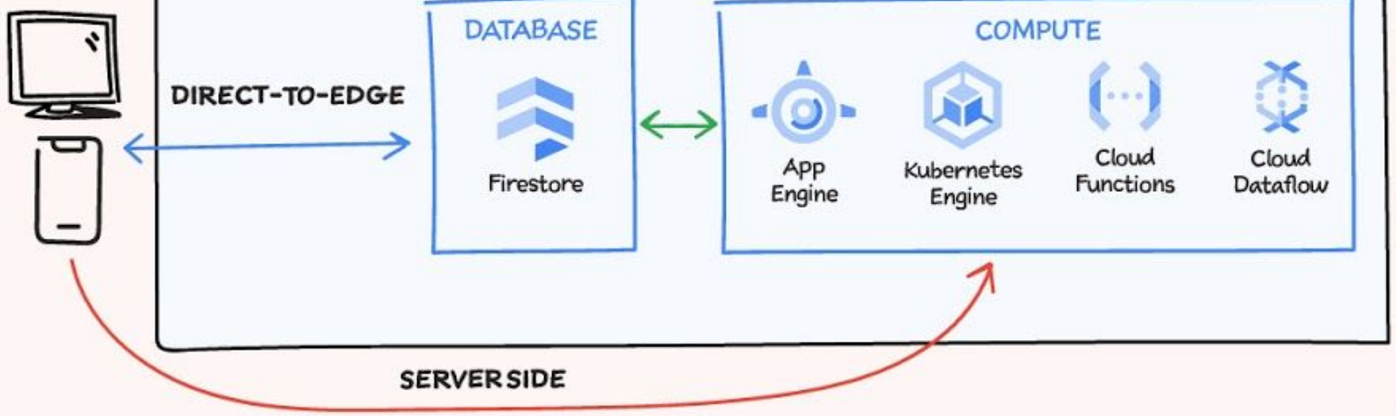
- Effortless syncing
- Offline mode — Perfect for mobile devices in case network is lost
- Built-in Security
- Global Strong Consistency
- Scalable
- SLA 99.999% SLA

→ BACKEND-AS-A-SERVICE

Allows apps to directly connect to the database

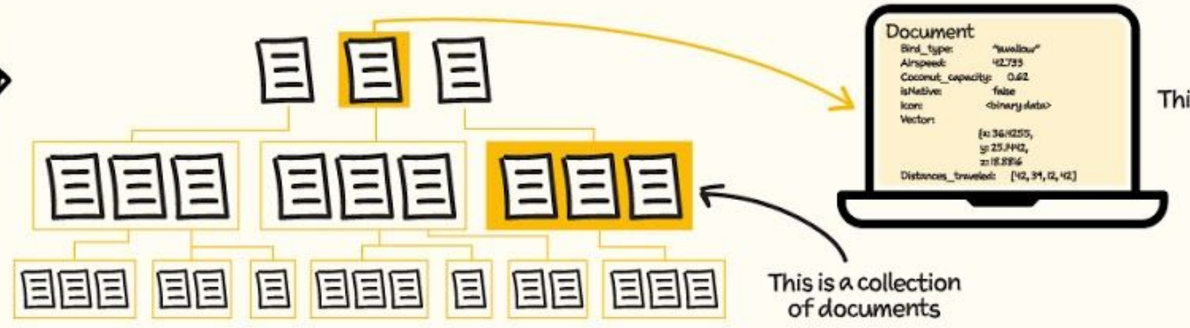

## How do you use it?

→ 2 MODES - DIRECT-TO-EDGE & SERVER-SIDE

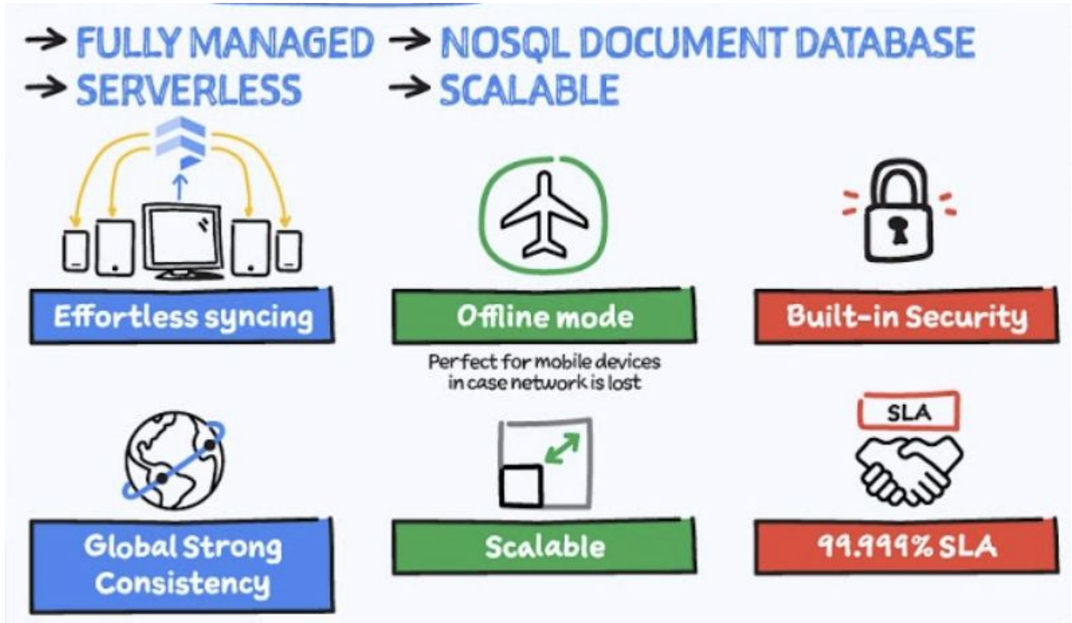


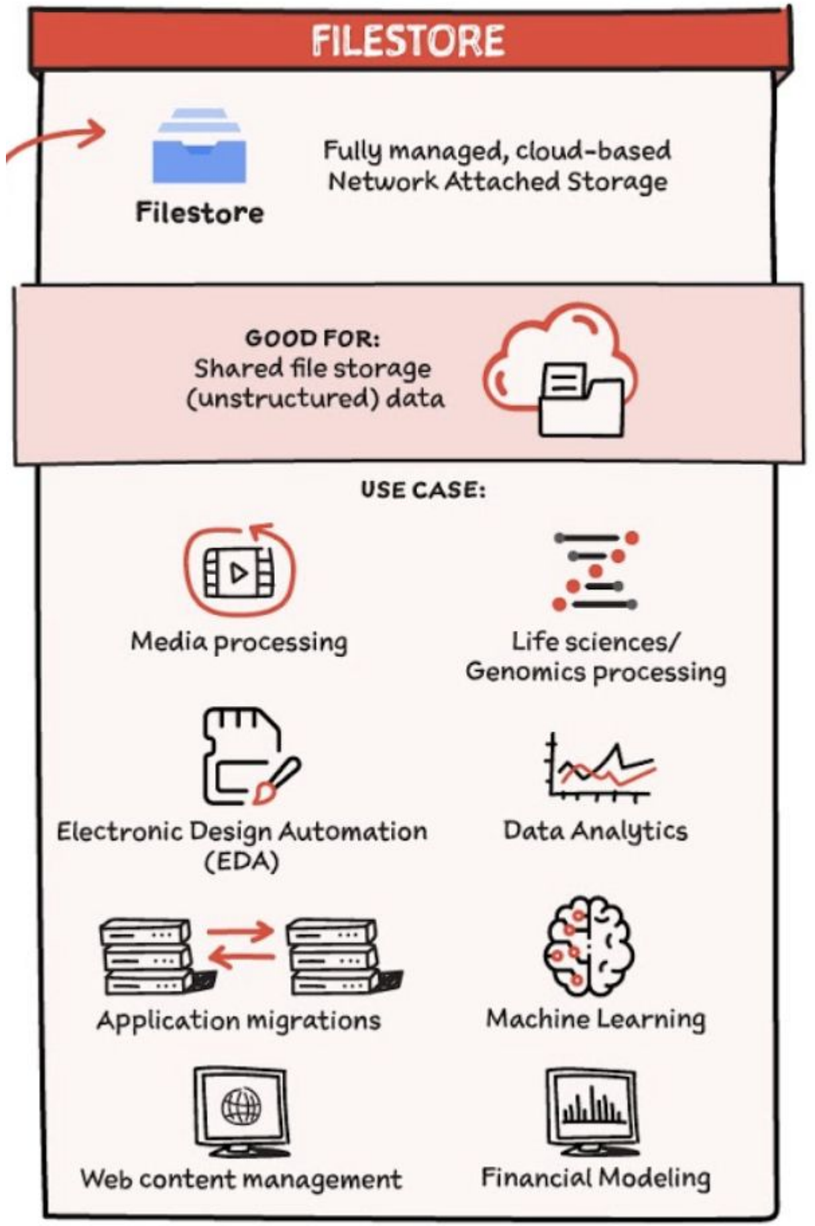

## How is data stored?

→ DOCUMENTS ARE STORED IN COLLECTIONS



# Firestore vs Filestore



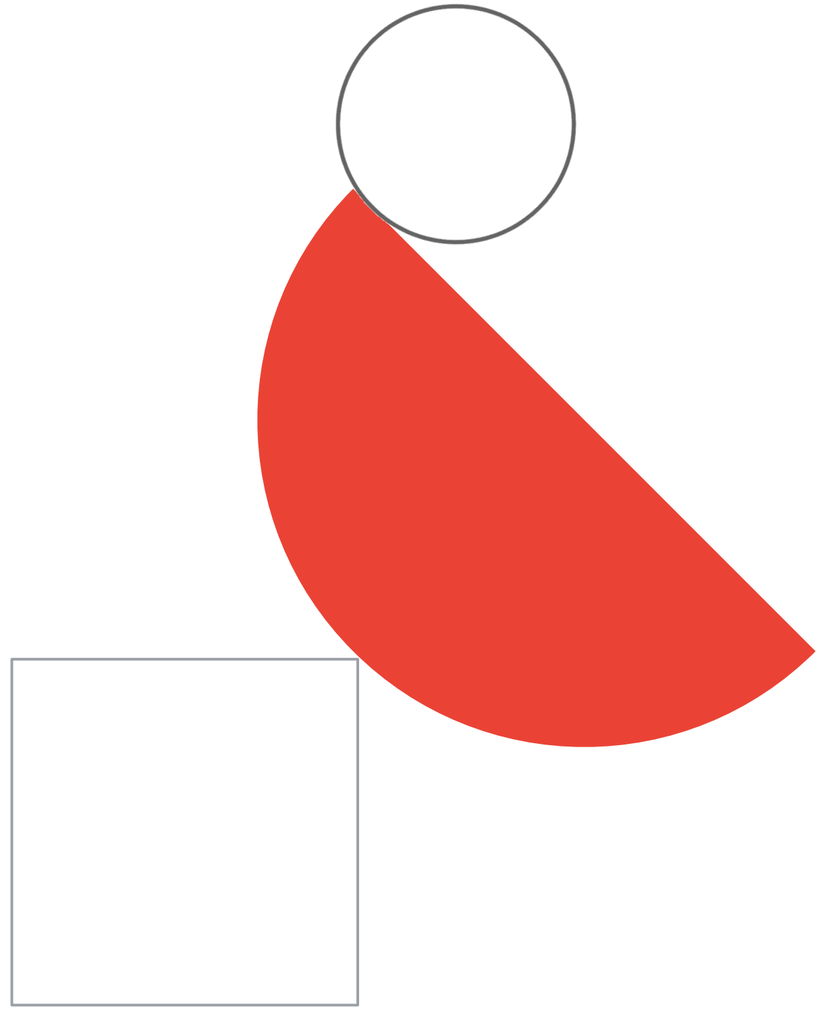




## … vs Firebase

***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

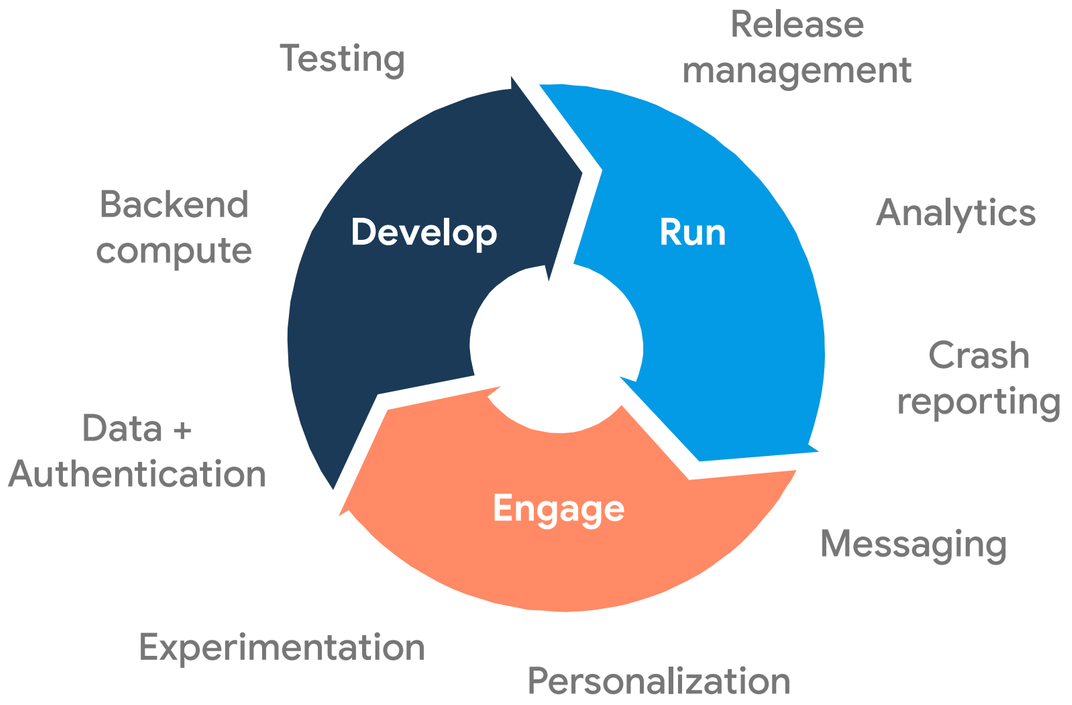
*** Platform, NOT a database ***



# Firebase is Google's complete app development platform

Complete = it provides different products to:

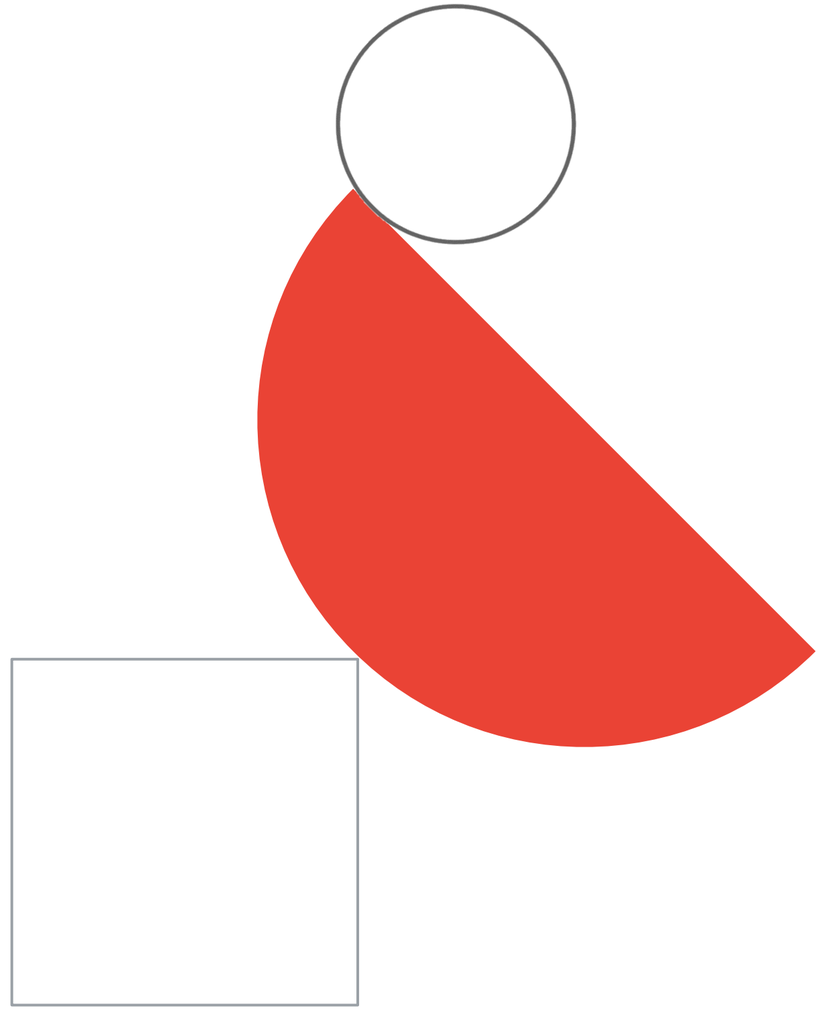
- Build apps
- Test apps
- Implement authentication (Firebase Authentication can be a part of PCA exam on very high-level!)
- Run apps
- Run analytics
- Personalize apps
- And more...




iOS Android Web C++ Unity

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

# Memorystore



# Memorystore #GCPSketchnote

@PVERGADIA THECLOUDGIRL.DEV

6.29.2021

## What is Memorystore?


- APPLICATION
  - Web browser
- MEMORYSTORE
  - Fully managed in-memory data store for Redis & Memcached
  - Scalable, more secure, & highly available
- DATABASES

| | Redis | Memcached |
|---|---|---|
| Storage | 100s of GB | TBs |
| Throughput (in QPS) | 100s of thousands | Millions |

