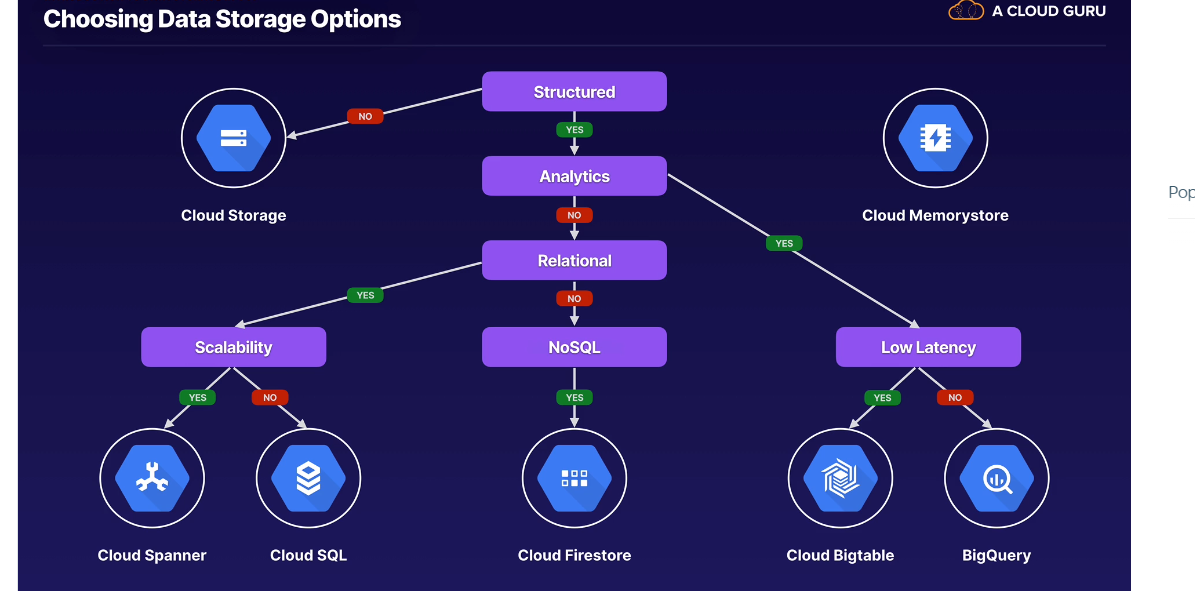
Introduction to Data Storage in GCP:



Cloud Storage:

* Unstructured object storage.
* Regional, dual-region or multi-region.
* Standard, nearline, or coldline.
* Storage event triggers (key feature when handling data storage) – achieved using Google Pub/Sub

Cloud Bigtable:

* Petabyte-scale NoSQL DB.
* High throughput and scalability.
* Designed for wide column key/value data.
* Designed for high volume writes making it suitable for time-series, transactional, IoT Data.

BigQuery:

* Petabyte-scale analytics data warehouse.
* Fast SQL queries across large datasets.
* Can be used for analytics, foundations for BI or train ML/AI models on it.
* Public datasets available.

Cloud Spanner:

* Google’s global SQL-based relational DB.
* Horizontal scalability and high availability.
* Strong Consistency.
* Gaining traction in the Financial Sector.

Cloud SQL:

* Managed MySQL and PostgreSQL.
* Built-in backups, replicas and failover.
* Doesn’t scale horizontally but scale vertically.
* Also offer MSFT SQL Server.

Cloud Firestore:

* Fully managed NoSQL document DB
* Designed for large collections of small JSON documents – more like Mongo DB
* Also provides real-time DB with offline caching and mobile SDKs out of the box.
* Offers strong consistency.

Cloud Memorystore:

* Provides managed Redis instances.
* Mainly used as in-memory DB, cache or message broker.
* Built in HA – depending on tier selected.
* Can be vertically scaled.

**Working with Data:**

Managing Data:

1. Consider Sources and Sinks:

* Does the data require preparation work? Is the source format appropriate for the data sink?

1. Structured and unstructured.

* How is data being processed and stored? Does structure data conform to the correct model?

1. Batched and streaming:

* Are pipelines required to ingest data? Are latency and time-windowing a concern?