In memory caching layer - high-speed storage layer that stores a subset of data & reduces latency


## What is your applications' availability need?

### MEMORYSTORE FOR REDIS

**BASIC TIER**

- Single Redis instance, ideal for caching use cases
- Instance health monitoring & automatic recovery from failures
- No SLA

**STANDARD TIER**

- Replicated Redis instance, increased availability
- One secondary replica deployed across zones, protection from zone failures
- Seamless scale up down
- 99.9% availability SLA

## FEATURES & CAPABILITIES

### SECURE BY DEFAULT


Data is protected from the internet using VPC networks, private IP & IAM integration

Instance Auth, Data encrypted in-transit

### SEAMLESS SCALE & HA

Standard high availability instances are replicated across zones

### DEEP INSIGHTS

Monitor instances using cloud operations

### BACKUP DATA

Easily backup instance data or import data into Memorystore from GCS buckets using RDB files

### NO CODE CHANGES

OSS compliance allows using Memorystore without any code changes

## CLOUD MEMORYSTORE USE CASES

### GAMING

- Session Store
- Distributed Lock

### SOCIAL MEDIA

- Content Caching
- Fast Data Ingestion
- Jobs and Queues
- Stream Processing

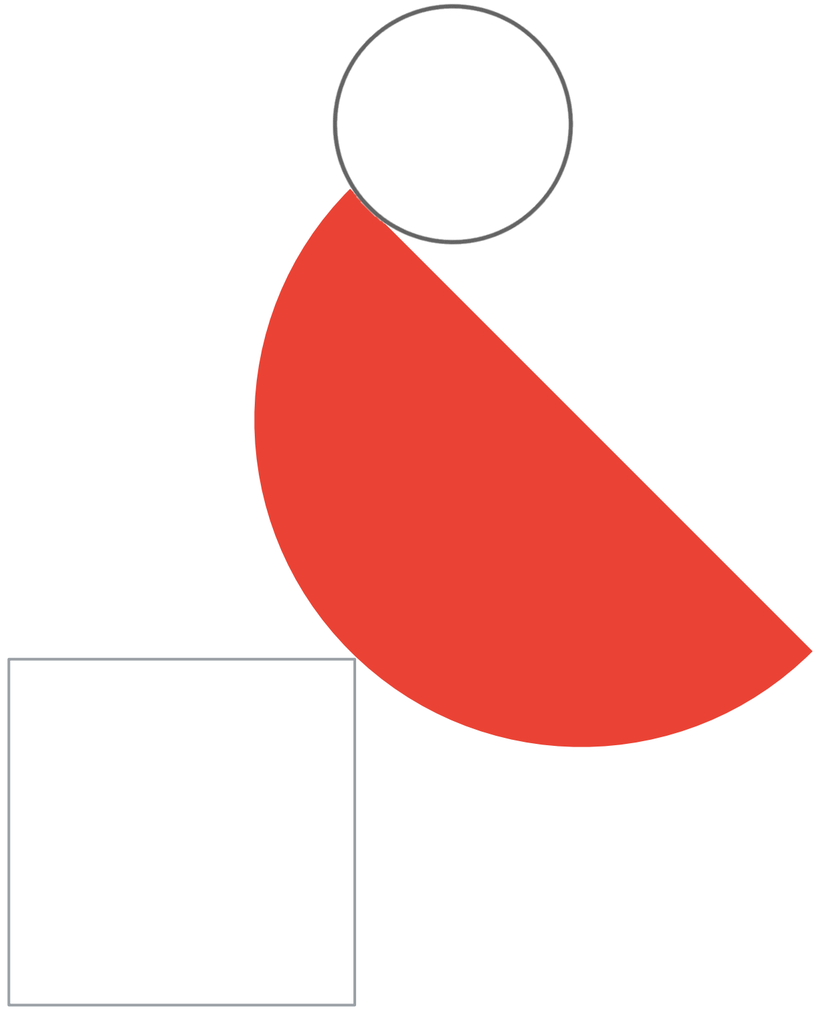
### IoT

- Real-time Analytics
- Counters
- Leaderboard
- Page Ranking

### ANALYTICS

- Recommendation Engine

# Spanner



# What workloads fit Cloud Spanner best?

01

**Sharded RDBMS**

Manually sharding is difficult. People do it to achieve scale. Cloud Spanner gives you relational data and scale.

02

**Scalable relational data**

Scalable relational database. Instead of moving to NoSQL, move from one relational database to a more scalable relational database.

03

**Manageability/HA**

Highly automated. Online Schema changes and patching. No planned downtime and comes with up to a 99.999% availability SLA.

04

**Multi-region**

Write once and automatically replicate your data to multiple regions.

Most customers use regional instances, but multi-region is there if you need it.

# When Cloud Spanner fits less well

**TIP**

It’s NOT a straightforward thing to migrate a different RDBMS to Cloud Spanner. Be familiar with challenges on high level.

1. Lift and shift
2. Lots of in-database business logic (triggers, stored procedures)
3. Compatibility needed
4. App is very sensitive to very low latency (micro/nano/low single digit ms)

   Lots of analytics / OLAP type of queries / workloads

Google Cloud

# Cloud Spanner

#GCPSketchnote

@PVERGADIA THECLOUDGIRL.DEV

5.7.2021

## What is Cloud Spanner?

- ✓ FULLY MANAGED
- ✓ HORIZONTALLY SCALABLE
- ✓ GLOBALLY CONSISTENT
- ✓ RELATIONAL DATABASE
- ✓ MULTI-VERSION DATABASE

**Relational Semantics**

Schemas, ACID transactions, SQL

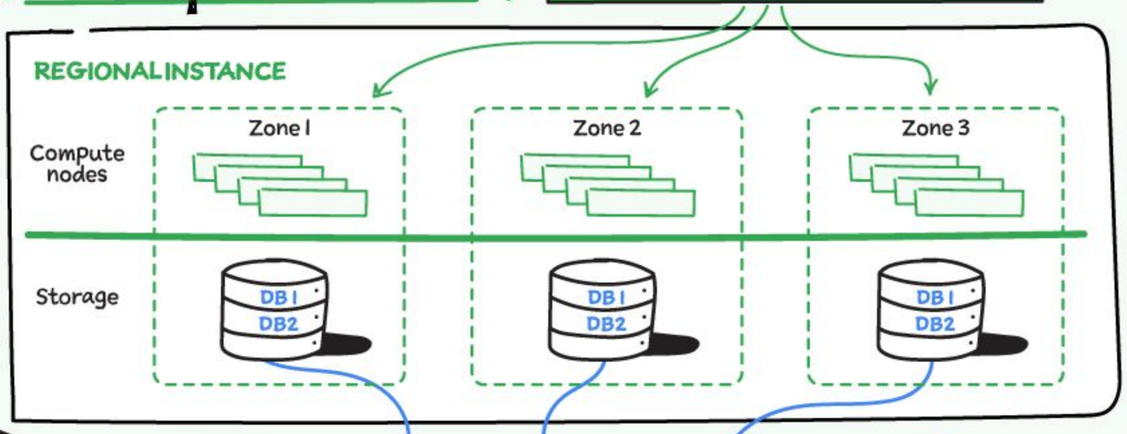
**Horizontal Scale**

99.999% SLA, fully managed, and scalable

Relational | Non-Relational

## How does Cloud Spanner work?

This Spanner instance contains 4-nodes




This Spanner instance is hosting 2 databases across 3 zones

## How does Spanner provide «global consistency?»

**SPANNER GLOBAL CONSISTENCY**

**TrueTime**

Synchronizes clocks in all machines across datacenters

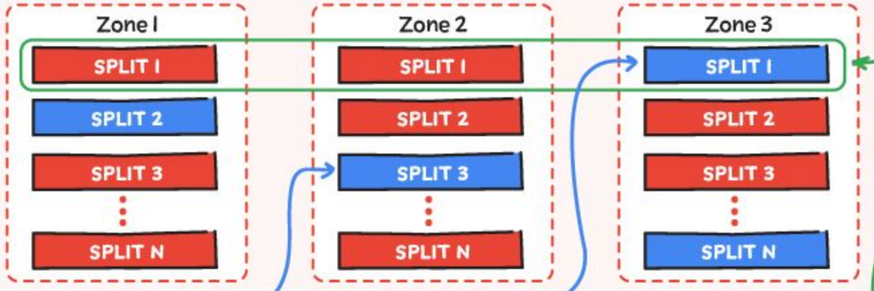
+

**Google's Global Network**

Fast & redundant

## How does Spanner provide high availability* & scalability*?

Zero downtime for planned maintenance or schema changes




(1) **WHAT IS A SPLIT?**

Each table in the database is broken down into several splits using ranges of the primary key

| SPLIT ID | 0 | 1 | ... | n |
|---|---|---|---|---|
| KEY RANGE | [-∞,3] | [4,224] | ... | [2457,∞] |

(2) Splits are re-balanced dynamically based on amount of data + load

(3) Paxos group for split 1

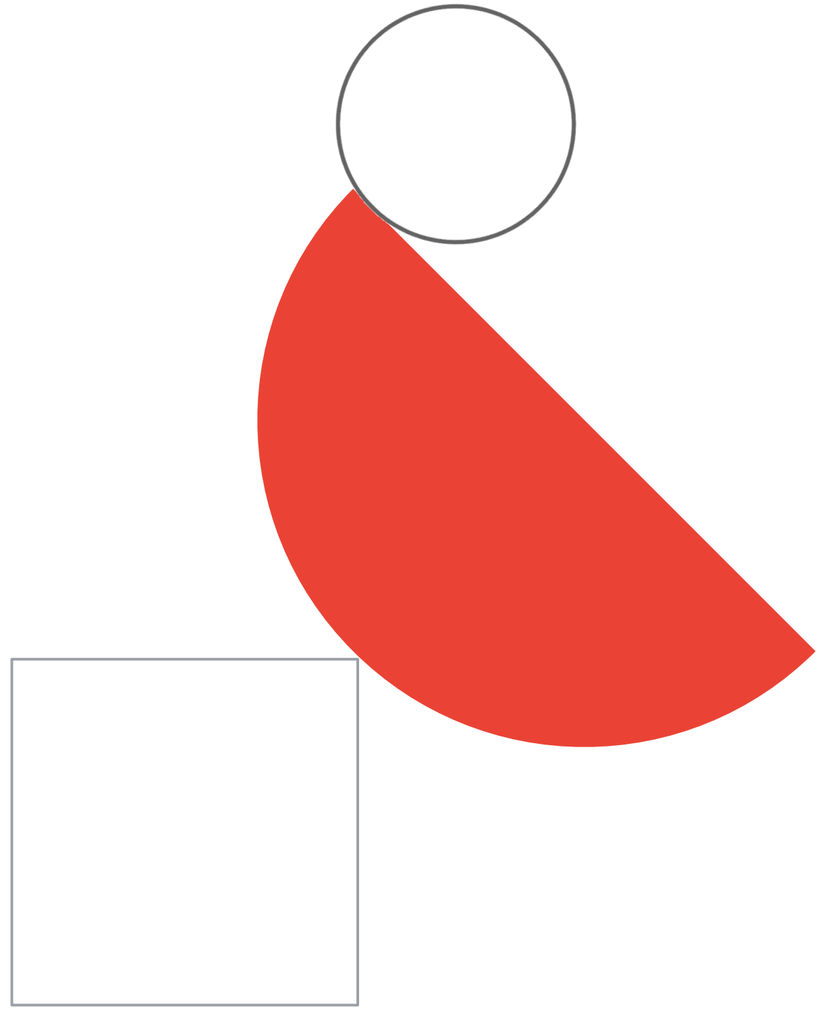
(4) Replicas participate in voting for writes to achieve a majority quorum (2 out of 3) and serve reads

(5) Different splits can have leaders in different zones

Leader manages writes for that split

All replicas serve reads

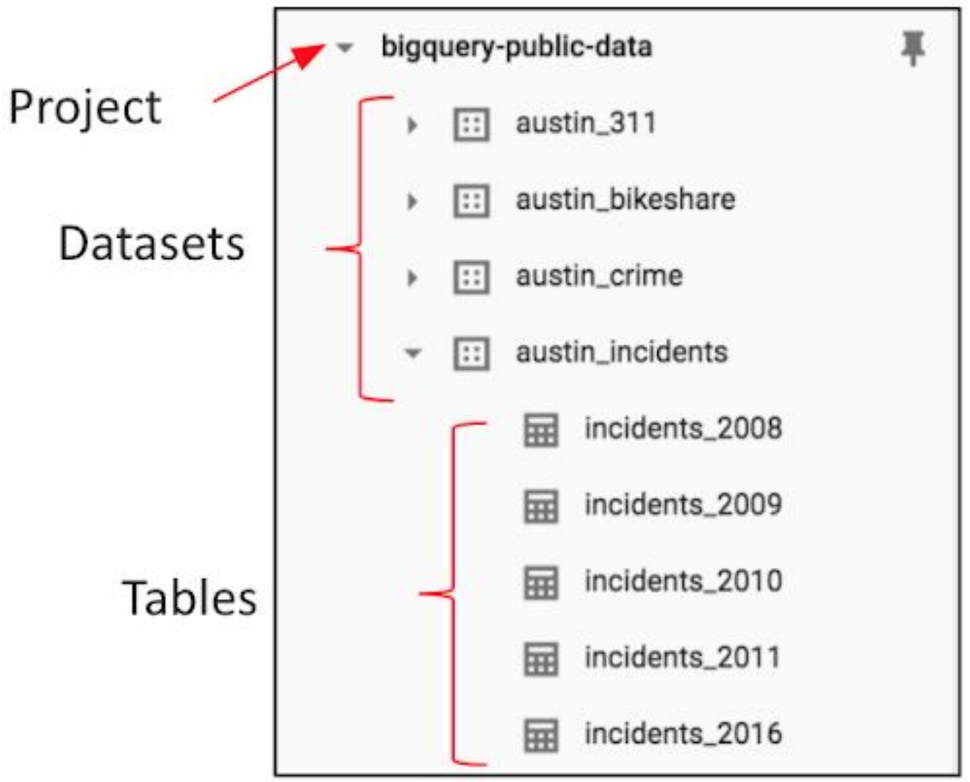
# Bigquery



# BigQuery hierarchy

## Project -> Dataset -> Tables (-> Partitions)

- For each query, BigQuery executes a full-column scan.
- BigQuery performance and query costs are based on the amount of data scanned.
- You can set the geographic location of a Dataset at creation time only.
- All tables that are referenced in a query must be stored in datasets in the same location.
- When you copy a table (bq cp), the datasets that contain the source table and destination table must reside in the same location.
  - You can copy a dataset (NOT with bq cp, but with BigQuery Data Transfer Service) within a region or from one region to another
- Dataset names are case-sensitive



# BigQuery: Controlling access to datasets

## Common BigQuery predefined roles

***Exam Tips:*** *It's a common practice to have a Dataset in one project and perform queries from another one (split billing!).*

| | |
|---|---|
| Admin | Full Access to all datasets |
| Data Editor | Access to edit all contents of the datasets |
| Data Owner | Full access to datasets and all of their contents |
| Data Viewer | Access to view datasets and all of their contents |
| Job User | Access to run jobs |
| Metadata Viewer | Access to view table and dataset metadata |
| User | Access to run queries and create datasets |
| Read Sessions User | Access to create and use read sessions |

| Capability | dataViewer | dataEditor | dataOwner | user | jobUser | admin |
|---|---|---|---|---|---|---|
| List/get projects | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| List tables | ✓ | ✓ | ✓ | ✓ | ⊘ | ✓ |
| Get table data/metadata | ✓ | ✓ | ✓ | ⊘ | ⊘ | ✓ |
| Create tables | ⊘ | ✓ | ✓ | ⊘ | ⊘ | ✓ |
| Modify/delete tables | ⊘ | ✓ | ✓ | ⊘ | ⊘ | ✓ |
| List/get datasets | ✓ | ✓ | ✓ | ✓ | ⊘ | ✓ |
| Create new datasets | ⊘ | ✓ | ✓ | ✓ | ⊘ | ✓ |
| Modify/delete datasets | ⊘ | ⊘ | ✓ | Self-created datasets | ⊘ | ✓ |
| Create jobs/queries | ⊘ | ⊘ | ⊘ | ✓ | ✓ | ✓ |
| Cancel jobs | ⊘ | ⊘ | ⊘ | Self-created jobs | Self-created jobs | Any jobs |
| Get/list saved queries | ⊘ | ⊘ | ⊘ | ✓ | ⊘ | ✓ |
| Create/update/delete saved queries | ⊘ | ⊘ | ⊘ | ⊘ | ⊘ | ✓ |
| Get transfers | ⊘ | ⊘ | ⊘ | ✓ | ⊘ | ✓ |
| Create/update/delete transfers | ⊘ | ⊘ | ⊘ | ⊘ | ⊘ | ✓ |

# 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

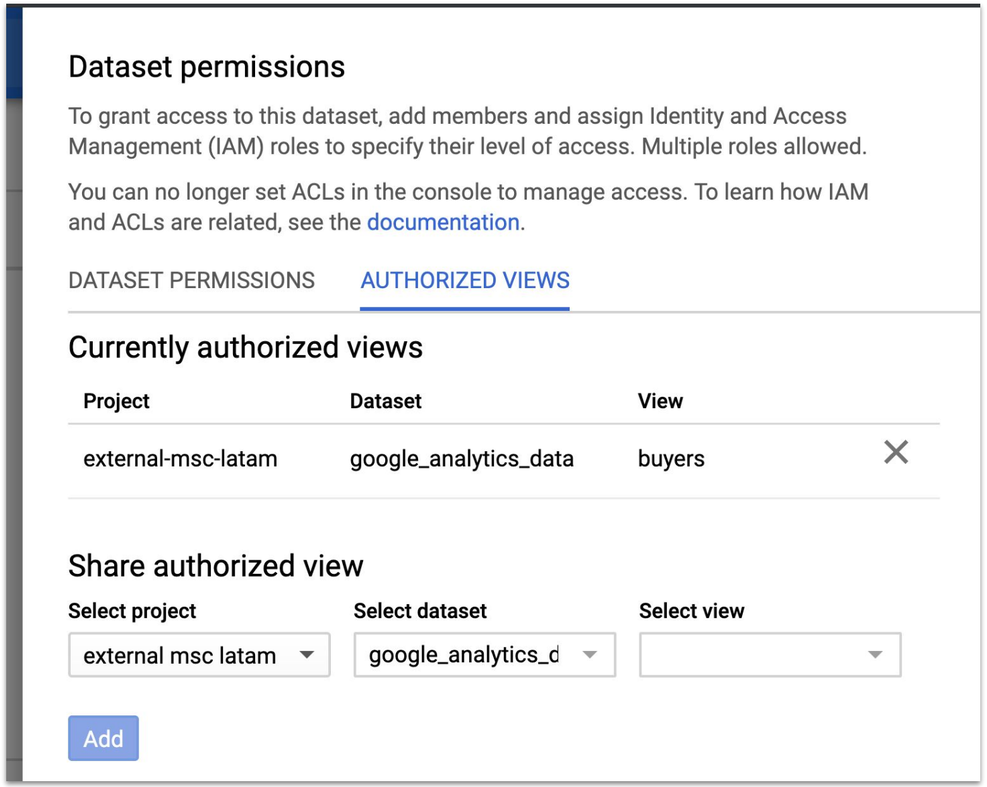
## Authorized Views

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.

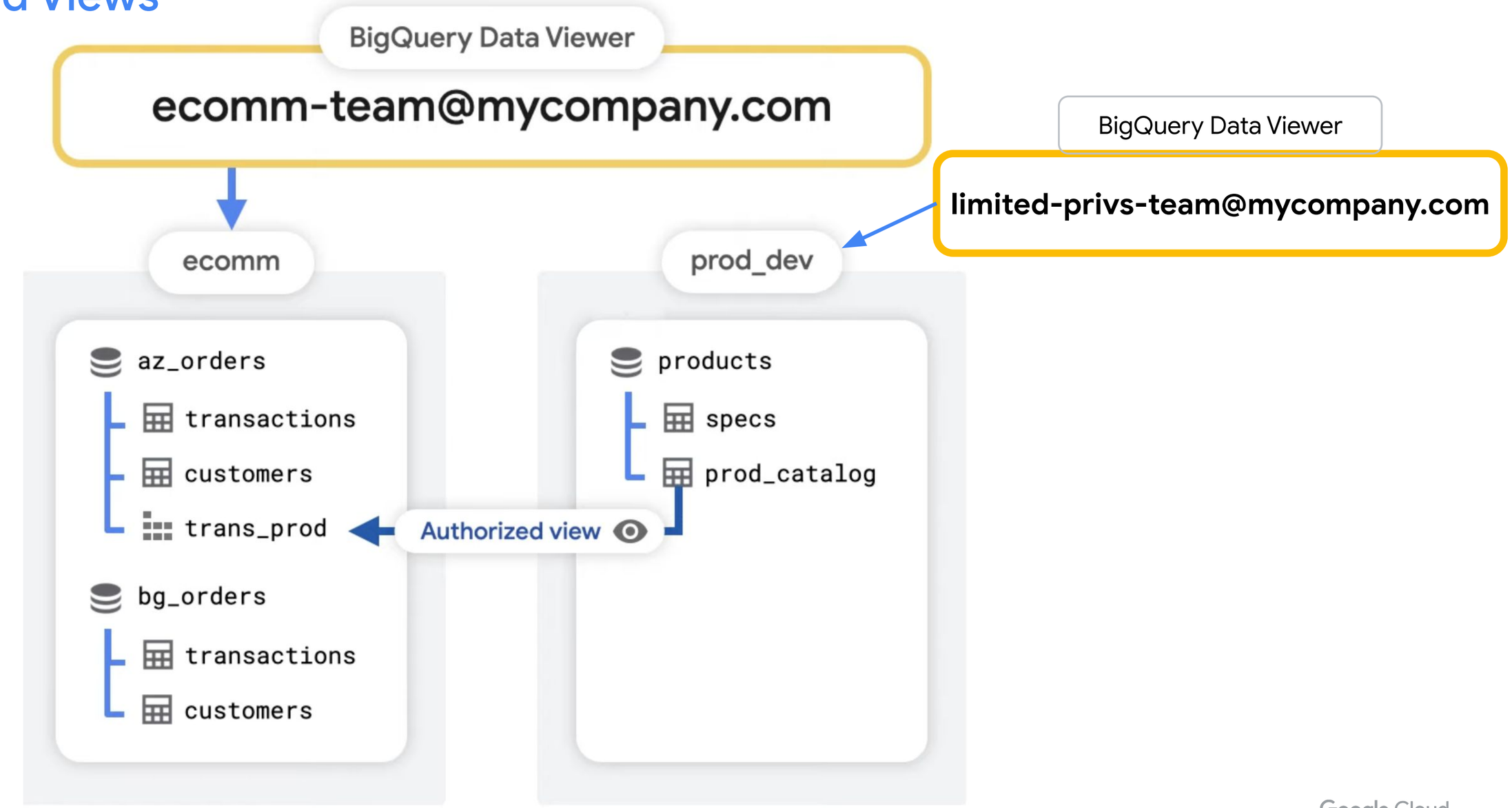
3. **Authorized Views:** An authorized view allows you to share query results with particular users and groups without giving them access to the underlying tables.

***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: Controlling access to datasets

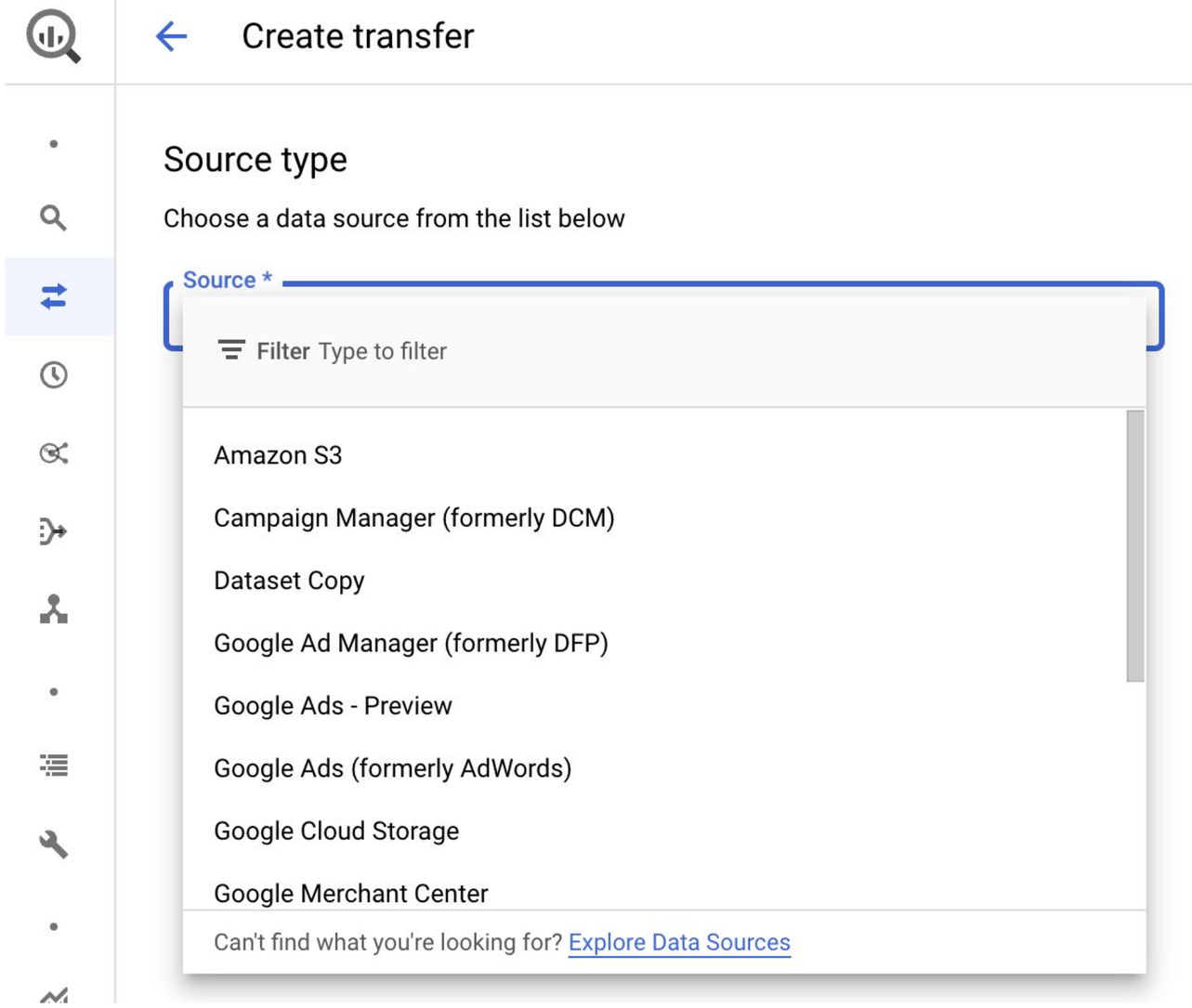
## Authorized Views



# BigQuery - Data Transfer Service

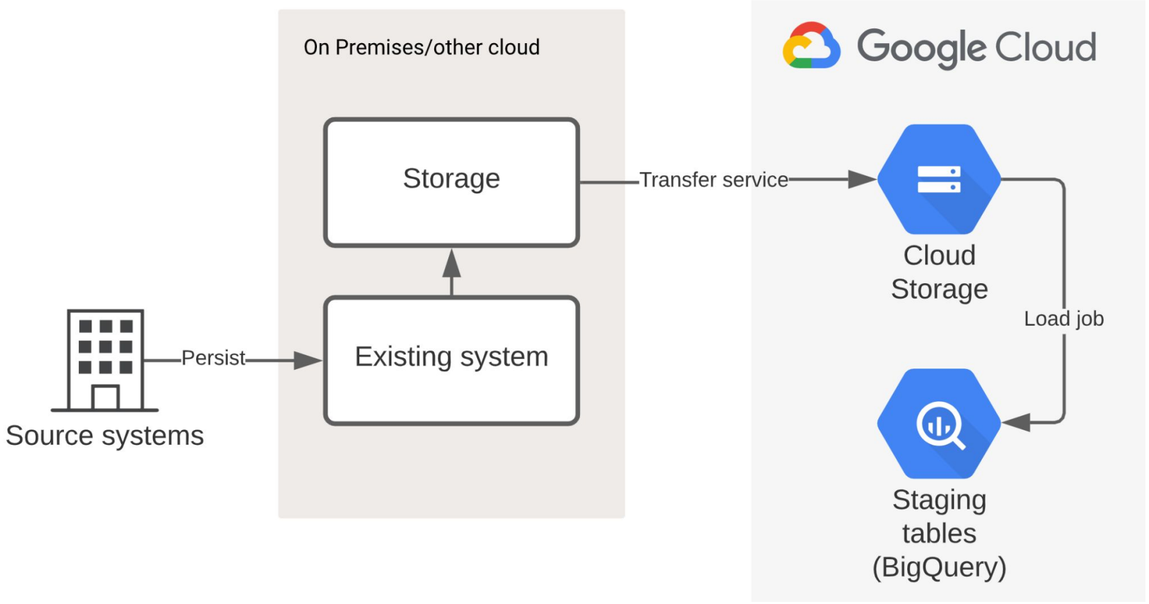
## Mostly useful for regular data transfers to BigQuery

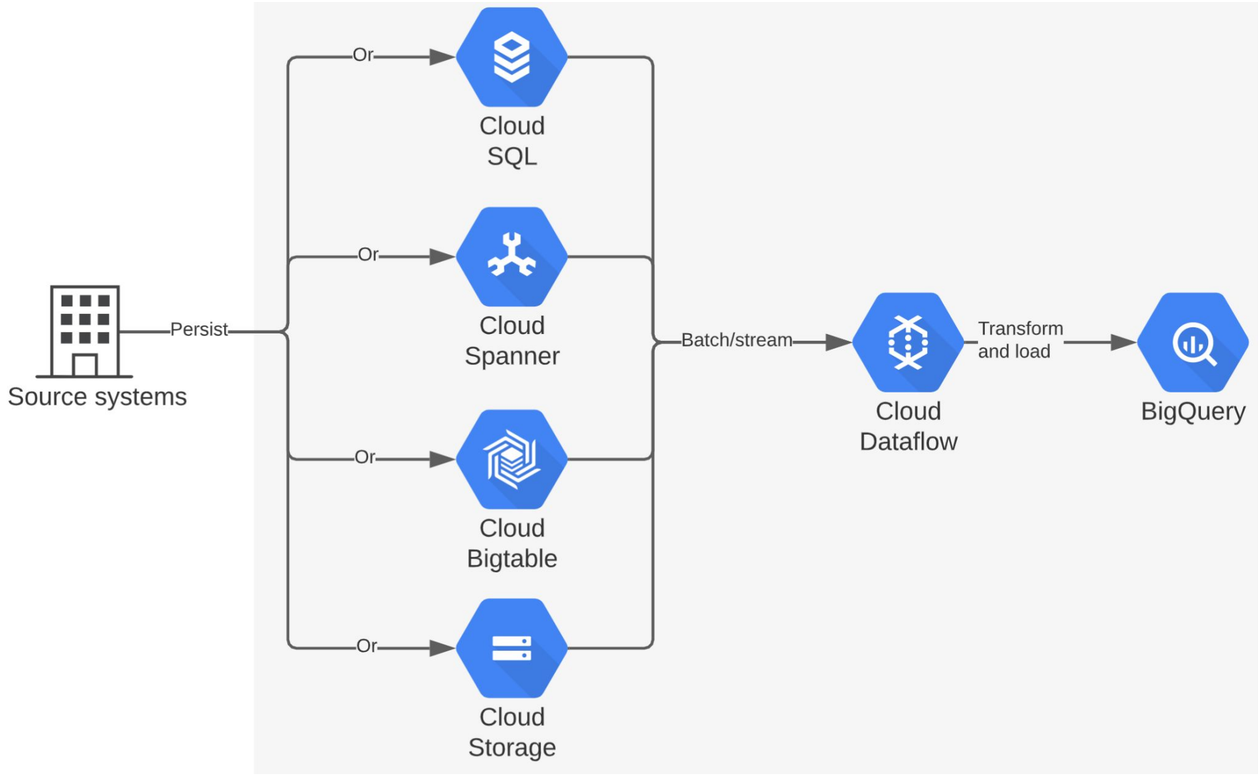
- BigQuery Data Transfer Service automates data movement from various sources into BigQuery on a scheduled, managed basis.
- You can initiate data backfills to recover from any outages or gaps.



# BigQuery - Batch vs Streaming inserts

## Most common architectures



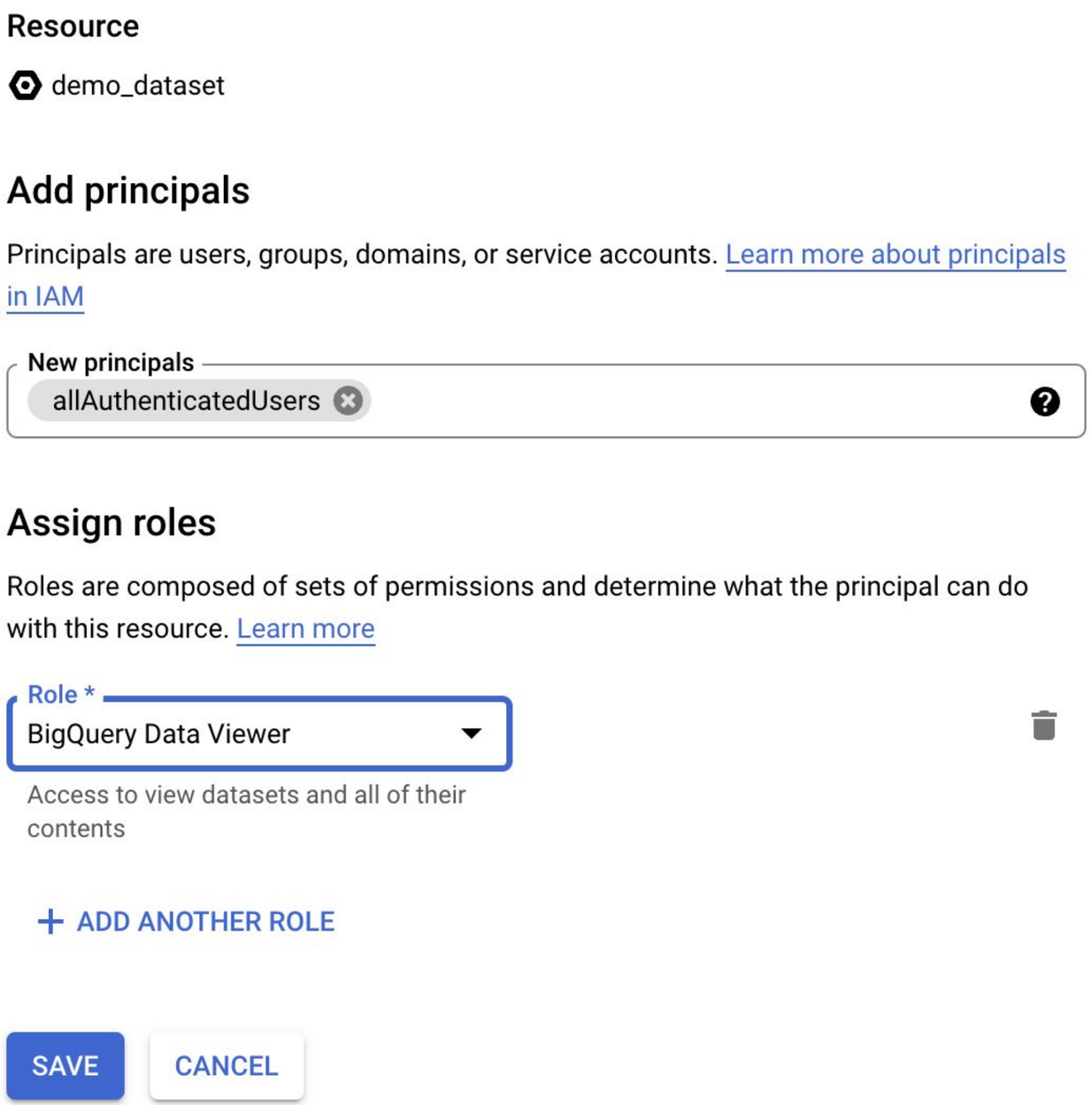




***Exam Tip:*** *There is additional cost for streaming (both inserts and reads) in BigQuery.*

# BigQuery: Sharing Datasets with others

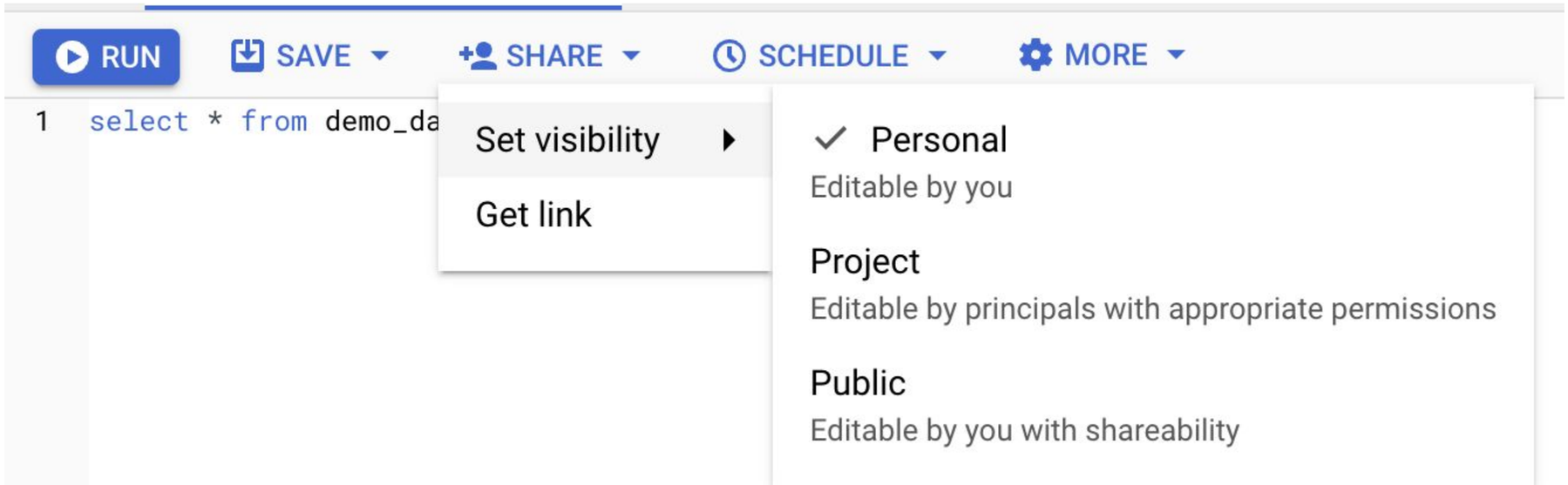
## AllAuthenticatedUsers

The special setting allAuthenticatedUsers makes a dataset public. Authenticated users must use BigQuery within their own project and have access to run BigQuery jobs so that they can query the Public Dataset. The billing for the query goes to their project, even though the query is using public or shared data. In summary, the cost of a query is always assigned to the active project from where the query is executed.



# BigQuery: Sharing Queries with others

## Mostly for collaboration

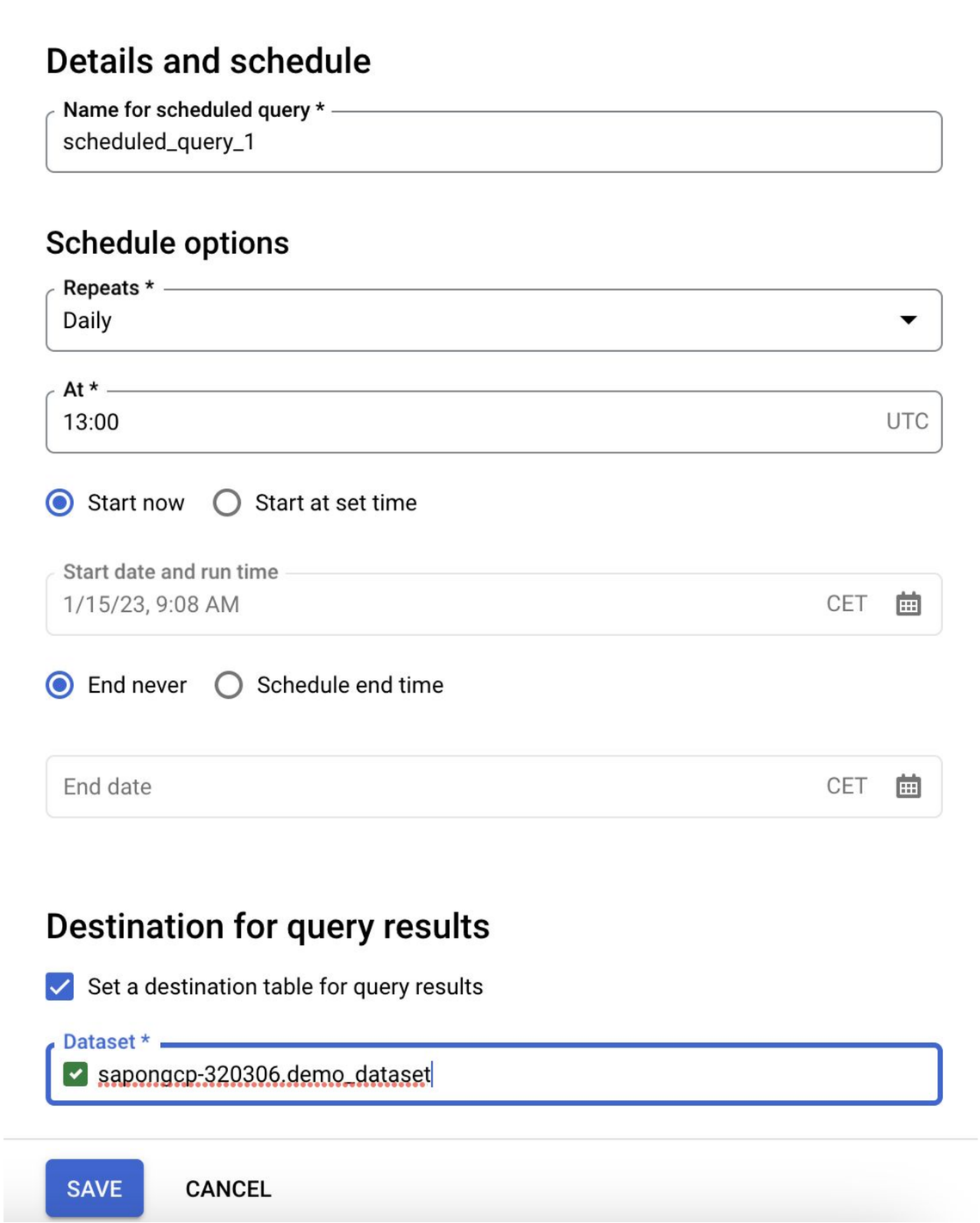



- Query needs to be saved first, before it's shared;
- Can share incomplete / invalid queries -> collaboration;
- Project-level saved queries are visible to principals with the required permissions;
- Public saved queries are visible to anyone with a link to the query;

# BigQuery: Scheduling queries

## Mostly useful for regular execution

- Scheduled queries use features of BigQuery Data Transfer Service.
- If the destination table for your results doesn't exist when you set up the scheduled query, BigQuery attempts to create the table for you.
- You can set up a scheduled query to authenticate as a service account.



# BigQuery: Query results caching

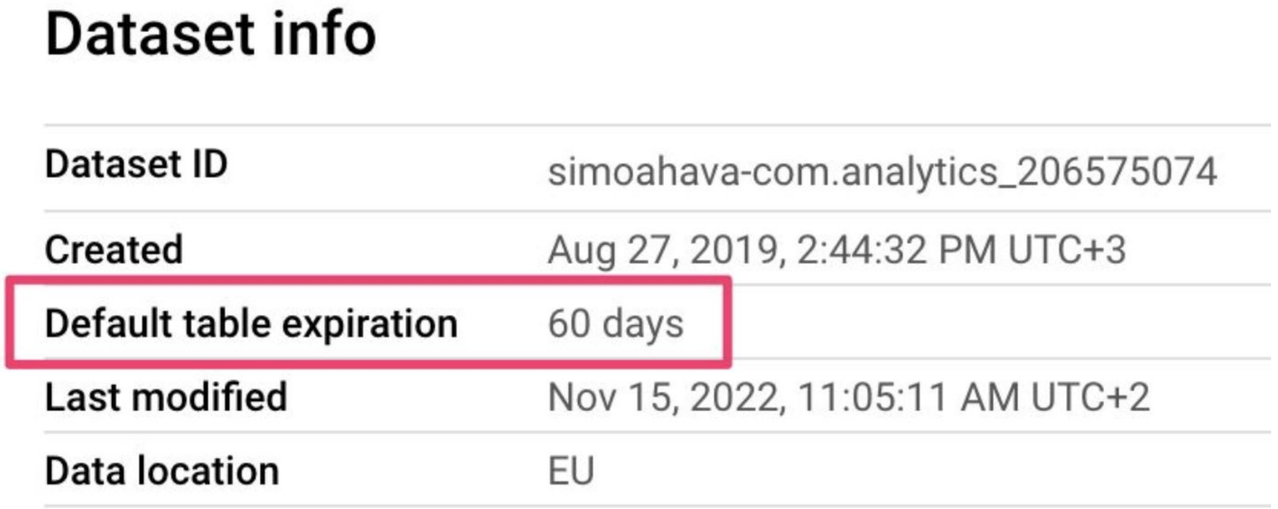
## Limit Access by Data Lifecycle Stages

- Query results are cached to improve performance and reduce costs for repeated queries
- Cache is per user
- Still subject to quota policies
- Cache results have a size limit of 128 MB compressed
- No charge for queries that use cached results
- Results are cached for approximately 24 hours
- Lifetime extended when a query returns a cached result
- Use of cached results can be turned off (useful for benchmarking)

# 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

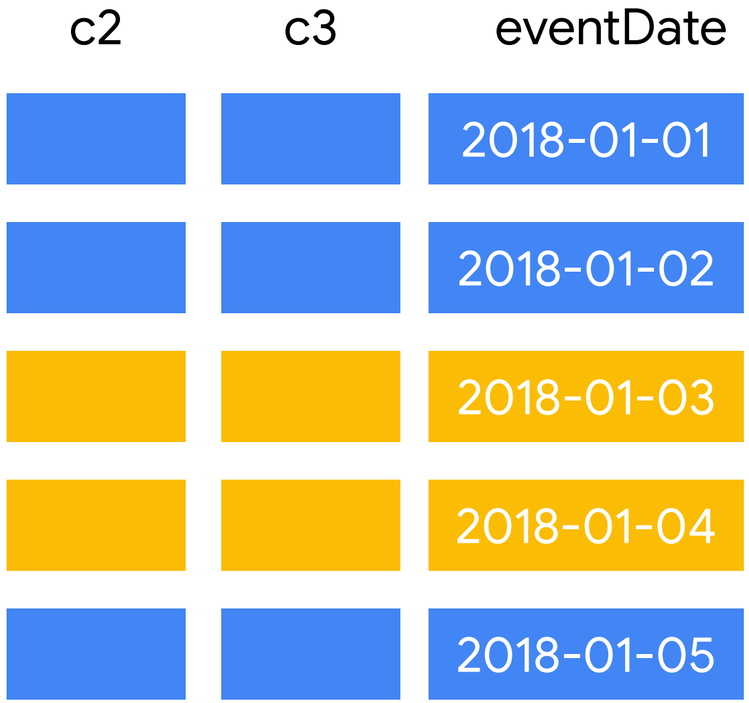
# BigQuery: Table Partitioning

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.

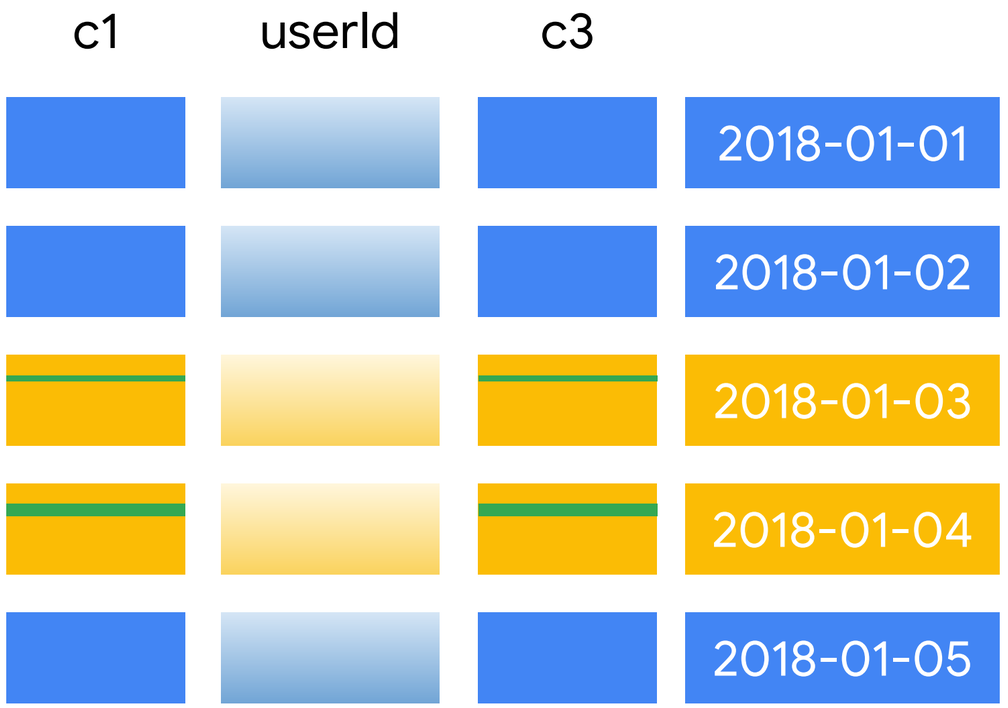
You can partition BigQuery tables by:

- 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 partitioning AND clustering

## Both partitioning and clustering can improve performance and reduce query cost

**Orders table**

**Not Clustered; Not partitioned**

| Order_Date | Country | Status |
|---|---|---|
| 2022-08-02 | US | Shipped |
| 2022-08-04 | JP | Shipped |
| 2022-08-05 | UK | Canceled |
| 2022-08-06 | KE | Shipped |
| 2022-08-02 | KE | Canceled |
| 2022-08-05 | US | Processing |
| 2022-08-04 | JP | Processing |
| 2022-08-04 | KE | Shipped |
| 2022-08-06 | UK | Canceled |
| 2022-08-02 | UK | Processing |
| 2022-08-05 | JP | Canceled |
| 2022-08-06 | UK | Processing |
| 2022-08-05 | US | Shipped |
| 2022-08-06 | JP | Processing |
| 2022-08-02 | KE | Shipped |
| 2022-08-04 | US | Shipped |

**Orders table**

**Clustered by Country; Not partitioned**

| Order_Date | Country | Status |
|---|---|---|
| 2022-08-04 | JP | Shipped |
| 2022-08-04 | JP | Processing |
| 2022-08-05 | JP | Canceled |
| 2022-08-06 | JP | Processing |
| 2022-08-06 | KE | Shipped |
| 2022-08-02 | KE | Canceled |
| 2022-08-04 | KE | Shipped |
| 2022-08-02 | KE | Shipped |
| 2022-08-05 | UK | Processing |
| 2022-08-06 | UK | Canceled |
| 2022-08-02 | UK | Canceled |
| 2022-08-06 | UK | Processing |
| 2022-08-02 | US | Shipped |
| 2022-08-05 | US | Processing |
| 2022-08-05 | US | Shipped |
| 2022-08-04 | US | Shipped |

**Orders table**

**Clustered by Country; Partitioned by Order_Date (Daily)**

| | Order_Date | Country | Status |
|---|---|---|---|
| **Partition:** 2022-08-02 | 2022-08-02 | KE | Shipped |
| | 2022-08-02 | KE | Canceled |
| **Clusters:** Country | 2022-08-02 | UK | Processing |
| | 2022-08-02 | US | Shipped |

| | Order_Date | Country | Status |
|---|---|---|---|
| **Partition:** 2022-08-04 | 2022-08-04 | JP | Shipped |
| | 2022-08-04 | JP | Processing |
| **Cluster:** Country | 2022-08-04 | KE | Shipped |
| | 2022-08-04 | US | Shipped |

| | Order_Date | Country | Status |
|---|---|---|---|
| **Partition:** 2022-08-05 | 2022-08-05 | JP | Canceled |
| | 2022-08-05 | UK | Canceled |
| **Cluster:** Country | 2022-08-05 | US | Shipped |
| | 2022-08-05 | US | Processing |

| | Order_Date | Country | Status |
|---|---|---|---|
| **Partition:** 2022-08-06 | 2022-08-06 | JP | Processing |
| | 2022-08-06 | KE | Shipped |
| **Cluster:** Country | 2022-08-06 | UK | Canceled |
| | 2022-08-06 | UK | Processing |

***Exam Tip:*** *You can combine partitioning with clustering. Data is first partitioned and then data in each partition is clustered by the clustering columns.*

# BigQuery: Storage Pricing

Storage pricing is the cost to store data that you load into BigQuery. You pay for *active storage* and *long-term* storage.

- **Active storage** includes any table or table partition that has been modified in the last 90 days.
- **Long-term storage** includes any table or table partition that has not been modified for 90 consecutive days. The **price of storage for that table automatically drops by approximately 50%**. There is no difference in performance, durability, or availability between active and long-term storage.

The first 10 GB of storage per month is free.

US (multi-region)

| Operation | Pricing | Details |
|---|---|---|
| Active storage | $0.02 per GB | The first 10 GB is free each month. |
| Long-term storage | $0.01 per GB | The first 10 GB is free each month. |

# BigQuery

#GCPSketchnote

@PVERGADIA THECLOUDGIRL.DEV

8.5.2020

## What is BigQuery?

ENTERPRISE DATA WAREHOUSE

INGEST → STORE → ANALYZE → VISUALIZE

BATCH DATA

STREAMING DATA

Google will allocate as many servers as needed to run your query

99.99% SLA

ON DEMAND INSIGHTS

REAL-TIME INSIGHTS

## Unique BigQuery FEATURES

**BQ OMNI** (Private Alpha)
- Enables hybrid & multi-cloud queries

Data Storage

Query Engine

Anthos Clusters

**BQ ML**
- Create model
- Predict

**BI ENGINE**

BigQuery BI Engine

Data Studio

Sheets

Partner BI Tools

**DATA QnA** (Private Alpha)
- Get answers through NLP

Business User

Chat

Data QnA

Get answers through NLP

GEOSPACIAL

CONNECTED SHEETS

## BigQuery PRICING

FREE!

Free storage for all public datasets

Query upto 1TB/month for free

Batch Load is free

Streaming Inserts

Export is free

$ =

Storage

- ACTIVE <90 days
- LONG TERM >90 days

Query

- ON DEMAND $ per query
- FLAT RATE $ for dedicated resources

## SECURITY for BigQuery

0 Avoid risk of data loss

1 Identity and Access Management

2 VPC controls

3 Customer managed encryption key (CMEK)

## How it WORKS!

| GCS, HTTPs Post | Web, IoT, Mobile | ETL, ELT data | GCS, s3, Teradata, ... |
|---|---|---|---|
| BATCH INGESTION | STREAMING INGESTION | QUERY MATERIALIZATION | DATA TRANSFER SERVICE (DTS) |

INGEST

AUTOMATIC BACKUP & RESTORE

PUBLIC DATA SETS

ODBC and JDBC drivers (at no cost)

SQL ANSI:2011 compliant

COMPUTE

QUERY

Visualize

STORAGE

Looker

Data Studio

Connected Sheets

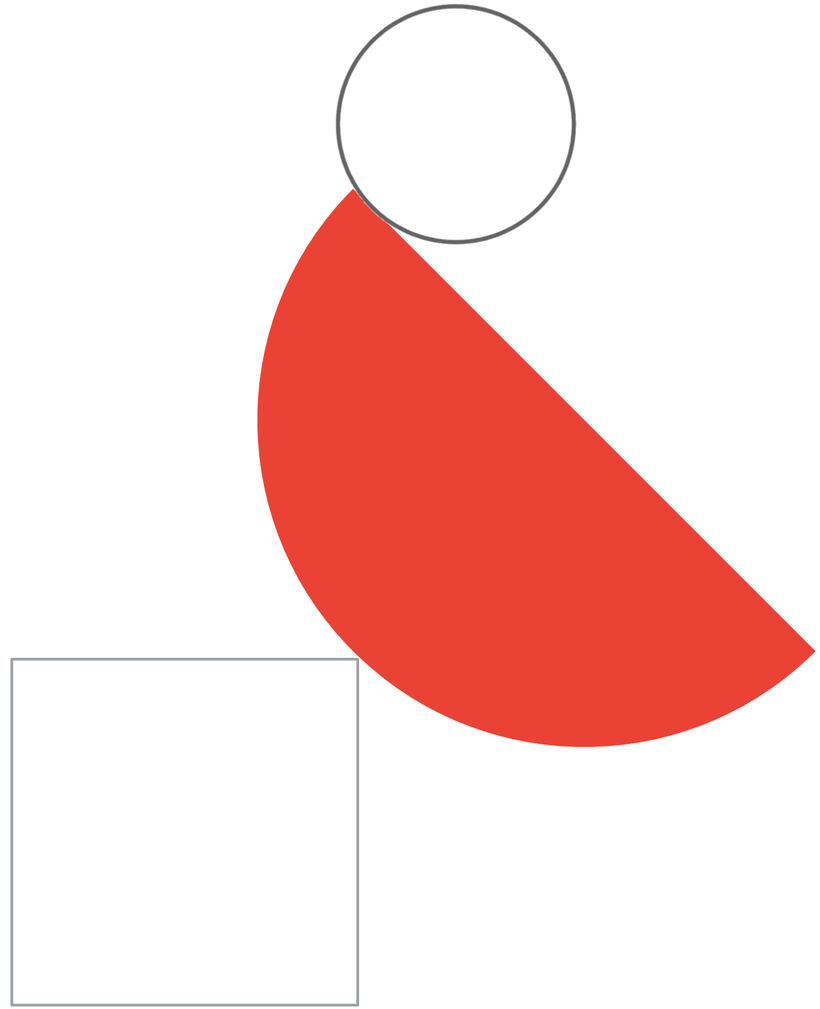
Partner BI Tools

## BigQuery Use case example

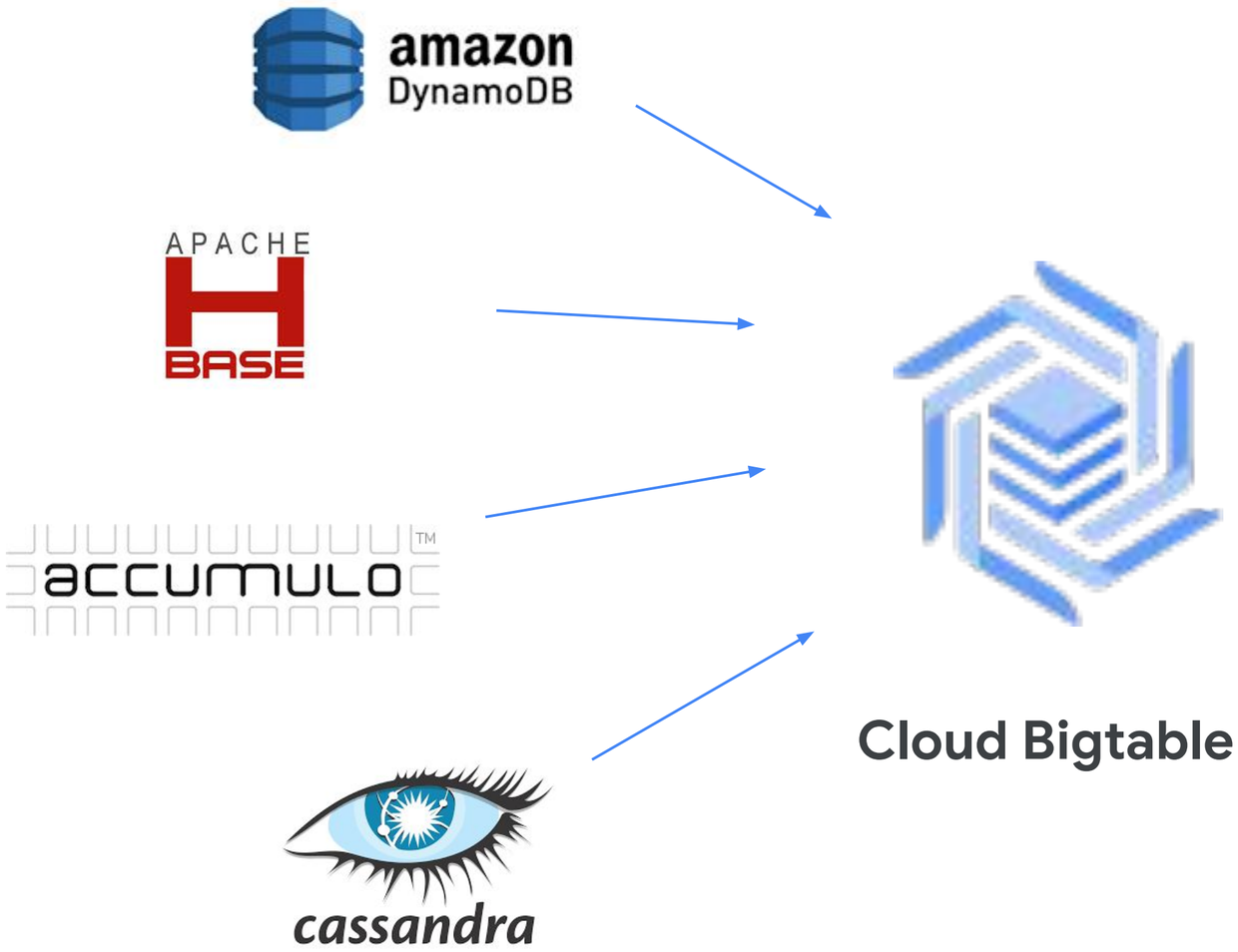
Data Analytics

| Trigger | Ingest | Enrich | Analyze | Visualize |
|---|---|---|---|---|
| Google Cloud Storage: Mobile, Web, Data Store, IoT | Batch Load, Pub/Sub, Streaming Inserts | Dataflow | BigQuery, AI Platform, Bigtable | Studio, Third-party BI, Google Sheets |

# Bigtable



# Bigtable is a common migration target for key-value, wide-column and time-series databases




- **Petabyte-scale**
- **fully managed** NoSQL database service for use cases where **low latency** random data access, **scalability** and **reliability** are critical.
- **scales seamlessly**
- **integrates** with the Apache® ecosystem and supports the HBase™ API.

# 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.*

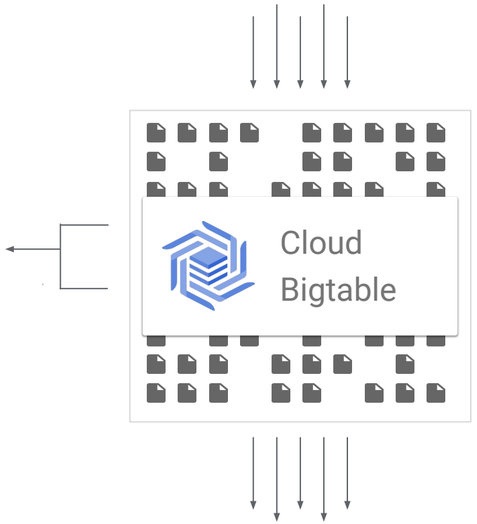
# Bigtable for analytics... ?

## 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

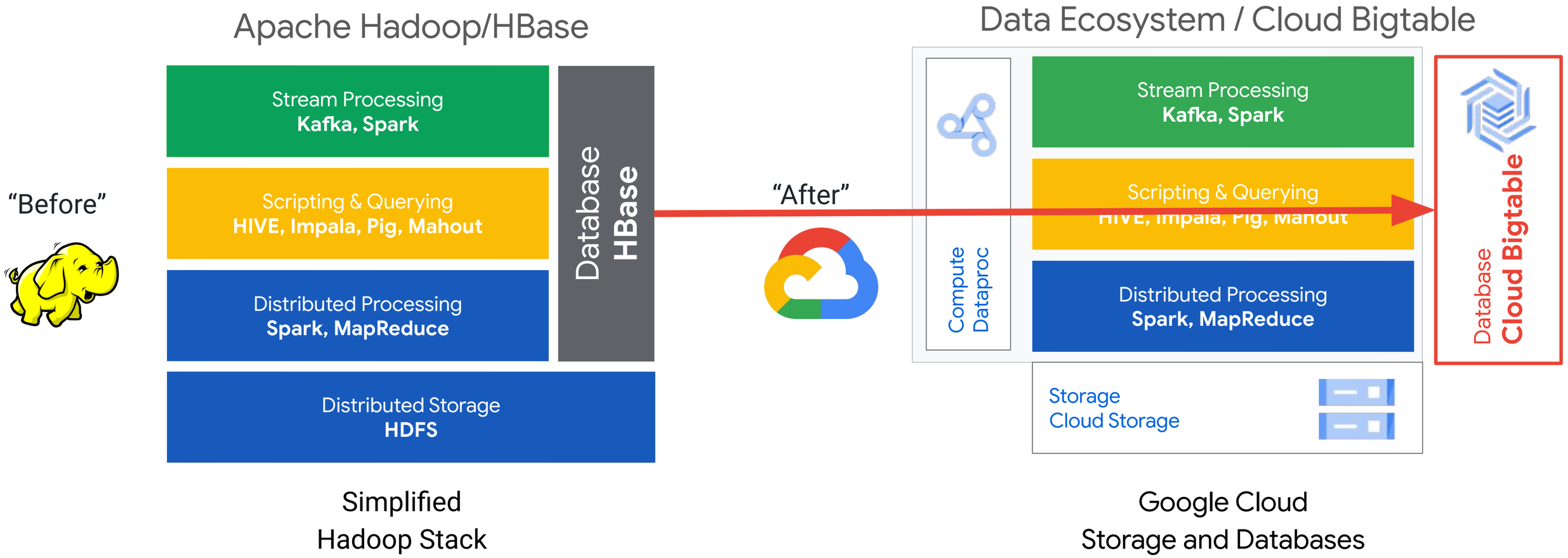
Organizational insights

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

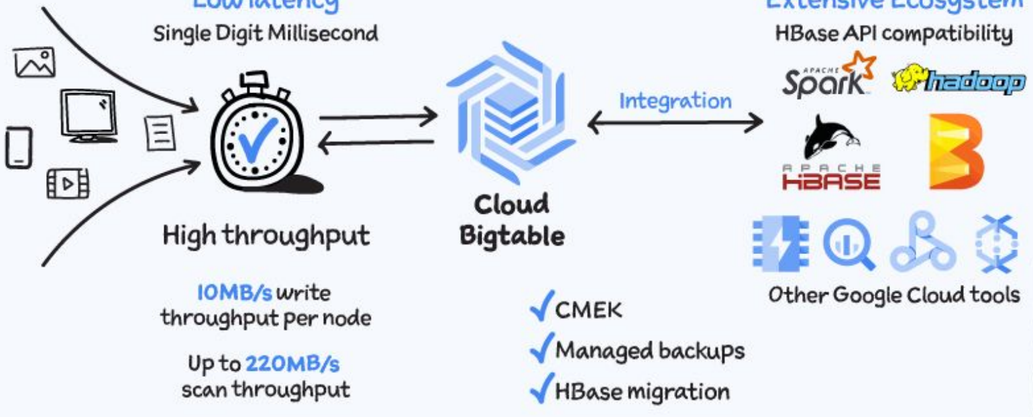
# Cloud Bigtable

#GCPSketchnotes

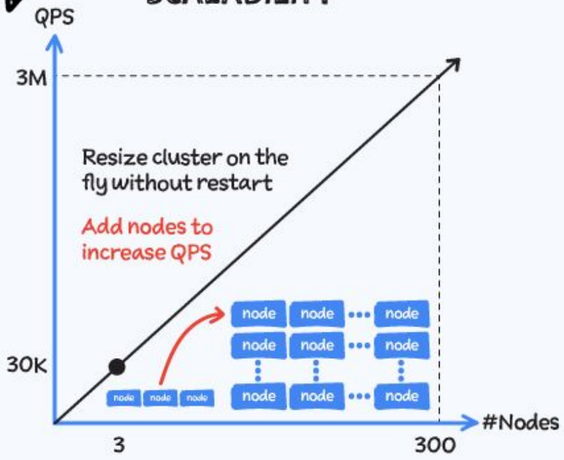
@PVERGADIA
THECLOUDGIRL.DEV 4.14.2021

## What is Cloud Bigtable?

**FULLY MANAGED PETABYTE-SCALE NOSQL DATABASE**
**For heavy reads/writes**



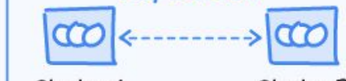

## SCALABILITY




Each node provides upto 10,000 QPS (read/write)

## FAILOVER FOR HIGH AVAILABILITY (HA)

Single/multi-region/global Replication



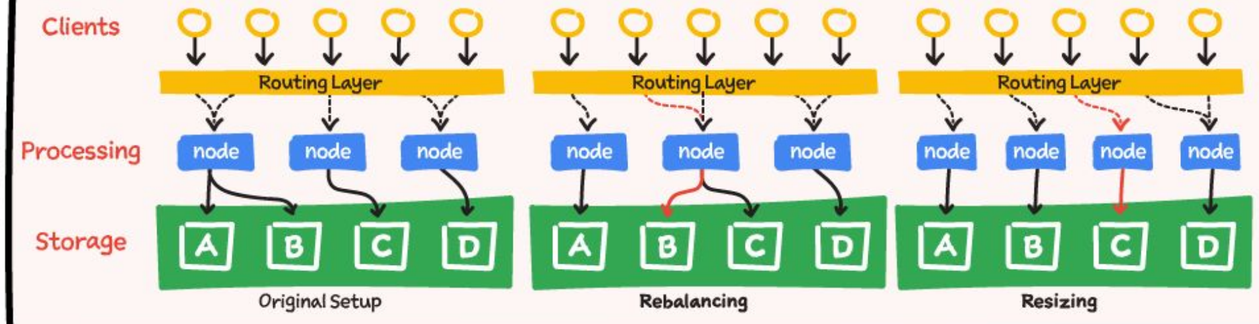

No manual steps to ensure consistency, repair data, or synchronize writes/deletes

## How does Cloud Bigtable Optimize throughput?

Cloud Bigtable separates processing from storage, each node has access to a group of database rows

Rebalance load automatically to improve performance

Resize nodes (with no downtime) for best overall throughput

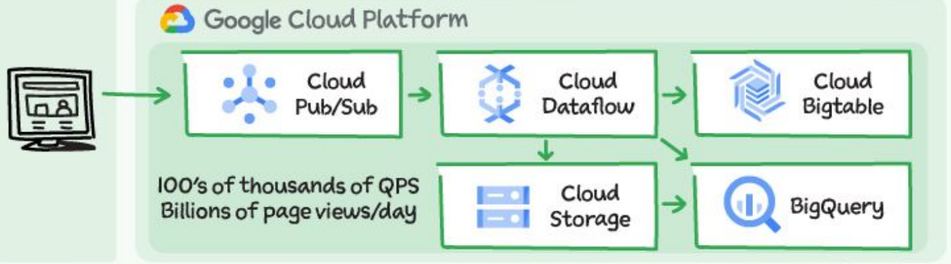



## Cloud Bigtable «Use case examples»

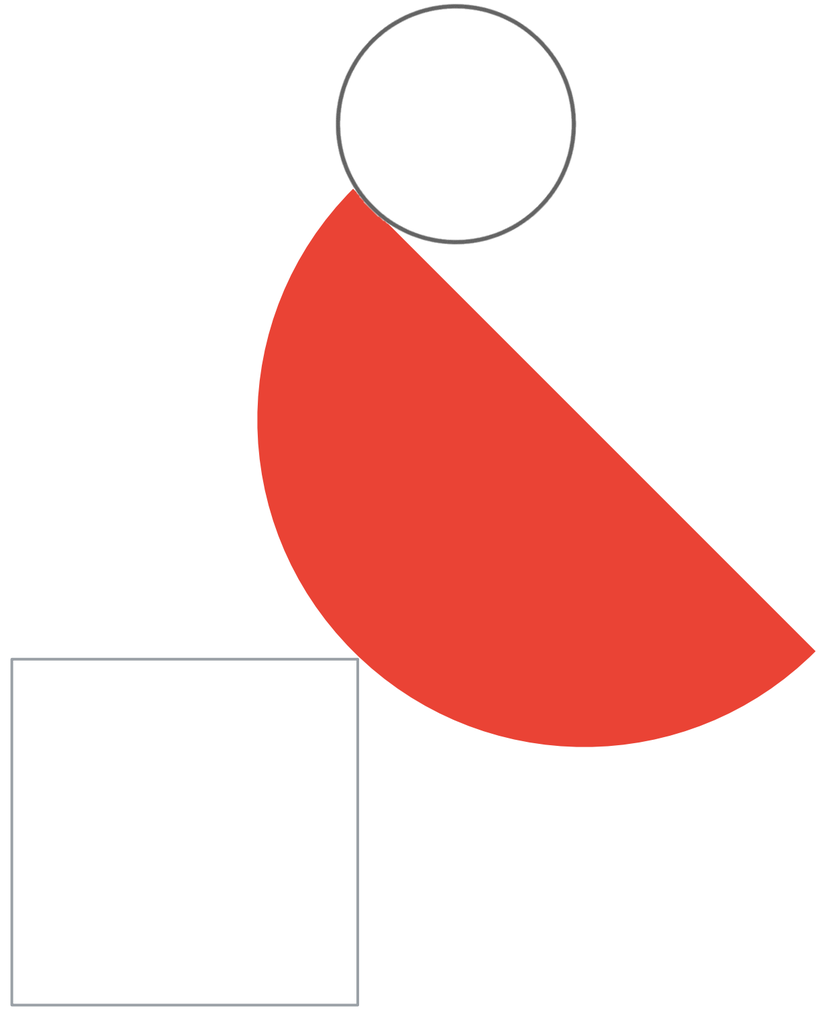
**REAL-TIME ANALYTICS FOR HUGE WORKLOADS**

FinTech · AdTech · Gaming · IoT · ML/AI Personalization

**REAL-TIME AD BIDDING PLATFORM**



# Comparing GCP storage solutions



# SQL vs noSQL

| SQL (aka ‘Relational’) | NoSQL (aka ‘Non-relational’) |
|---|---|
| “traditional” table-based RDBMSes | key-value, wide column, document |
| Strongly typed, fixed schemas | Dynamic schemas |
| Almost all ACID-compliant | Mostly BASE |
| Considerable percentage of logic can be done in database | Most of logic needs to be offloaded to application layer |
| Default choice for most monoliths | Suitable for some microservices |
| performance capped at some point (vertical scaling only, plus sharding, offloading read-only etc) | Processing nodes often separate from storage nodes (if network is fast enough) |
| In GCP: Cloud SQL, Cloud Spanner<br>Outside of GCP: MySQL, Oracle, PostgreSQL, Microsoft SQL Server. | In GCP: Firestore, Bigtable<br>Outside of GCP: MongoDB, Redis, Cassandra, HBase, CouchDB |

# OLTP vs OLAP

| **OLTransactionalP** | **OLAnalyticalP** |
|---|---|
| For processing data in transaction-oriented apps | Multi-dimensional, analytical queries used in BI, reporting, data mining etc |
| Large amounts of transactions | Large volume of data |
| A mix of Inserts, Updates, Deletes on individual records. | Loading data from source + selects. Optimized for high throughput reads on large number of records |
| Tables are normalized | Tables are not normalized |
| ACID & (mostly) SQL | SQL (sometimes NoSQL) |
| Cloud SQL, Cloud Spanner | BigQuery |

***Exam Tip:*** *Here you'll find a GREAT Decision tree for database choices on AWS, Microsoft Azure, Google Cloud Platform, and cloud-agnostic*

# Cloud Storage


| Overview | Ideal for |
| --- | --- |
| • Fully managed, highly reliable | • Images and videos |
| • Cost-efficient, scalable object/blob store | • Objects and blobs |
| • Objects access via HTTP requests | • Unstructured data |
| • Object name is the only key | • Static website hosting |

# Cloud Datastore


## Overview

- Fully managed NoSQL
- Scalable

## Ideal for

- Semi-structured application data
- Durable key-value data
- Hierarchical data
- Managing multiple indexes
- Transactions

# Cloud Firestore


## Overview

- Fully managed, serverless, NoSQL
- Scalable
- Native mobile and web client libraries
- Real-time updates

## Ideal for

- Document-oriented data
- Large collections of small documents
- Native mobile and web clients
- Durable key-value data
- Hierarchical data
- Managing multiple indexes
- Transactions

# Cloud Bigtable


## Overview

- High performance wide column NoSQL database service
- Sparsely populated table
- Can scale to billions of rows and thousands of columns
- Can store TB to PB of data

## Ideal for

- Operational applications
- Analytical applications
- Storing large amounts of single-keyed data
- MapReduce operations

# Cloud SQL


| Overview | Ideal for |
| --- | --- |
| • Managed service<br>  ○ Replication<br>  ○ Failover<br>  ○ Backups | • Web frameworks |
| • MySQL, PostgreSQL, and SQL Server<br>• Relational database service | • Structured data<br>• OLTP workloads |
| • Proxy allows for secure access to your Cloud SQL Second Generation instances without whitelisting | • Applications using MySQL/PGS |

# Cloud Spanner


## Overview

- Mission-critical relational database service
- Transactional consistency
- Global scale
- High availability
- Multi-region replication
- 99.999% SLA

## Ideal for

- Mission-critical applications
- High transactions
- Scale and consistency requirements

# BigQuery


## Overview

- Low-cost enterprise data warehouse for analytics
- Fully managed
- Petabyte scale
- Fast response times
- Serverless

## Ideal for

- Online Analytical Processing (OLAP) workloads
- Big data exploration and processing
- Reporting via Business Intelligence (BI) tools

# Which Database should I use?

#GCPSketchnotes @PVERGADIA THECLOUDGIRL.DEV

07.10.2021

## RELATIONAL

### Cloud SQL

Managed MySQL, PostgreSQL, SQL Server

**Good For:**

General purpose SQL DB

**Use Case:**

- Web frameworks
- ERP
- CRM
- Ecommerce and web
- SaaS application

### Cloud Spanner

Cloud-native with large scale, consistency, 99.999% availability

**Good For:**

RDBMS+ scale, HA, HTAP

**Use Case:**

- Gaming
- Global financial ledger
- Supply chain/ inventory management

### Bare Metal

Lift and shift Oracle workloads to Google Cloud

**Good For:**

RDBMS+ scale, HA, HTAP

**Use Case:**

- Legacy applications
- Data center retirement

## NON-RELATIONAL (NO SQL)

### DOCUMENT

### Firestore

Cloud Native, serverless, NoSQL document database, backend-as-a-service, global strong consistency, 99.999% SLA

**Good For:**

Large scale, complex hierarchical data

**Use Case:**

- Mobile/web/ IoT applications
- Real-time sync
- Offline sync
- Personalized apps

### KEY VALUE

### Cloud Bigtable

Cloud-native NoSQL wide-column store for large scale, low-latency workloads

**Good For:**

Heavy read + write, events

**Use Case:**

- Personalization
- Adtech
- Recommendation engines
- Fraud detection

## IN MEMORY

### Memory Store

Fully managed Redis and Memcached for sub-millisecond data access

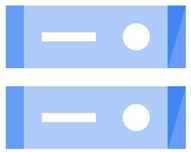
**Good For:**

In-memory and Key-value store

**Use Case:**

- Caching
- Session store
- Gaming
- Personalization
- Leaderboard
- Adtech
- Social chat or news feed

# Comparing storage and database

| In memory | Relational | | Non-relational | | Object | Warehouse |
|---|---|---|---|---|---|---|
|  |  |  |  |  |  |  |
| App Engine Memcache | Cloud SQL | Cloud Spanner | Firestore | Cloud Bigtable | Cloud Storage | BigQuery |
| **Good for:** Web/mobile apps, gaming | **Good for:** Web frameworks | **Good for:** RDBMS+scale, HA, HTAP | **Good for:** Hierarchical, mobile, web | **Good for:** Heavy read + write, events | **Good for:** Binary or object data | **Good for:** Enterprise data warehouse |
| **Such as:** Game state, user sessions | **Such as:** CMS, eCommerce | **Such as:** User metadata, Ad/Fin/MarTech | **Such as:** User profiles, Game State | **Such as:** AdTech, financial, IoT | **Such as:** Images, media serving, backups | **Such as:** Analytics, dashboards |

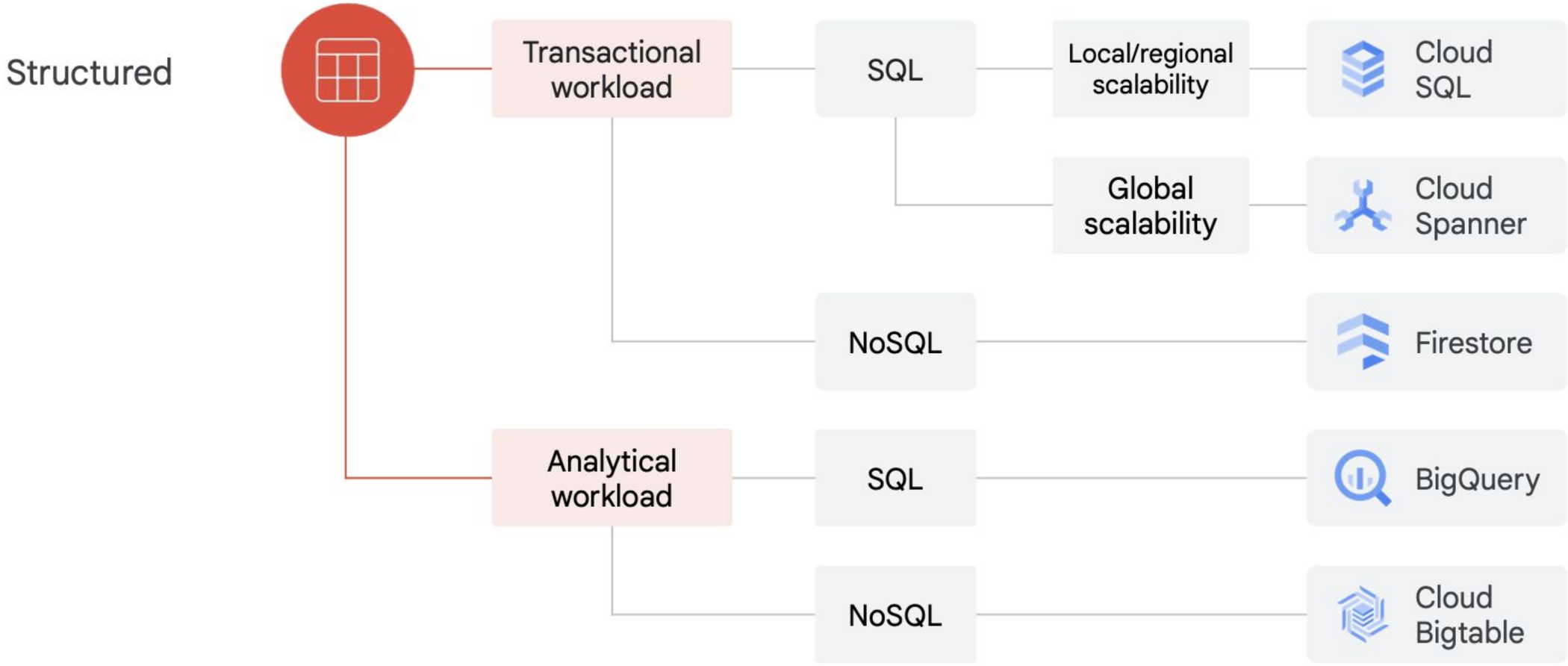
**TIP**

Try to read from bottom up (what's the most appropriate storage for analytics workloads? What's good for global, horizontally scalable RDBMS?

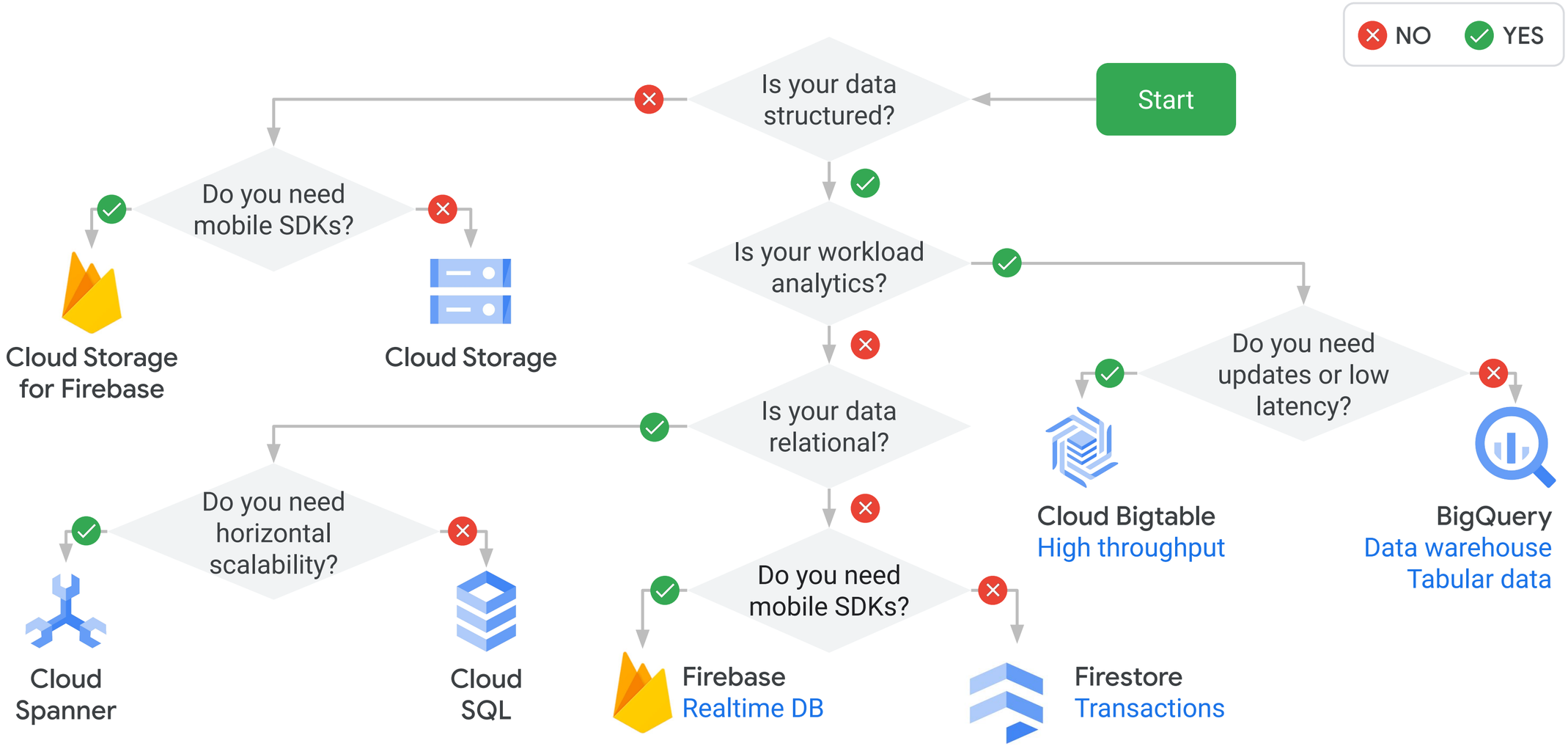
# Comparing storage options: Technical details

| | Firestore | Bigtable | Cloud Storage | Cloud SQL | Cloud Spanner | BigQuery |
|---|---|---|---|---|---|---|
| **Type** | NoSQL document | NoSQL wide column | Blobstore | Relational SQL for OLTP | Relational SQL for OLTP | Relational SQL for OLAP |
| **Transactions** | Yes | Single-row | No | Yes | Yes | No |
| **Complex queries** | Yes | No | No | Yes | Yes | Yes |
| **Capacity** | Terabytes+ | Petabytes+ | Petabytes+ | 10,230 GB | Petabytes | Petabytes+ |
| **Unit size** | 1 MB/entity | ~10 MB/cell ~100 MB/row | 5 TB/object | Determined by DB engine | 10,240 MiB/ row | 10 MB/row |

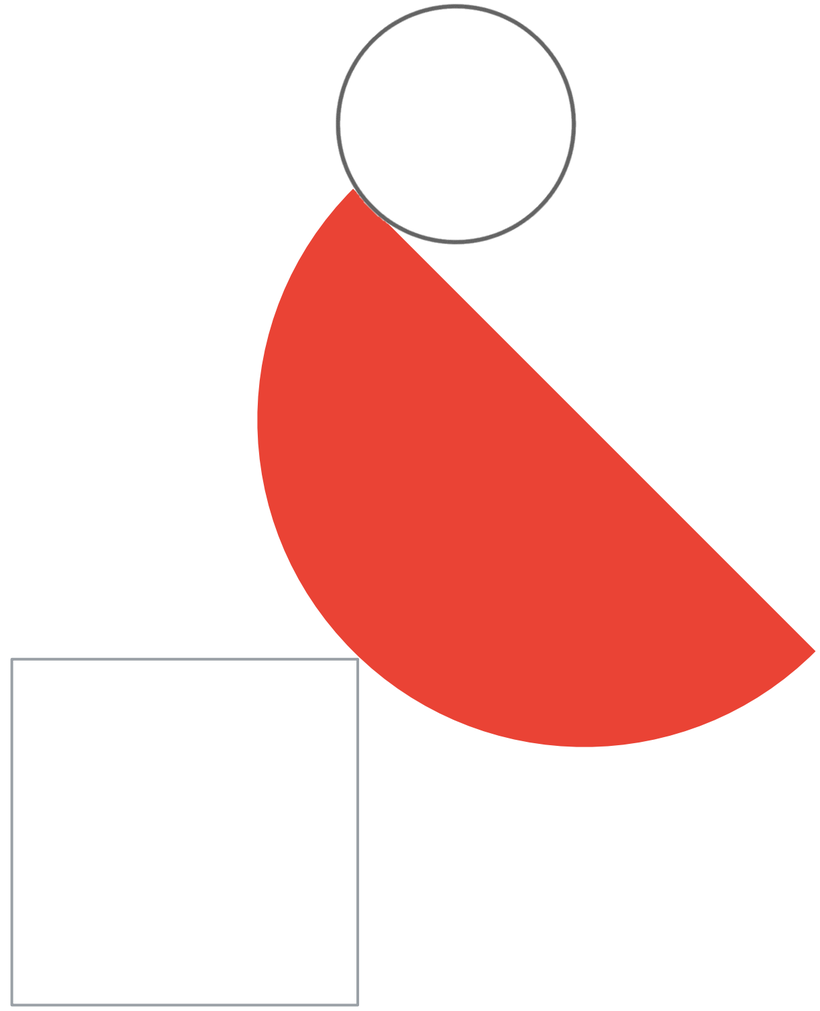
# GCP: storage service decision tree



# GCP: storage service decision tree (version #2)



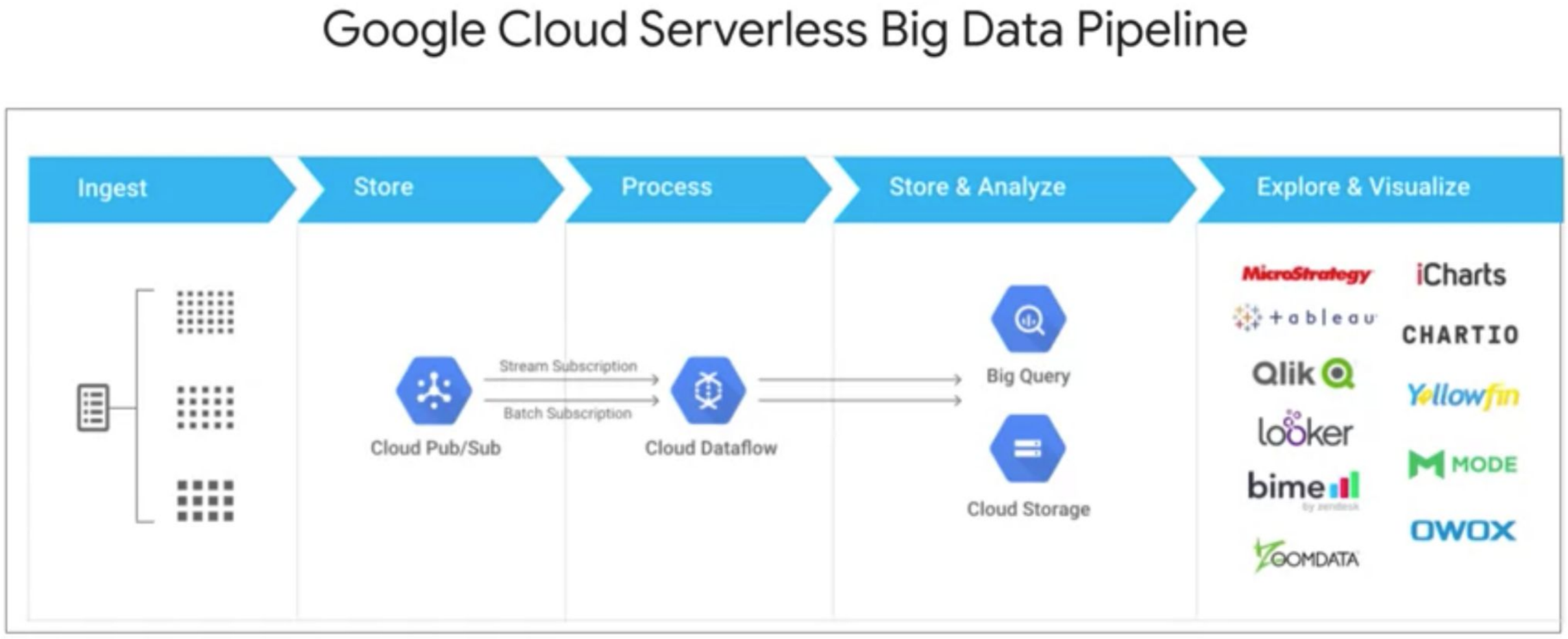
# Mountkirk Games case study analysis



Google Cloud


Mountkirk Games

# Analytics pipeline



# Proposed Technical Solutions

MountKirk Games

- Containers
  - GKE with multiple regional clusters and Workload Identity
  - Services exposed via global load balancers
  - (possibly) Connect the clusters with Anthos (which gives additional benefits: control and encryption of the traffic, centralized management etc).
  - Cluster and workload autoscaling -> either configure GKE autoscalers, or just deploy GKE clusters in AutoPilot mode.
  - Additional Node Pools with preemptible instances.
  - Additional Node Pools with GPUs.
- Cloud Spanner as database for leaderboards.
  - Deployed in multi-region setup to minimize latency from GKE clusters to Spanner.
- CI / CD pipeline for rapid deployment:
  - Cloud Source Repositories to store and work on source code
  - Cloud Build
  - Artifact Registry (previously: Container Registry, focused only on containers) for storing artifacts after they're built
  - Cloud Deploy; Alternatively, 3rd party software (Jenkins, Spinnaker etc)
- Migrate for GKE/Anthos -> migrating VM-based workloads to Kubernetes (GKE).
- Cloud Operations Suite for monitoring / telemetry.
  - GCS buckets / BigQuery to store logs for longer periods of time.
- Advanced analytics: source (GKE game servers or GCS buckets) -> Pub/Sub -> Dataflow -> BigQuery + Data Studio / Looker
- GCP Game Servers: possibly, but the architecture above will handle it just fine as well.

# [Mountkirk Games case study] Diagnostic Question #1

For this question, refer to the Mountkirk Games case study. Mountkirk Games wants to migrate from their current analytics and statistics reporting model to one that meets their technical requirements on Google Cloud Platform.

Which two steps should be part of their migration plan? (Choose two.)

A. Evaluate the impact of migrating their current batch ETL code to Cloud Dataflow.

B. Write a schema migration plan to denormalize data for better performance in BigQuery.

C. Draw an architecture diagram that shows how to move from a single MySQL database to a MySQL cluster.

D. Load 10 TB of analytics data from a previous game into a Cloud SQL instance, and run test queries against the full dataset to confirm that they complete successfully.

E. Integrate Cloud Armor to defend against possible SQL injection attacks in analytics files uploaded to Cloud Storage.

# [Mountkirk Games case study] Diagnostic Question #1

For this question, refer to the Mountkirk Games case study. Mountkirk Games wants to migrate from their current analytics and statistics reporting model to one that meets their technical requirements on Google Cloud Platform.

Which two steps should be part of their migration plan? (Choose two.)

**A. Evaluate the impact of migrating their current batch ETL code to Cloud Dataflow.**

**B. Write a schema migration plan to denormalize data for better performance in BigQuery.**

C. Draw an architecture diagram that shows how to move from a single MySQL database to a MySQL cluster.

D. Load 10 TB of analytics data from a previous game into a Cloud SQL instance, and run test queries against the full dataset to confirm that they complete successfully.

E. Integrate Cloud Armor to defend against possible SQL injection attacks in analytics files uploaded to Cloud Storage.

# [Mountkirk Games case study] Diagnostic Question #2

Mountkirk Games wants to limit the physical location of resources to their operating Google Cloud regions.

What should you do?

A. Configure an organizational policy which constrains where resources can be deployed.

B. Configure IAM conditions to limit what resources can be configured.

C. Configure the quotas for resources in the regions not being used to 0.

D. Configure a custom alert in Cloud Monitoring so you can disable resources as they are created in other regions.

# [Mountkirk Games case study] Diagnostic Question #2

Mountkirk Games wants to limit the physical location of resources to their operating Google Cloud regions.

What should you do?

**A. Configure an organizational policy which constrains where resources can be deployed.**

B. Configure IAM conditions to limit what resources can be configured.

C. Configure the quotas for resources in the regions not being used to 0.

D. Configure a custom alert in Cloud Monitoring so you can disable resources as they are created in other regions.

# [Mountkirk Games case study] Diagnostic Question #3

Mountkirk Games wants you to design their new testing strategy. How should the test coverage differ from their existing backends on the other platforms?

A. Tests should scale well beyond the prior approaches

B. Unit tests are no longer required, only end-to-end tests

C. Tests should be applied after the release is in the production environment

D. Tests should include directly testing the Google Cloud Platform (GCP) infrastructure

# [Mountkirk Games case study] Diagnostic Question #3

Mountkirk Games wants you to design their new testing strategy. How should the test coverage differ from their existing backends on the other platforms?

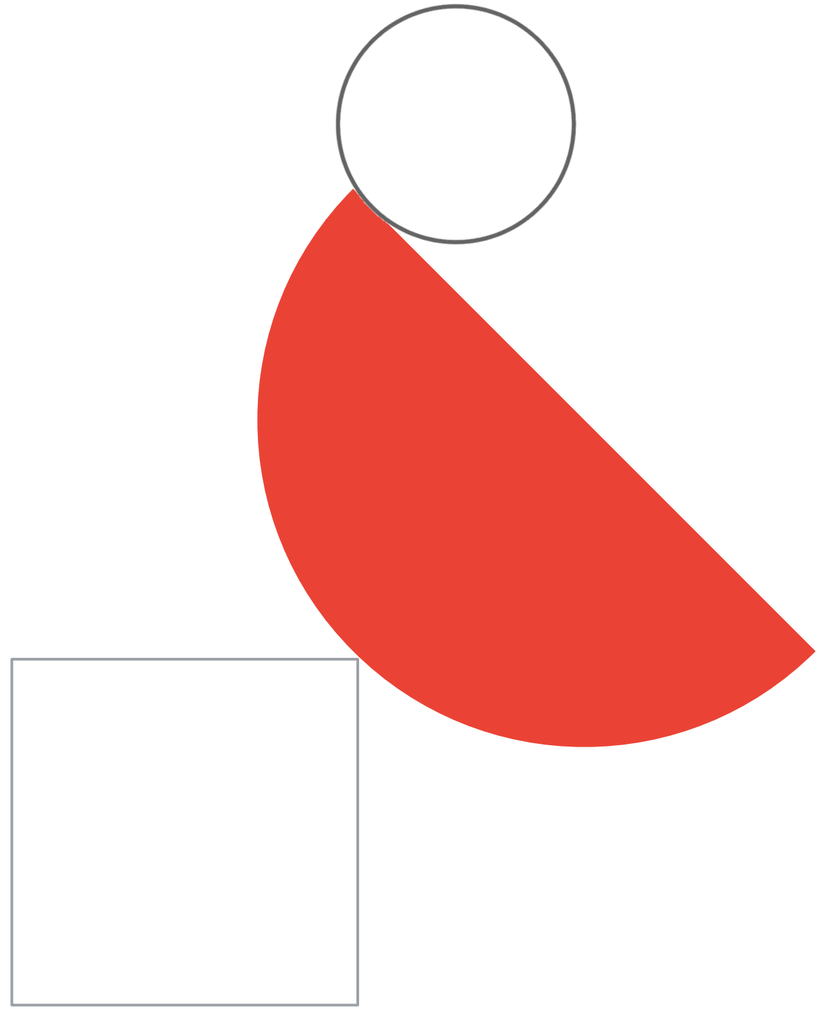
A. Tests should scale well beyond the prior approaches

B. Unit tests are no longer required, only end-to-end tests

C. Tests should be applied after the release is in the production environment

**D. Tests should include directly testing the Google Cloud Platform (GCP) infrastructure**

# [optional] Links to useful materials



# Optional materials 1

**[ READING ]**

- Make sure you know the differences between BigQuery and BigTable.
- Be aware how BigQuery table partitioning works.

**[ VIDEOS ]**

- Cloud Networking 104 (Load Balancers): Cloud OnAir: Networking 104 - Everything You Need to Know About Load Balancers on GCP
- Querying external data with BigQuery
- BigQuery: What is BigQuery?
- [IMPORTANT TO KNOW] Sharing BigQuery data with others: Protect data with authorized views
- BigTable: What is Cloud Bigtable?
- Data Studio introduction: Data Studio in a minute
- BigTable: What can you do with Bigtable?
- Cloud Spanner [5 min]: What is Cloud Spanner | Cloud Spanner Explained | Cloud Native Relational Database
- Cloud Spanner [2x5min]: How to set up a Cloud Spanner instance & Cloud Spanner: Database deep dive
- Introduction to Firestore: Introduction to Firestore | NoSQL Document Database
- What is Dataprep? (do not confuse with Dataproc, Dataflow and other Data<service>) No code data wrangling with Dataprep #GCPSketchnote
- Decision tree to migrate Apache Hadoop workloads to Dataproc: Decision tree to migrate Apache Hadoop workloads to Dataproc #GCPSketchnote

# Optional materials 2

- Creating a large Dataproc Cluster with preemptible VMs: Creating a large Dataproc Cluster with preemptible VMs
- What is Cloud Build?: What is Cloud Build? #GCPSketchnote
- Three ways to improve CI/CD in your serverless app
- How to protect secrets with Secret Manager: Level Up - Secret Manager
- What is Cloud Armor?: What is Cloud Armor? #GCPSketchnote

**[ PODCASTS ]**

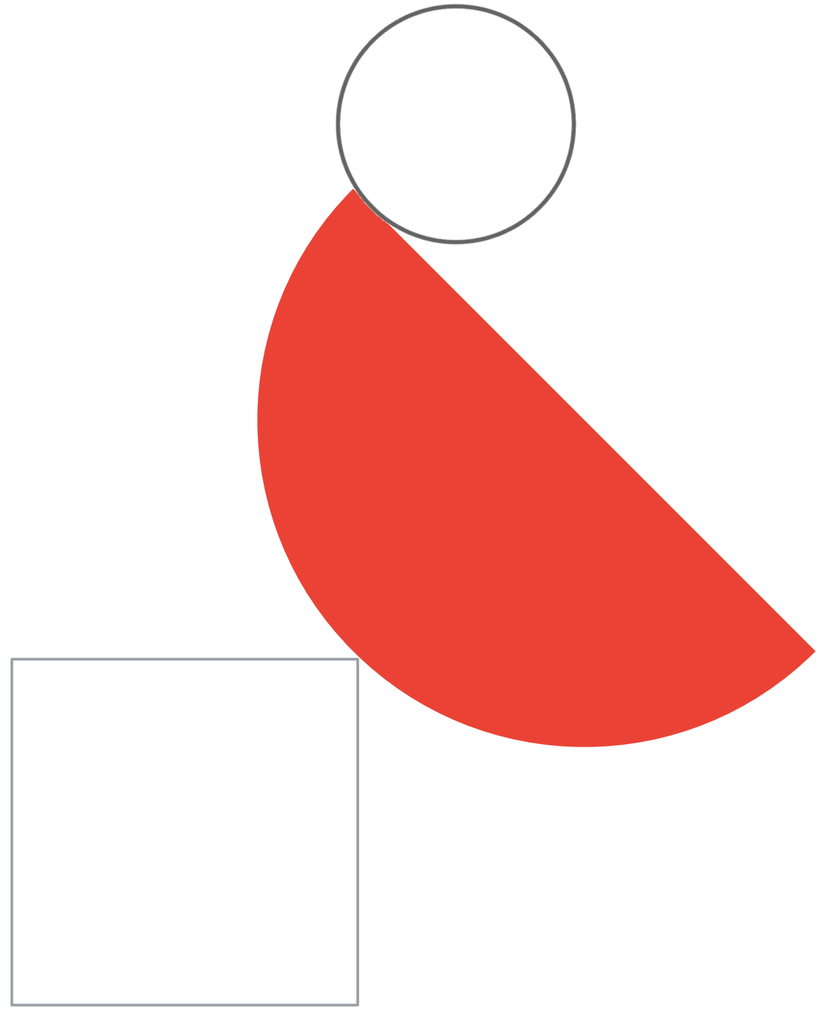
- BigQuery Admin Reference Guides
- Firebase (not to be mixed up with Firestore!)
- Cloud Functions
- Cloud BigTable

**[ DEEP DIVES ]**

- BigQuery and Cloud Spanner deep dive: Under the hood of Google Cloud data technologies: BigQuery and Cloud Spanner
- (~20 mins) BigQuery lab that will familiarize you with basics and show interesting insights at the same time.

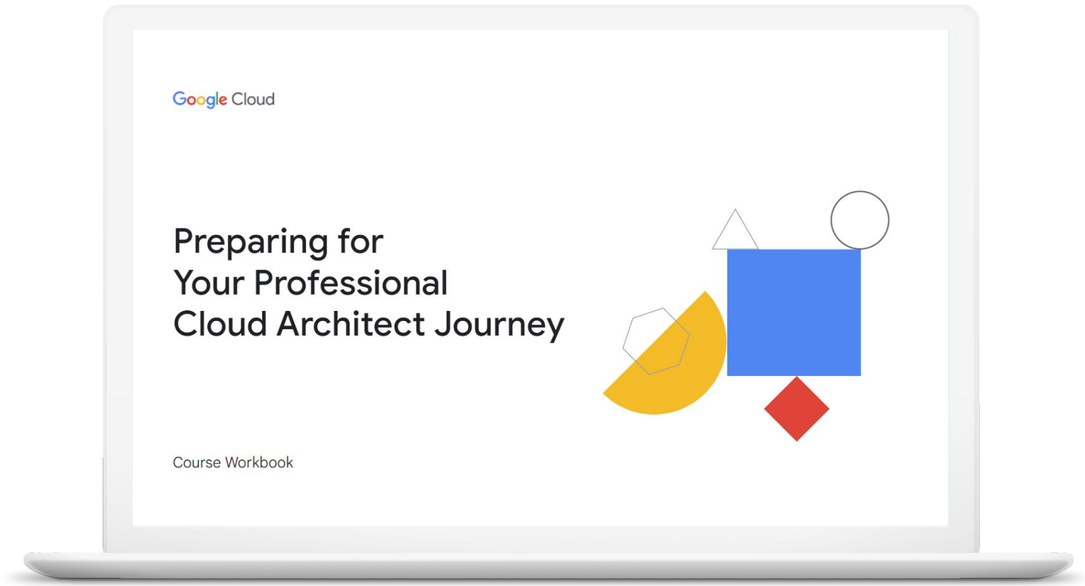
# Diagnostic Questions

## for Exam Guide Section 4: Analyzing and optimizing technical and business processes



# PCA Exam Guide Section 4:

Analyzing and optimizing technical and business processes




4.1 Analyzing and defining technical processes

4.2 Analyzing and defining business processes

4.3 Developing procedures to ensure reliability of solutions in production

# 4.1 Analyzing and defining technical processes

Considerations include:

- Software development life cycle (SDLC)
- Continuous integration / continuous deployment
- Troubleshooting / root cause analysis best practices
- Testing and validation of software and infrastructure
- Service catalog and provisioning
- Business continuity and disaster recovery

# 4.1 Diagnostic Question 01 Discussion

You are asked to implement a lift and shift operation for Cymbal Direct's Social Media Highlighting service. You **compose a Terraform configuration file** to build all the necessary Google Cloud resources.

What is the next step in the Terraform workflow for this effort?

A. **Commit the configuration file** to your software repository.

B. Run **terraform plan** to verify the contents of the Terraform configuration file.

C. Run **terraform apply** to deploy the resources described in the configuration file.

D. Run **terraform init** to download the necessary provider modules.

# 4.1 Diagnostic Question 01 Discussion

You are asked to implement a lift and shift operation for Cymbal Direct's Social Media Highlighting service. You **compose a Terraform configuration file** to build all the necessary Google Cloud resources.

What is the next step in the Terraform workflow for this effort?

A. **Commit the configuration file** to your software repository.

B. Run **terraform plan** to verify the contents of the Terraform configuration file.

C. Run **terraform apply** to deploy the resources described in the configuration file.

D. Run **terraform init** to download the necessary provider modules.

# 4.1 Diagnostic Question 02 Discussion

You have implemented a manual **CI/CD process for the container services** required for the next implementation of the Cymbal Direct's Drone Delivery project. You want to **automate the process**.

What should you do?

A. **Implement and reference a source repository** in your Cloud Build configuration file.

B. **Implement a build trigger** that applies your build configuration when a new software update is committed to Cloud Source Repositories.

C. **Specify the name** of your Container Registry in your Cloud Build configuration.

D. **Configure and push a manifest fil**e into an environment repository in Cloud Source Repositories.

# 4.1 Diagnostic Question 02 Discussion

You have implemented a manual **CI/CD process for the container services** required for the next implementation of the Cymbal Direct's Drone Delivery project. You want to **automate the process**.

What should you do?

A. **Implement and reference a source repository** in your Cloud Build configuration file.

B. **Implement a build trigger** that applies your build configuration when a new software update is committed to Cloud Source Repositories.

C. **Specify the name** of your Container Registry in your Cloud Build configuration.

D. **Configure and push a manifest fil**e into an environment repository in Cloud Source Repositories.

# 4.1 Diagnostic Question 03 Discussion

You have an application implemented on **Compute Engine.** You want to **increase the durability** of your application.

What should you do?

A. Implement a **scheduled snapshot** on your Compute Engine instances.

B. Implement a **regional managed instance group.**

C. Monitor your application's usage metrics and implement **autoscaling**.

D. Perform **health checks** on your Compute Engine instances.

# 4.1 Diagnostic Question 03 Discussion

You have an application implemented on **Compute Engine.** You want to **increase the durability** of your application.

What should you do?

A. Implement a **scheduled snapshot** on your Compute Engine instances.

B. Implement a **regional managed instance group.**

C. Monitor your application's usage metrics and implement **autoscaling**.

D. Perform **health checks** on your Compute Engine instances.

# 4.1 Diagnostic Question 04 Discussion

Developers on your team frequently write new versions of the code for one of your applications. You want to **automate the build process when updates are pushed to Cloud Source Repositories.**

What should you do?

A. Implement a **Cloud Build configuration file** with build steps.

B. Implement a **build trigger** that references your repository and branch.

C. Set proper **permissions** for Cloud Build to access deployment resources.

D. Upload **application updates and Cloud Build configuration files** to Cloud Source Repositories.

# 4.1 Diagnostic Question 04 Discussion

Developers on your team frequently write new versions of the code for one of your applications. You want to **automate the build process when updates are pushed to Cloud Source Repositories.**

What should you do?

A. Implement a **Cloud Build configuration file** with build steps.

B. Implement a **build trigger** that references your repository and branch.

C. Set proper **permissions** for Cloud Build to access deployment resources.

D. Upload **application updates and Cloud Build configuration files** to Cloud Source Repositories.

# 4.1 Diagnostic Question 05 Discussion

Your development team used **Cloud Source Repositories**, **Cloud Build**, and **Artifact Registry** to successfully implement the build portion of an application's CI/CD process.. However, the deployment process is erroring out. Initial troubleshooting shows that the **runtime environment does not have access to the build images.** You need to advise the team on how to resolve the issue.

What could cause this problem?

A. The r**untime environment does not have permissions to the Artifact Registry** in your current project.

B. The **runtime environment does not have permissions** to **Cloud Source Repositories** in your current project.

C. The Artifact Registry might be in a **different project.**

D. You need to specify the Artifact Registry **image by name.**

# 4.1 Diagnostic Question 05 Discussion

Your development team used **Cloud Source Repositories**, **Cloud Build**, and **Artifact Registry** to successfully implement the build portion of an application's CI/CD process.. However, the deployment process is erroring out. Initial troubleshooting shows that the **runtime environment does not have access to the build images.** You need to advise the team on how to resolve the issue.

What could cause this problem?

A. The r**untime environment does not have permissions to the Artifact Registry** in your current project.

B. The **runtime environment does not have permissions** to **Cloud Source Repositories** in your current project.

C. The Artifact Registry might be in a **different project.**

D. You need to specify the Artifact Registry **image by name.**

# 4.1 Diagnostic Question 06 Discussion

You are implementing a **disaster recovery plan** for the cloud version of your drone solution. **Sending videos to the pilots** is crucial from an operational perspective.

What design pattern should you choose for this part of your architecture?

A. **Hot** with a **low** recovery time objective (RTO)

B. **Warm** with a **high** recovery time objective (RTO)

C. **Cold** with a **low** recovery time objective (RTO)

D. **Hot** with a **high** recovery time objective (RTO)

# 4.1 Diagnostic Question 06 Discussion

You are implementing a **disaster recovery plan** for the cloud version of your drone solution. **Sending videos to the pilots** is crucial from an operational perspective.

What design pattern should you choose for this part of your architecture?

A. **Hot** with a **low** recovery time objective (RTO)

B. **Warm** with a **high** recovery time objective (RTO)

C. **Cold** with a **low** recovery time objective (RTO)

D. **Hot** with a **high** recovery time objective (RTO)

# 4.1 Diagnostic Question 07 Discussion

The number of requests received by your application is **nearing the maximum specified in your design**. You want to **limit the number of incoming requests** until the system can handle the workload.

What design pattern does this situation describe?

A. Applying a **circuit breaker**
B. Applying **exponential backoff**
C. Increasing **jitter**
D. Applying **graceful degradation**

# 4.1 Diagnostic Question 07 Discussion

The number of requests received by your application is **nearing the maximum specified in your design**. You want to **limit the number of incoming requests** until the system can handle the workload.

What design pattern does this situation describe?

A. Applying a **circuit breaker**
B. Applying **exponential backoff**
C. Increasing **jitter**
D. Applying **graceful degradation**

# 4.1 Diagnostic Question 08 Discussion

The pilot subsystem in your Delivery by Drone service is critical to your service. You want to ensure that **connections to the pilots can survive a VM outage** without affecting connectivity.

What should you do?

A. Configure proper **startup scripts** for your VMs.

B. Deploy a **load balancer** to distribute traffic across multiple machines.

C. Create **persistent disk snapshots**.

D. Implement a **managed instance group** and **load balancer**.

# 4.1 Diagnostic Question 08 Discussion

The pilot subsystem in your Delivery by Drone service is critical to your service. You want to ensure that **connections to the pilots can survive a VM outage** without affecting connectivity.

What should you do?

A. Configure proper **startup scripts** for your VMs.

B. Deploy a **load balancer** to distribute traffic across multiple machines.

C. Create **persistent disk snapshots**.

D. Implement a **managed instance group** and **load balancer**.

# 4.1 Diagnostic Question 09 Discussion

Cymbal Direct wants to improve its drone pilot interface. You want to **collect feedback on proposed changes** from the community of pilots **before rolling out updates systemwide**.

What type of deployment pattern should you implement?

A. You should implement **canary testing.**

B. You should implement **A/B testing.**

C. You should implement a **blue/green deployment.**

D. You should implement an **in-place release.**

# 4.1 Diagnostic Question 09 Discussion

Cymbal Direct wants to improve its drone pilot interface. You want to **collect feedback on proposed changes** from the community of pilots **before rolling out updates systemwide**.

What type of deployment pattern should you implement?

A. You should implement **canary testing.**

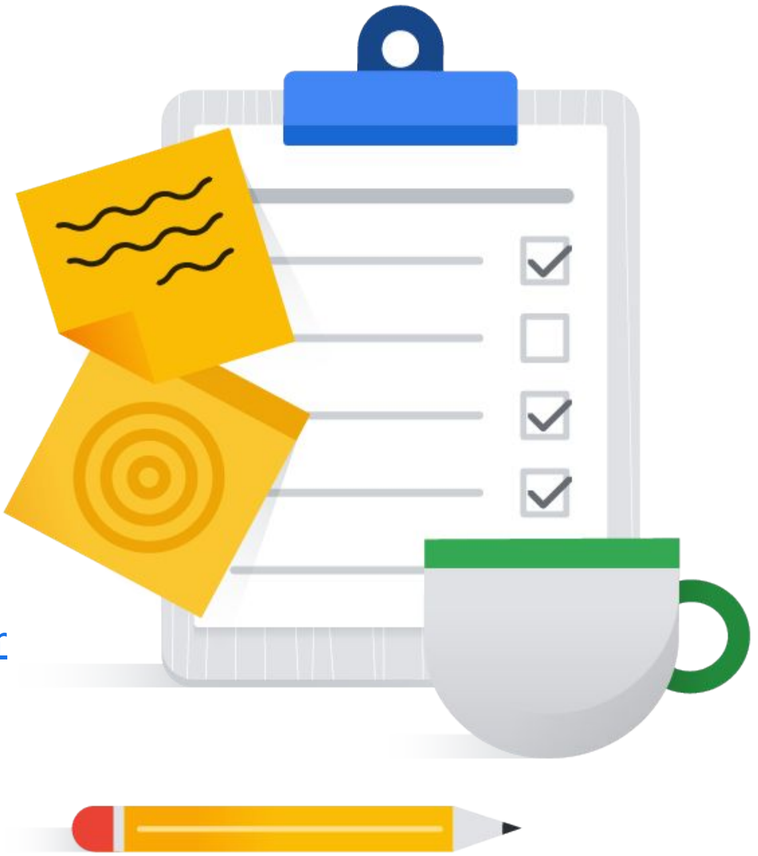
B. You should implement **A/B testing.**

C. You should implement a **blue/green deployment.**

D. You should implement an **in-place release.**

# 4.1 Analyzing and defining technical processes

## Resources to start your journey



[Securing the software development lifecycle with Cloud Build and SLSA]

[CI/CD with Google Cloud]

[Site Reliability Engineering]

[DevOps tech: Continuous testing | Google Cloud]

[Application deployment and testing strategies | Cloud Architecture Center]

[Chapter 17 - Testing for Reliability]

[Service Catalog documentation | Google Cloud]

[What is Disaster Recovery? | Google Cloud]

[API design guide]

# 4.2 Analyzing and defining business processes

Considerations include:

- Stakeholder management (e.g. influencing and facilitation)
- Change management
- Team assessment / skills readiness
- Decision-making processes
- Customer success management
- Cost optimization / resource optimization (capex / opex)

# 4.2 Analyzing and defining business processes

## Resources to start your journey

[What is Digital Transformation?]

[Cloud Cost Optimization: Principles for Lasting Success]

[Cost Optimization on Google Cloud for Developers and Operators]

[Certification solutions for Team Readiness]

# 4.3 Developing procedures to ensure reliability of solutions in production

- Chaos engineering
- Penetration testing

# 4.3 Diagnostic Question 10 Discussion

You want to establish **procedures for testing the resilience** of the delivery-by-drone solution.

A. **Block access** to **storage assets** in one of your zones.
B. Inject a **bad health check f**or one or more of your resources.
C. **Load test your application** to see how it responds.
D. **Block access** to all **resources** in a zone.

How would you simulate a scalability issue?

# 4.3 Diagnostic Question 10 Discussion

You want to establish **procedures for testing the resilience** of the delivery-by-drone solution.

A. **Block access** to **storage assets** in one of your zones.

B. Inject a **bad health check f**or one or more of your resources.

C. **Load test your application** to see how it responds.

D. **Block access** to all **resources** in a zone.

How would you simulate a scalability issue?

# 4.3 Developing procedures to ensure reliability of solutions in production

## Resources to start your journey

Site Reliability Engineering

Site Reliability Engineering (SRE) | Google Cloud

Patterns for scalable and resilient apps | Cloud Architecture Center

How to achieve a resilient IT strategy with Google Cloud

Patterns for scalable and resilient apps | Cloud Architecture Center

Disaster recovery planning guide | Cloud Architecture Center

# Make sure to...

## Enjoy the journey as much as the destination!