Data Modelling:

* Structured data always requires a consistent model.
  + The model may already be in place
  + Data maybe require preparation or transformation
* Data Modeling Stages

1. Conceptual – What are the entities in my data? What are their attributes and relationships? – e,g when building a data model for a cake shop, entities include

* CUSTOMERS + CAKE

1. Logical – What are the structures of my entities? Can the model be normalized?

* CUSTOMER ID (INT), NAME (STRING)
* CAKE ID (INT), NAME(STRING), VEGAN (BOOL)

1. Physical – How will I implement my DB? What keys and indexes do I need?

* CUSTOMER ID (INT) PRIMARY NAME (STRING)
* CAKE ID (INT) PRIMARY NAME(STRING) VEGAN (BOOL)

Relational Data:

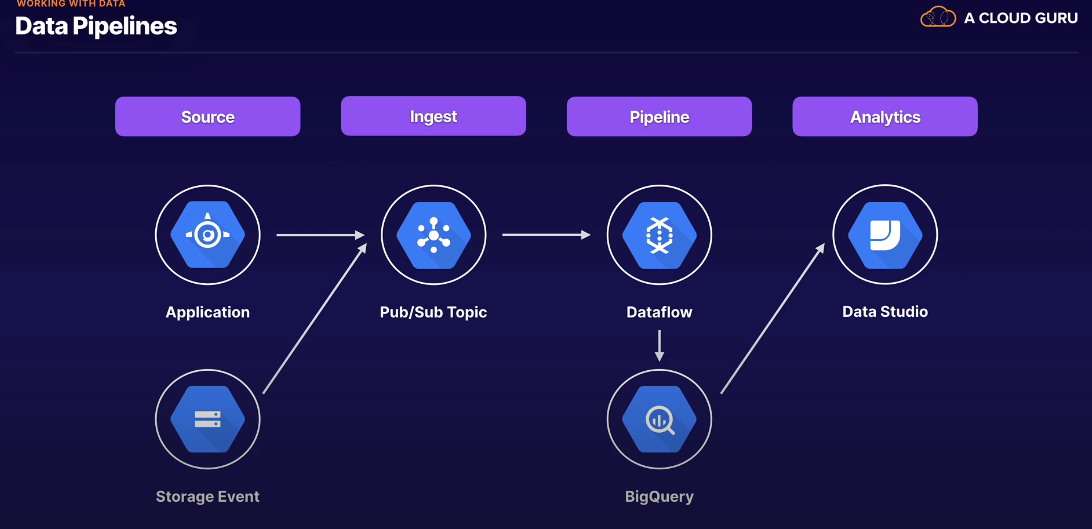
* Good Relational Schema Design.
* Normalization & reducing waste.
* Primary Keys and table relationships.
* Accuracy and integrity.

Non-relational Data:

* Simple key/value – e.g., Redis
* Flexible document store – you can put hierarchical docs.
* High volume columnar DB- use for Time series

Data Pipelines:

* Cloud Pub/Sub – a good day to decouple the ingestion of data from how it’s later processed.



Google Cloud Storage:

* Fully managed object storage.
* Multiple storage classes.
* Secure and durable.

Storage Classes:

* Standard – 99.99% regional availability, 99.99% availability in multi & dual-regions
* Nearline – 30 days min storage.
* 99.9% regional; 99.95% multi & dual
* Coldline – 90 days min
* Archive - 365 days min storage. – same availability as cold + nearline

Objects in cloud Storage:

* Objects are stored as opaque data
* Objects are immutable.
* Overwrites are atomic
* Objects can be versioned.

Advanced features:

* Parallel uploads of composite objects.
* Integrity checking – you can precalculated MD5.
* Supports Gzip transcoding
* Requestor pays – whoever is requesting pas for resources.

GCS Costs:

* Operation charges – (Classes: A-uploading, B-downloading, Free- deleting data).
* Network charges – egress charges
* Data retrieval charges (e.g cold)

Lifecycle Mgt:

* Apply a lifecycle config to a bucket.
* GCS periodically checks config.
* Matching rules applied to objects -can be used to delete objects or set storage classes.

Security and Access Control:

* IAM for bulk access to buckets.
* ACLs for granular access to buckets.
* Signed URLs.
* Signed policy documents

Service Accounts: