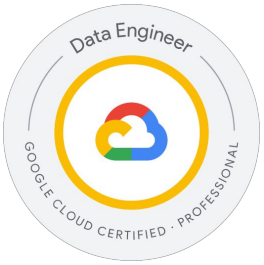
Google Cloud

Partner Certification Academy



Professional Data Engineer

The information in this presentation is classified:

Google confidential & proprietary

⚠ This presentation is shared with you under NDA.

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

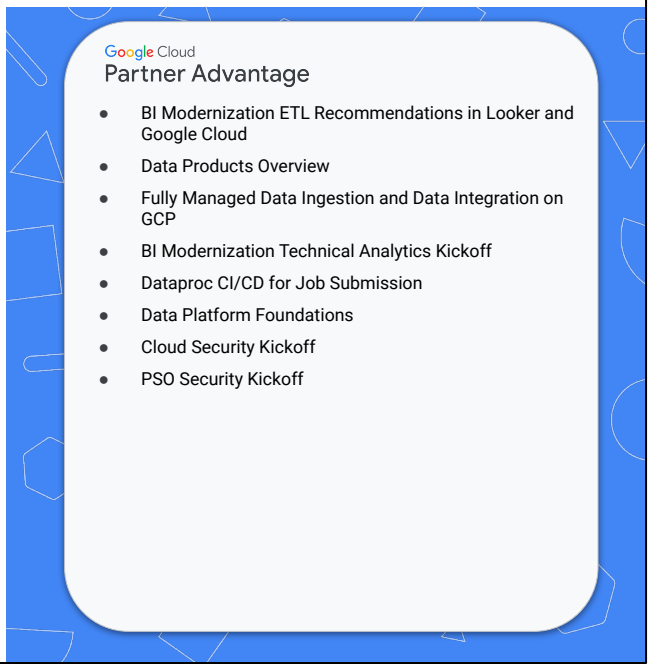
Thank you!

Google Cloud

Source Materials

Some of this program's content has been sourced from the following resources:

- Partner Advantage



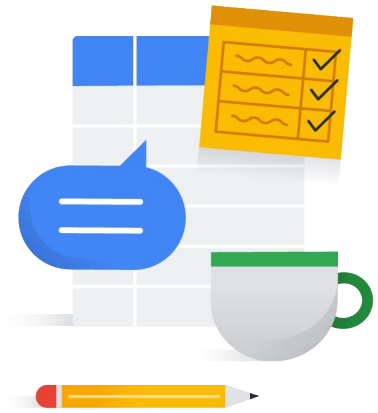
Google Cloud

Session logistics

- When you have a question, please:
  - Click the Raise hand button in Google Meet.
  - Or add your question to the Q&A section of Google Meet.
  - Please note that answers may be deferred until the end of the session.
- These slides are available in the Student Lecture section of your Qwiklabs classroom.
- The session is **not recorded.**
- Google Meet does not have persistent chat.
  - If you get disconnected, you will lose the chat history.
  - Please copy any important URLs to a local text file as they appear in the chat.

Program issues or concerns?

- Problems with **accessing** Cloud Skills Boost for Partners
  - partner-training@google.com
- Problems with **a lab** (locked out, etc.)
  - support@qwiklabs.com
- Problems with accessing Partner Advantage
  - https://support.google.com/googlecloud/topic/9198654



- Problems with accessing **Cloud Skills Boost for Partners**
  - partner-training@google.com
- Problems with a **lab** (locked out, etc.)
  - support@qwiklab.com
- Problems with accessing **Partner Advantage**
  - https://support.google.com/googlecloud/topic/9198654

This Week's Recommended Activities

1. Review the **exam guide** to assess your own level of expertise and readiness
2. Familiarize yourself with exam **Sample Questions**
3. Labs and Quests for this week:
   a. **Course:** Building Resilient Streaming Analytics Systems on GCP
   b. **Course:** Smart Analytics, Machine Learning, and AI on GCP
4. Labs and Quests for next week: (or review the content):
   a. **Course:** Serverless Data Processing with Dataflow: Foundations
   b. **Course:** Serverless Data Processing with Dataflow: Develop Pipelines

Week 1
Course: Google Cloud Big Data and Machine Learning Fundamentals
https://partner.cloudskillsboost.google/course_templates/3

Quest: Create and Manage Cloud Resources (this is an introductory quest) - Skill Badge
https://partner.cloudskillsboost.google/quests/120

Course: Modernizing Data Lakes & Data Warehouses with Google Cloud
https://partner.cloudskillsboost.google/course_templates/54

Week 2
Course: Building Batch Data Pipelines on Google Cloud
https://partner.cloudskillsboost.google/course_templates/53

Week 3
Course: Building Resilient Streaming Analytics Systems on GCP
https://partner.cloudskillsboost.google/course_templates/52

Course: Smart Analytics, Machine Learning, and AI on GCP
https://partner.cloudskillsboost.google/course_templates/55

Week 4
Course: Serverless Data Processing with Dataflow: Foundations

https://partner.cloudskillsboost.google/course_templates/218

Course: Serverless Data Processing with Dataflow: Develop Pipelines
https://partner.cloudskillsboost.google/course_templates/229

Week 5
Course: Serverless Data Processing with Dataflow: Operations
https://partner.cloudskillsboost.google/course_templates/264

Quest: Perform Foundational Data, ML and AI Tasks - Skill Badge
https://partner.cloudskillsboost.google/quests/117

Week 6
Lab: Optimizing BigQuery for Cost and Performance v1.5
https://partner.cloudskillsboost.google/focuses/18091?parent=catalog

Quest: Build and Optimize Data Warehouses with BigQuery - Skill Badge
https://partner.cloudskillsboost.google/quests/147

Lab: ETL Processing on Google Cloud Using Dataflow and BigQuery
https://partner.cloudskillsboost.google/focuses/11581?parent=catalog

Quest: Engineer data in Google Cloud - Skill Badge
https://partner.cloudskillsboost.google/quests/132

Week 7

Course: Preparing for the Google Cloud Professional Data Engineer Exam
https://partner.cloudskillsboost.google/course_templates/72

Practice: Professional Data Engineer Sample Questions
https://cloud.google.com/certification/practice-exam/data-engineer

Module Agenda

01 Data Transfer Services

02 Data Fusion

03 Composer

04 Cloud Monitoring and Logging

05 Tracking Billing Information

06 Quotas and Rate Limits

07 Pub/Sub, Pub/Sub Lite and Cloud Tasks

Data Transfer Services

Cloud Data Transfer Services

- Google provides a range of data transfer methods to get data into the Cloud
  - Choose based on type and volume of data
- Web console, gsutil, JSON API
  - Small amounts of data
- Storage Transfer Service
  - Bucket-to-bucket
  - Scheduled or ad hoc
- BigQuery Data Transfer Service
  - Import into BigQuery from GCS and selected Google applications
- Transfer Appliance
  - Offline import for large (20TB plus) amounts of data

gsutil: https://cloud.google.com/storage/docs/gsutil
GCS JSON API: https://cloud.google.com/storage/docs/json_api/
Storage Transfer Service: https://cloud.google.com/storage-transfer/docs/overview
BigQuery Data Transfer API: https://cloud.google.com/bigquery-transfer/docs/reference/datatransfer/rest/
Transfer Appliance: https://cloud.google.com/transfer-appliance/docs/4.0/specifications https://cloud.google.com/transfer-appliance/pricing

You have a number of choices for getting data into Google Cloud.

If you have a relatively small amount of data, you can just upload it using the CLI, the Web console, or your application code.

They also have a more automated Storage Transfer Service. With

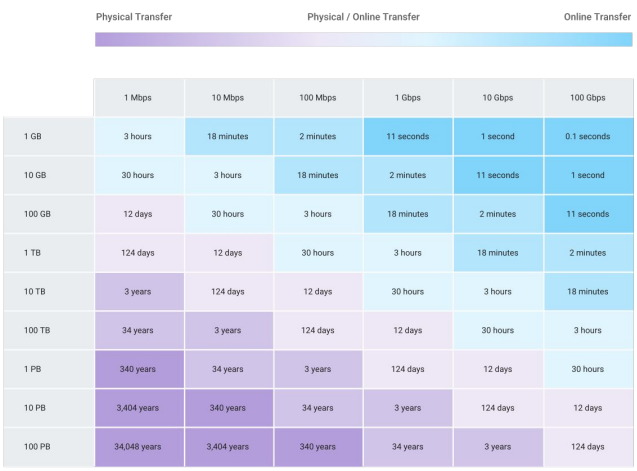
the transfer service, you specify a source and a destination. The source can be an on-premises server, an S3 bucket, or another Cloud Storage bucket. The destination is always a Cloud Storage bucket. You can then run it as a one-time job or set it to run on a schedule.

Similarly, BigQuery also has a transfer service if you prefer a BigQuery table over a storage bucket as your destination.

You can also order a Google Transfer Appliance. You connect the Transfer Appliance to your network and copy your data on it. Your data will be encrypted with a key you provide. Then you return the appliance to Google and they mount it on their network for you. You then rehydrate the data using your key. Google never has access to your key, so the data is safe.

Transfer Appliance

- If you have a lot of data, Google will send you a transfer appliance
  - Load the data locally and then ship it back
- Use when transfer times would be too long over your network



	1 Mbps	10 Mbps	100 Mbps	1 Gbps	10 Gbps	100 Gbps
1 GB	3 hours	18 minutes	2 minutes	11 seconds	1 second	0.1 seconds
10 GB	30 hours	3 hours	18 minutes	2 minutes	11 seconds	1 second
100 GB	12 days	30 hours	3 hours	18 minutes	2 minutes	11 seconds
1 TB	124 days	12 days	30 hours	3 hours	18 minutes	2 minutes
10 TB	3 years	124 days	12 days	30 hours	3 hours	18 minutes
100 TB	34 years	3 years	124 days	12 days	30 hours	3 hours
1 PB	340 years	34 years	3 years	124 days	12 days	30 hours
10 PB	3,404 years	340 years	34 years	3 years	124 days	12 days
100 PB	34,048 years	3,404 years	340 years	34 years	3 years	124 days

Use the Transfer Appliance when it would take too long to upload your data. Here's a chart showing approximately how long it would take to upload data based on its size and your bandwidth.

Note: Your data size is greater than or equal to 10TB.
See: https://cloud.google.com/transfer-appliance/docs/4.0/overview
How to Order: https://cloud.google.com/transfer-appliance/docs/4.0/order-appliance

Data Fusion

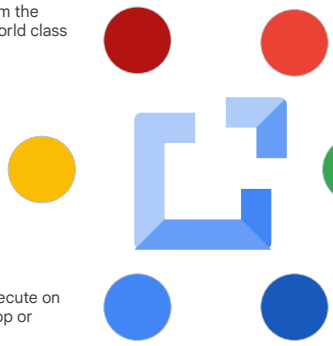
Cloud Data Fusion

Cloud Data Fusion

Key Features (open, intelligent and flexible)

Cloud-native DI
Built for Google Cloud natively from the ground up to take advantage of world class infrastructure, security, scale, and performance.

Comprehensive connectivity
Over 200+ connectors to connect to various sources ranging from databases, messaging, mainframes, social & IoT, enterprise apps, and SaaS for batch or real-time.



Hybrid and multi-cloud
Cloud Data Fusion supports both hybrid and multi cloud deployments for customers sovereignty requirements.

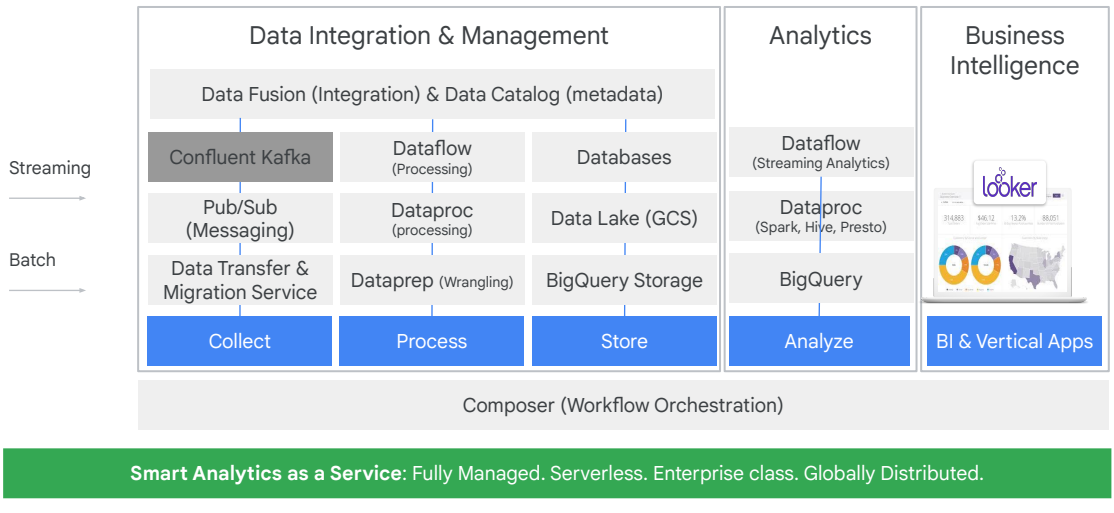
Unified wrangling and pipeline
Single platform that supports both wrangling and pipeline user experience and to easily switch between the two to improve business IT collaboration.

Pluggable runtime
Cloud Data Fusion runtime can execute on ephemeral Dataproc or any Hadoop or EMR.

Metadata driven
Cloud Data Fusion metadata driven approach helps with governance with complete insight into pipeline lineage.

Google's Smart Analytics Platform

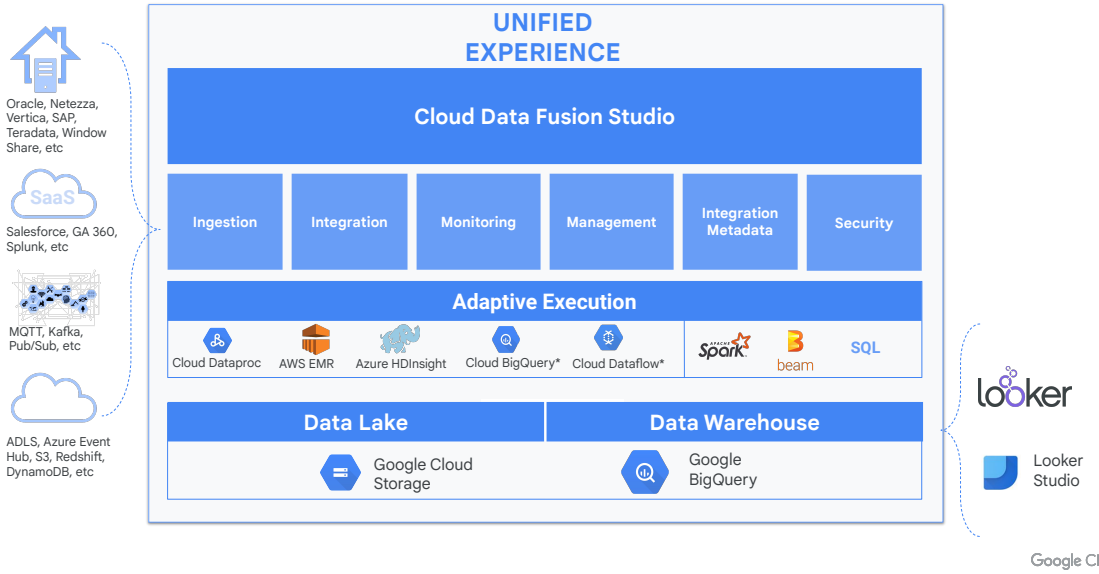
Collect, process, store, analyze and visualize data and insights



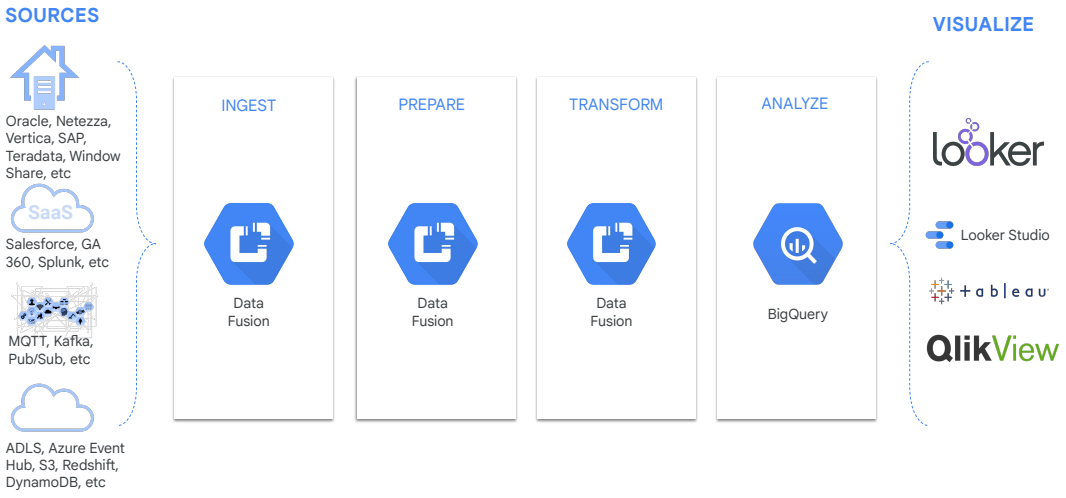
BigQuery is part of Google Cloud's comprehensive data analytics platform that covers the analytics value chain from Ingest >> process >> store >> advanced analytics and collaboration. BigQuery is deeply integrated with the GCP's analytical and data processing offering, allowing customers to build an enterprise ready cloud native data warehouse.

1. Cloud Pub/sub - Scaled messaging platform
2. DTS - Ads data for marketing cloud
3. Beam - Stream and batch processing with single programing model with Dataflow
4. Dataproc - Managed Hadoop and Spark platform
5. Dataprep - Analyst can now do data prep using visual tool
6. Data Fusion - Fully managed, code-free data integration service to manage ETL/ELT pipelines and also track lineage of that data.
7. BigQuery cloud-native, highly scalable data warehouse
8. GCS as your data lake for structured and unstructured data
9. Cloud ML Engine & Tensorflow for machine learning on top of data on BQ and GCS
10. Data studio and Sheet for your analysis

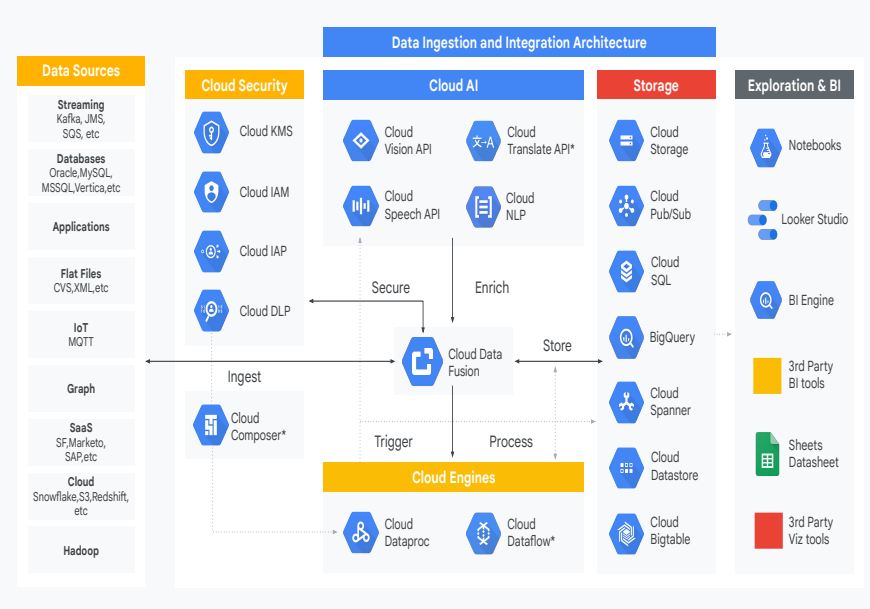
Unified pipelines with Data Fusion



Modern data warehouses with ease



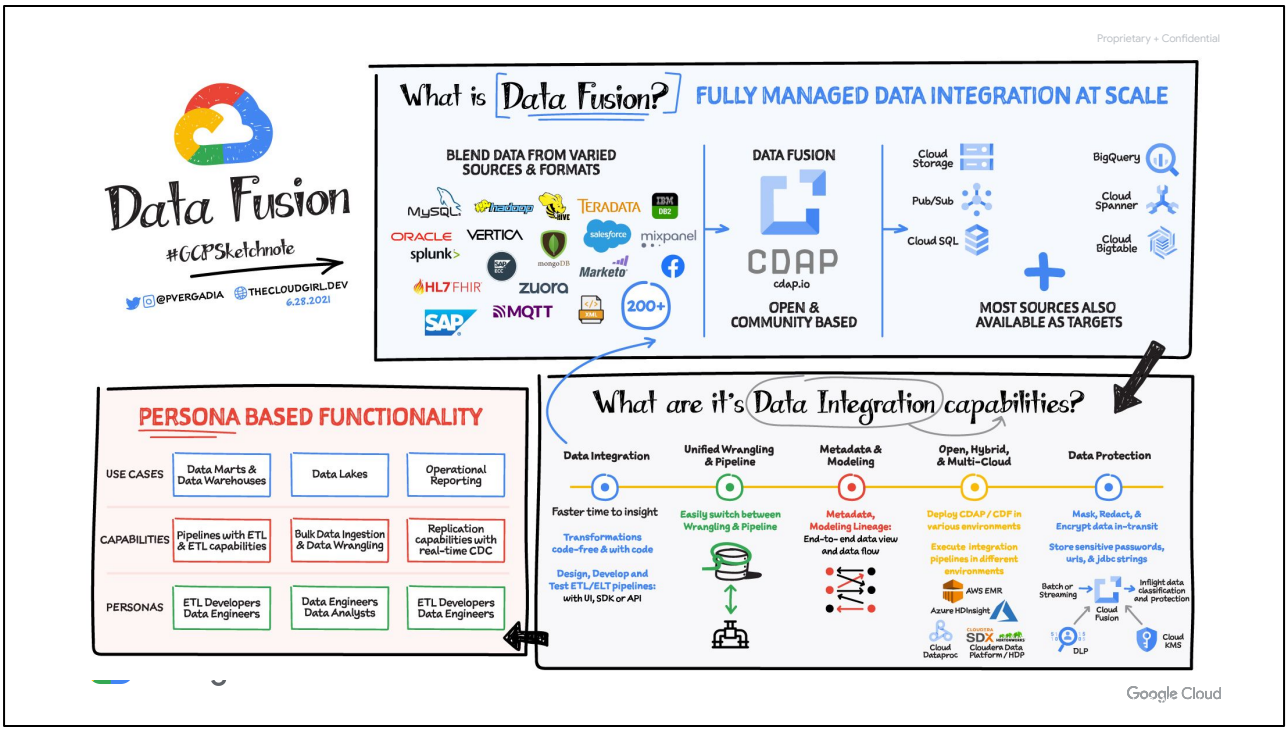
Cloud Data Fusion provides a graphical user interface and APIs that increase time efficiency and reduce complexity. It equips business users, developers and data scientists to quickly and easily build, deploy and manage data integration pipelines - transition



** Planned for a future release*

Cloud Composer vs Cloud Data Fusion

Criteria	Cloud Data Fusion	Cloud Composer
Product category	**Data Integration & Ingestion** service	**Workflow Automation and Scheduling system**
Think	**AWS Glue, AbInitio, Informatica, Talend, Dell Boomi**	**Control-M, Luigi**
Use case	**Data integration** across many to many systems including performing transformation and building complex data pipelines	Orchestrate across GCP components including Cloud Data Fusion Pipelines. Support for ETL, but not integration. Support ETL as engineering, but not as operations
Example	Extract data from on-prem Oracle and MySQL, transform and then load to BigQuery.	Provision Cloud Dataproc, Cloud Data Fusion and GCE VM, Trigger a CDF Pipeline to extract data from HTTP endpoint with static IP support.



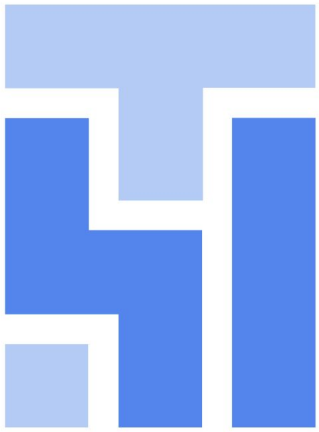
https://thecloudgirl.dev/datafusion.html

Cloud Data Fusion
https://cloud.google.com/data-fusion

- Cloud Data Fusion is a fully-managed, cloud native, enterprise data integration service for quickly building and managing data pipelines.
- It equips business users, developers and data scientists to quickly and easily build, deploy and manage data integration pipelines - transition

Composer

Cloud Composer



Cloud Composer

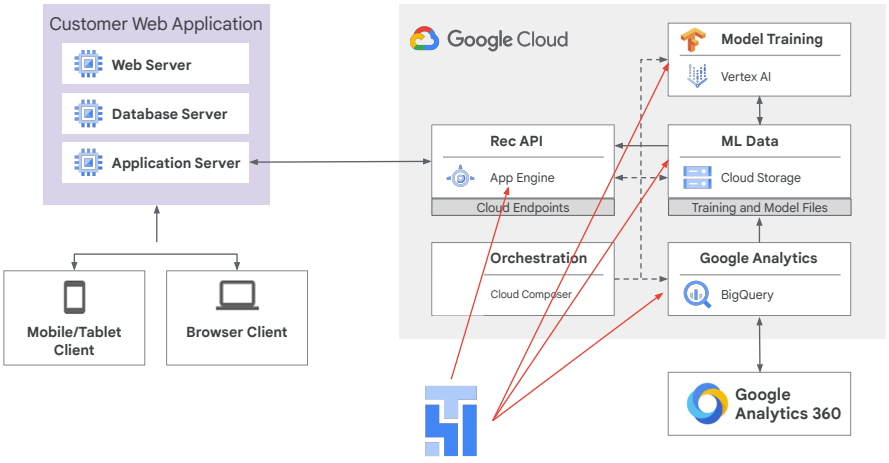
- Workflow orchestration service based on Apache Airflow
  - Can orchestrate workloads across Google Cloud, on-prem, or other clouds
  - Uses Python as orchestration language
- Built-in connectors for many Google Cloud services
  - Dataproc, Cloud MLE, GCS, Pub/Sub, BigQuery, Dataflow, etc.

Cloud Composer provides a managed orchestration service based on Apache Airflow and can be used to control the order of operations across Google Cloud, on-premise, or across other cloud providers.

Python is used as the orchestration language with Cloud Composer.

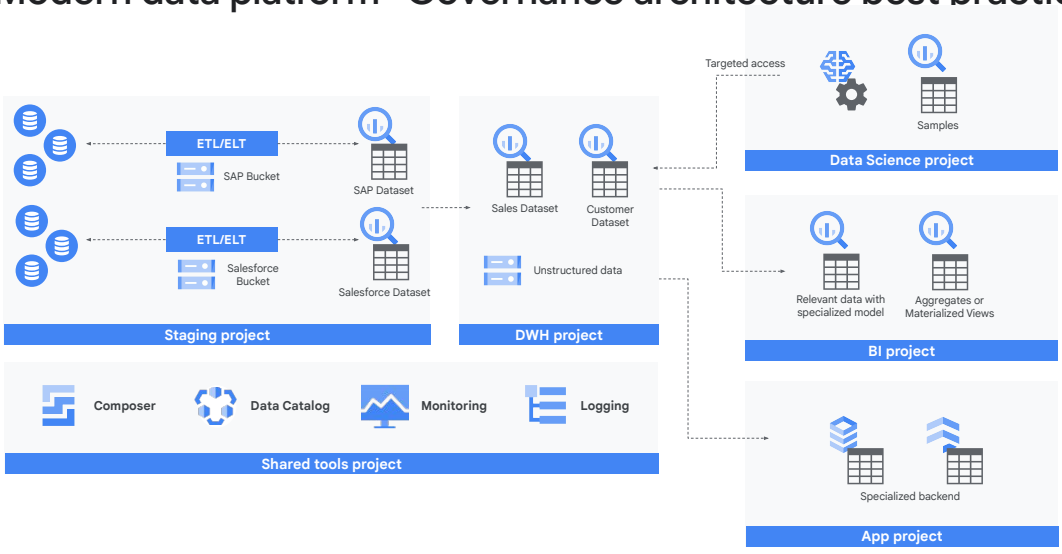
Cloud Composer has a variety of built-on connectors for Google Cloud services.

Cloud Composer orchestrates automatic workflows



Cloud Composer will command the Google Cloud services that we need to run. But Cloud Composer is simply a serverless environment on which an open source workflow tool runs.

Modern data platform- Governance architecture best practice



Cloud Composer

Cloud Composer is a fully managed data workflow orchestration service that empowers you to author, schedule, and monitor pipelines.

Pros	Cons
• Uses **open source Apache Airflow** under the hood, can define DAGs in Python and supports various operators to connect with other Google Cloud Services like DataFlow, BigQuery, etc. • Cloud Composer can aid in **pre- or post-processing** of the jobs • Offers **job lifecycle management** • Can visualize the dependencies and also monitor the running jobs using the **UI** • Infrastructure CI/CD can be managed in **isolation** with Job Orchestration	• More **operational overhead**; DevOps needs to manage both Cloud Composer, and Dataproc • Cloud Composer for simple & independent Dataproc jobs may result in an **over provisioning resources**

An example on how to use Cloud Composer with Dataproc can be found in this tutorial.
Note: Cloud Composer will create underlying peering connections and be cautious while deploying in a Shared VPC environment.

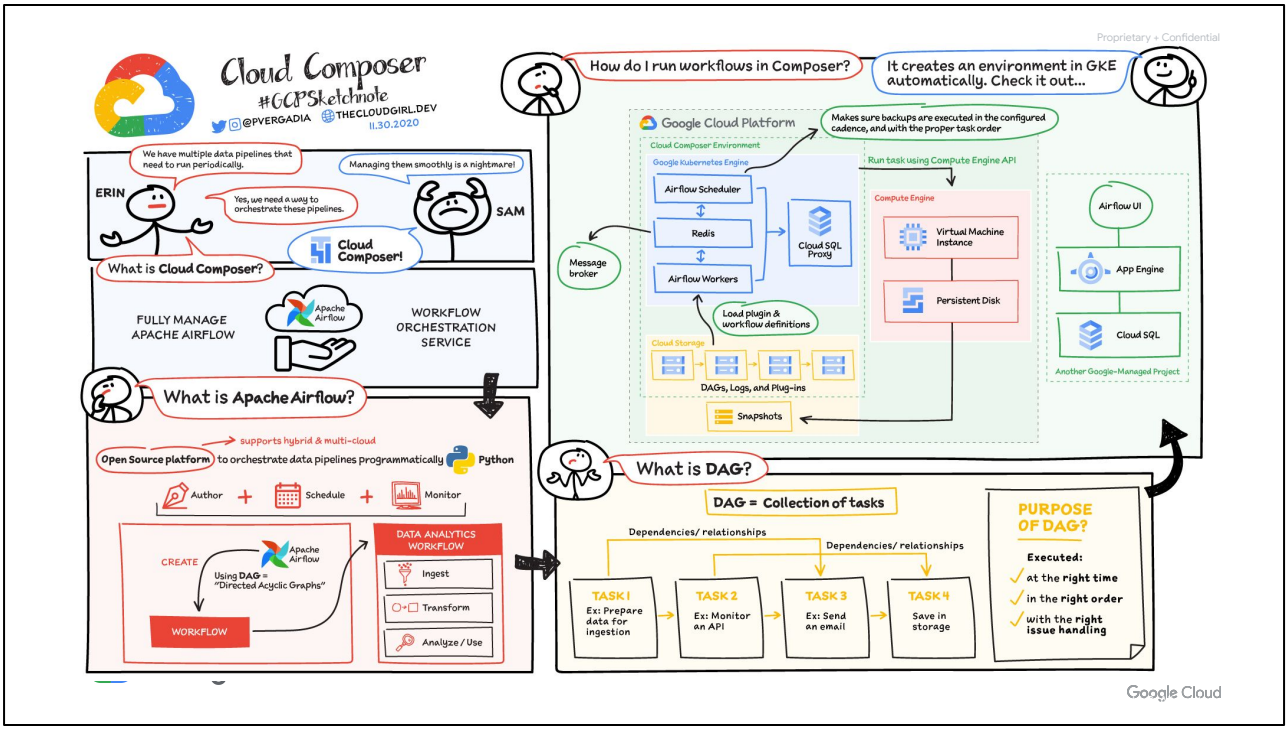
Best practices: Airflow DAG validation

- DAG Validation (test_dag_validation.py):
  - No syntax errors
  - No DAG parsing issues (Referencing undefined variables, trying to use undefined Connection IDs, DAG parsing time too long, no cycles)
- Custom Python Code behaves as expected (passes unit tests)
- References up-to-date executables.
- All your data processing code (ie. Dataflow source code, etc.) should be built in the same pipeline as your DAG deployments
  - Cloud Build should copy artifacts that were built/tested in the dev project over to the prod project, no human users should have access to do this.

Best practices: Airflow DAG deployment

- When you want to update a DAG change the dag_id
  - Delete the old DAG (These steps can be automated)
    - Pause DAG
    - Delete DAG file from GCS
    - Remove from Webserver / airflow-db
  - Deploy the new DAG (example automation)
    - Copy DAG file to GCS
    - Unpause DAG

- Updating a DAG in place can be problematic due to eventually consistent process of syncing DAGs GCS folder to the workers.



https://thecloudgirl.dev/Composer.html

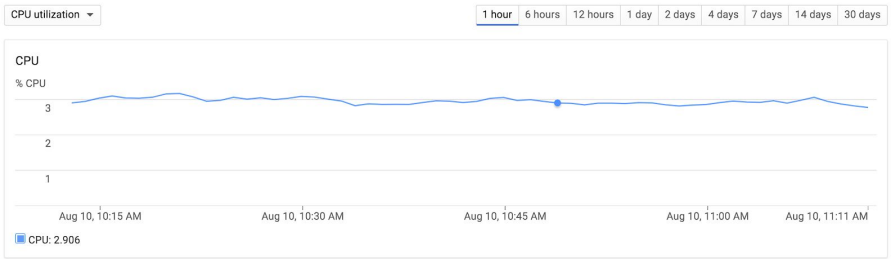
https://cloud.google.com/composer
A fully managed workflow orchestration service built on Apache Airflow.

Cloud Monitoring and Logging

Operations

Resource Monitoring

- Operations can be used to monitor many different resources
  - App Engine, disks, buckets, virtual machines, and many more
  - In addition to Google Cloud resources, AWS resources can be monitored as well
- Different metrics can be monitored depending on the resource
  - CPU usage, network traffic, uptime, message count, etc.
  - Custom metrics can be created



If you're creating virtual machines you are paying for CPUs, memory, and disk space. You should monitor those things to make sure you aren't paying for more resources than you need. This will optimize your spend.

There are lots of other things you may want to monitor. What are you spending on your App Engine application? Maybe you are allowing users to upload pictures in a Cloud Storage bucket. You might want to know how much data is in the bucket.

You're paying for network traffic out. Maybe you want to keep an eye on that and turn on the CDN if you get too much traffic. You might also want to know where your requests are coming from, and deploy your servers closer to users.

Operations has many built-in metrics that you can monitor for most every service. You can also use Operations to monitor AWS resources.

You can even create your own custom metrics programmatically. Thus, what you can monitor is really only limited by your imagination. For example, if you were an online game developer, you could create a custom metric to monitor the number of game users and the number of moves.

Monitoring Agents

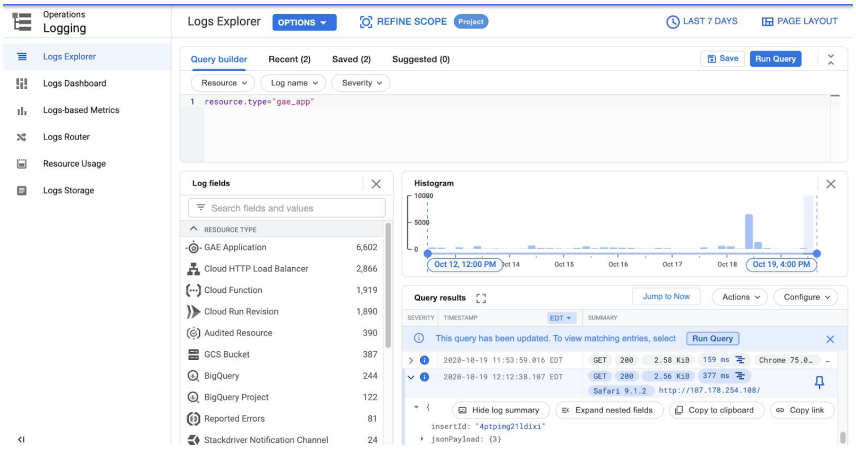
- Installed on VMs to provide additional metrics not available externally
  - Memory usage and uptime, for example
- Also provides metrics on common applications
  - Apache, Cassandra, CouchDB, HBase, IIS, JVM, Kafka, etc.
- See the docs: https://cloud.google.com/monitoring/agent/

Sometimes Operations needs a little help from inside a machine to monitor certain metrics. Memory usage might be the most common example of this. To solve this issue, you just install the Operations monitoring agent on the virtual machine you want to monitor.

The monitoring agent will also detect common applications on your machines and allow you to monitor those as well. Applications that are automatically supported include Apache Web Server, Cassandra, Internet Information Services, and many more.

See cloud.google.com/monitoring/agent for more information.

Logs



Every Google Cloud request is logged automatically. You can also write messages from your applications to the logs. Like the monitoring agent, there is a Operations logging agent you can install on virtual machines to make this easy.

You can search and filter the logging information in the UI.

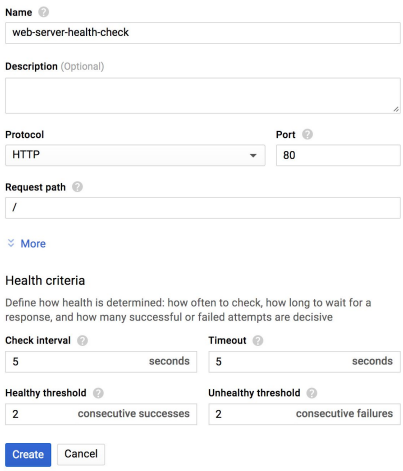
Log Exports

- Logs can be exported for further analysis
- Export destinations include:
  - BigQuery
  - Cloud Storage
  - Pub/Sub
- Can create advanced filters to specify which log entries to export
- Set a destination to Pub/Sub to immediately respond to a log export

You can also set up log exports. Log data can be exported to Cloud Storage if you want to archive it. Or, the logs can be exported to BigQuery if you want to write SQL queries to analyze the logs. You can also export log data into Pub/Sub and get real time log analysis if you like.

Health Checks

- Used to ensure machines are ready to take requests
  - Instance groups will create new machines if existing ones aren't healthy
  - Load balancers use health checks to determine if they can send requests to machines



When you set up a load balancer, you need a health check to ensure the load balancer doesn't send requests to unhealthy instances.

If you are creating an instance group, you need a health check to set up the auto healing feature.

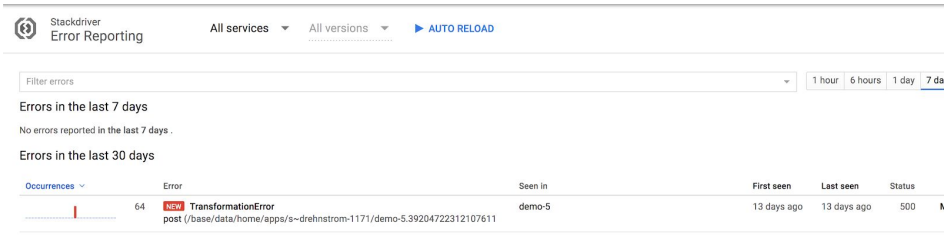
Health checks are easy. Just make a request to the service and see if it works. Some of the parameter values may not be entirely obvious though. It's possible for a single request to fail on a healthy machine. The opposite might be true too. One request might succeed once, but the machine might not really be healthy. That's what the Healthy and Unhealthy threshold values are for. You might set these values a bit higher than 1 just to be sure.

The check interval determines how often requests are made. Be aware, there is more than one server on Google's side than runs the health checks, so there are actually more health checks than what you specify in the interval.

One other thing to be aware of, if your service is not public and the health checks never seem to run, it's likely you need to create a firewall rule to open your service ports to the health checkers. See the documentation for the IP address range of the health check servers.

Error Reporting

- Automatically set up with App Engine services
  - Can enable for services running in Compute Engine
  - Can integrate with Operations logs
- Can enable automatic notification for errors



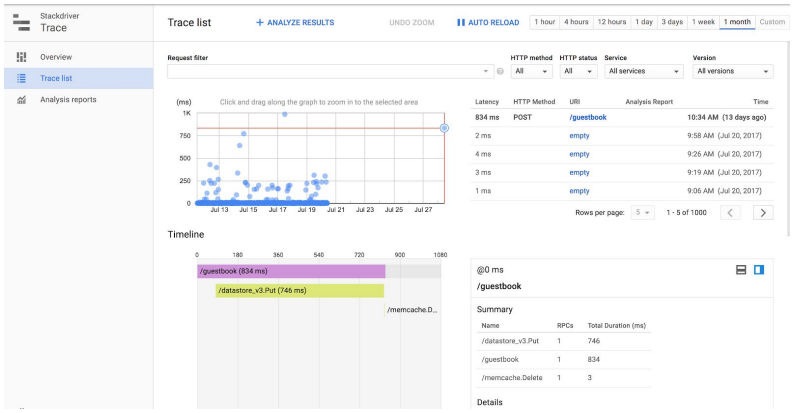
The Google SRE team developed a separate Error Reporting service to make it easier to monitor your applications for errors. This just works if you are deploying your applications using App Engine.

You can also set up notifications so you are aware if your applications start generating errors.

Trace

Displays requests along with their timings

- Useful for debugging performance problems



Google Cloud

If you are a programmer, you might be trying to debug a performance problem and sometimes this can be difficult. It may not be obvious what is specifically taking a long time within a single service request.

The Trace service makes this easier. Whenever a request is made to an App Engine application, the request is added to the Trace. The Trace breaks down exactly what happened within the request and how long each piece took.

Tracking Billing Information

Set up Cloud Billing data export to BigQuery

To export Cloud Billing data to BigQuery, you need to take the following steps:

- Create a project where the Cloud Billing data will be stored, and enable billing on the project (if you have not already done so).
- Configure permissions on the project and on the Cloud Billing account.
- Enable the BigQuery Data Transfer Service API (required to export your pricing data).
- Create a BigQuery dataset in which to store the data.
- Enable Cloud Billing export of cost data and pricing data to be written into the dataset.

Set up Cloud Billing data export to BigQuery
https://cloud.google.com/billing/docs/how-to/export-data-bigquery-setup

Billing data tables
Shortly after enabling Cloud Billing export to BigQuery, billing data tables are automatically created in the BigQuery dataset.

- Daily cost detail table – In your BigQuery dataset, this table is named `gcp_billing_export_v1_<BILLING_ACCOUNT_ID>.`
- Pricing table – In your BigQuery dataset, this table is `named cloud_pricing_export`.

Warning: Do not modify these tables, such as by adding a column or manually re-creating the tables. Doing so can cause data to stop exporting, or any data appended manually to the tables to be lost, until the changes are reverted or the tables are deleted.

If you want to consolidate billing data from past configurations or versions of Cloud Billing export to BigQuery, we recommend keeping the data in separate tables and using a UNION query instead.

Cloud Billing data tables in BigQuery

After enabling Cloud Billing export to BigQuery, billing data tables are automatically created in the BigQuery dataset:

- Daily cost detail table – In your BigQuery dataset, this table is named `gcp_billing_export_v1_<BILLING_ACCOUNT_ID>`
- Pricing table – In your BigQuery dataset, this table is named `cloud_pricing_export`

Do not modify these tables, such as by adding a column or manually re-creating the tables. Doing so can cause data to stop exporting, or any data appended manually to the tables to be lost, until the changes are reverted or the tables are deleted

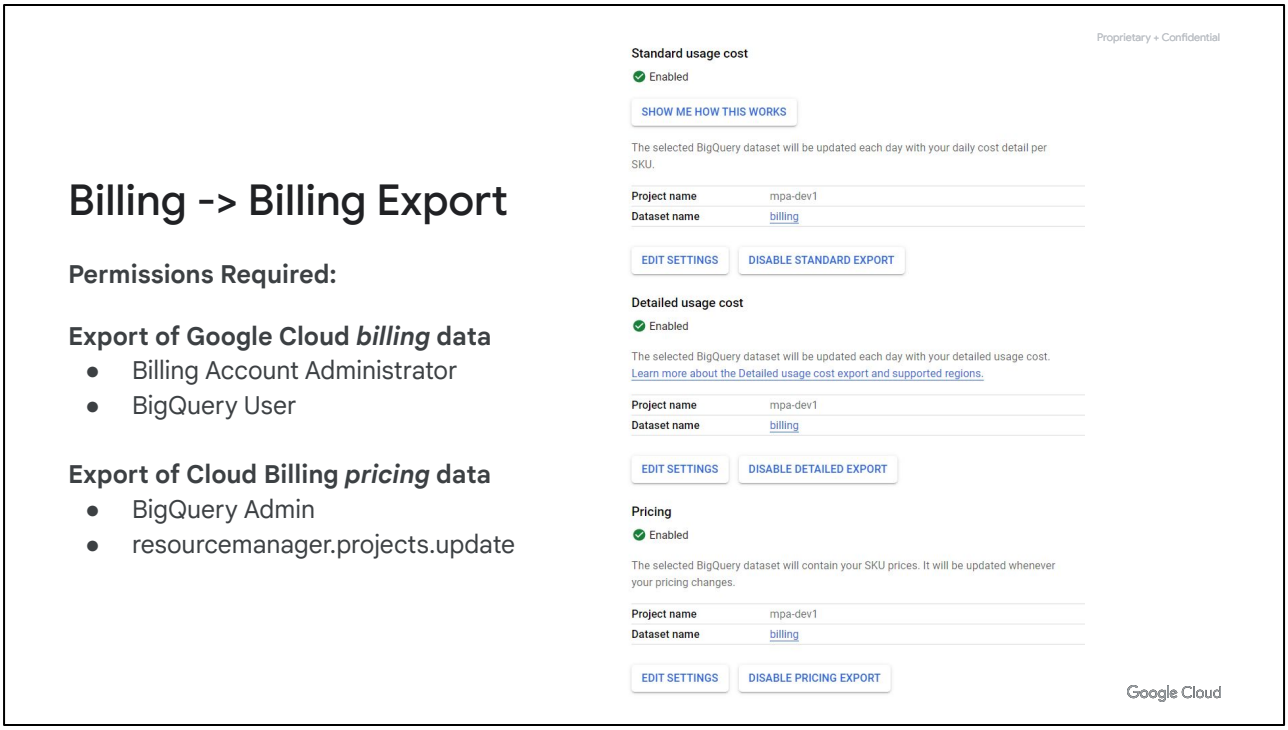
Set up Cloud Billing data export to BigQuery
https://cloud.google.com/billing/docs/how-to/export-data-bigquery-setup

Billing data tables
Shortly after enabling Cloud Billing export to BigQuery, billing data tables are automatically created in the BigQuery dataset.

- Daily cost detail table – In your BigQuery dataset, this table is named `gcp_billing_export_v1_<BILLING_ACCOUNT_ID>.`
- Pricing table – In your BigQuery dataset, this table is `named cloud_pricing_export`.

Warning: Do not modify these tables, such as by adding a column or manually re-creating the tables. Doing so can cause data to stop exporting, or any data appended manually to the tables to be lost, until the changes are reverted or the tables are deleted.

If you want to consolidate billing data from past configurations or versions of Cloud Billing export to BigQuery, we recommend keeping the data in separate tables and using a UNION query instead.



https://cloud.google.com/billing/docs/how-to/export-data-bigquery-setup#required_permissions

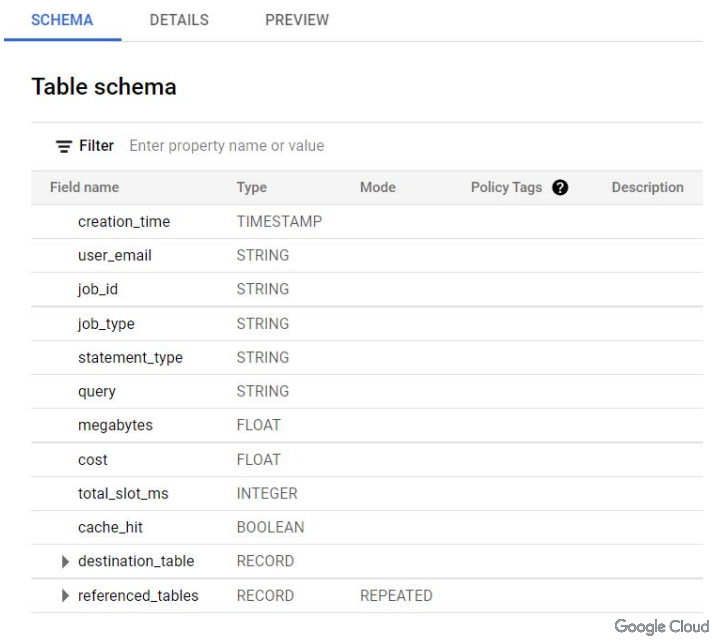
To enable and configure the export of Google Cloud **billing data** to a BigQuery dataset, you need the following permissions:

- Billing Account Administrator role for the target Cloud Billing account
- BigQuery User role for the Cloud project that contains the BigQuery dataset that will be used to store the Cloud Billing data

Additionally, to enable and configure the export of Cloud Billing **pricing data**, you need the following permissions:

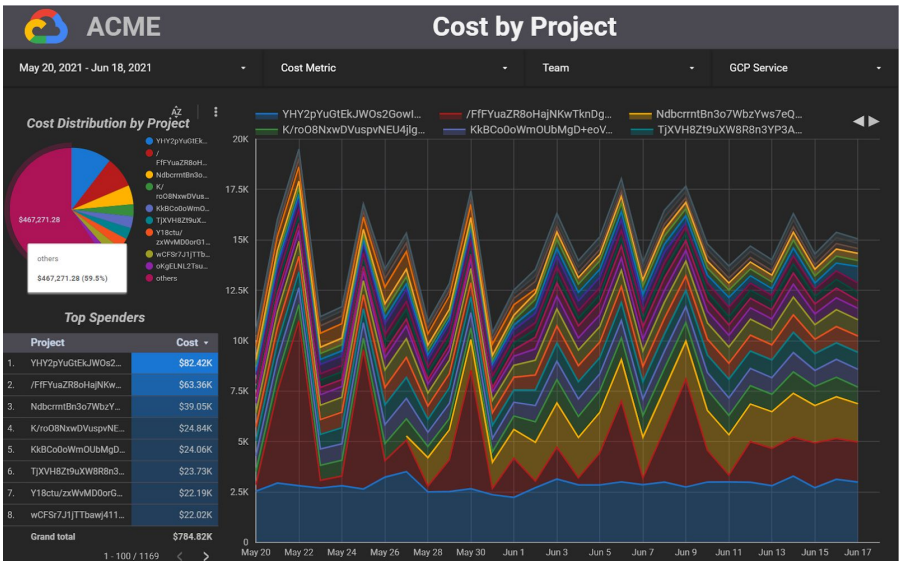
- BigQuery Admin role for the Cloud project that contains the BigQuery dataset that will be used to store the Cloud Billing pricing data
- `resourcemanager.projects.update` permission for the Cloud project containing the target dataset. This is included in the `roles/editor` role.

Billing Export Schema



https://cloud.google.com/billing/docs/how-to/export-data-bigquery-tables
https://cloud.google.com/billing/docs/how-to/bq-examples

Looker Studio Report



Google Cloud

[Getting reservations metadata using INFORMATION_SCHEMA | BigQuery (google.com)]

Everything shown on this report sample was produced by querying billing exports

Quotas and Rate Limits

Why use Quotas and Rate Limits

- Prevent runaway consumption in case of an error or malicious attack
- Prevent billing spikes or surprises
- Forces sizing consideration and periodic review
- Quotas and Rate Limits are hard ceilings
- Budgets are guidelines that do not have ceilings

https://cloud.google.com/bigquery/quotas

All resources are subject to project quotas or limits

- How many resources you can create per project
  - *5 VPC networks/project*
- How quickly you can make API requests in a project: rate limits
  - *5 admin actions/second (Cloud Spanner)*
- How many resources you can create per region
  - *24 CPUs region/project*

Increase: Quotas page in Cloud Console or a support ticket

All resources in Google Cloud are subject to project quotas or limits. These typically fall into one of the three categories shown here:

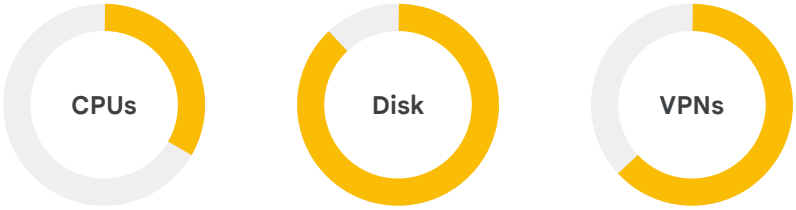
- How many resources you can create per project. For example, you can only have 5 VPC networks per project.
- How quickly you can make API requests in a project or rate limits. For example, by default, you can only make 5 administrative actions per second per project when using the Cloud Spanner API.
- There also regional quotas. For example, by default, you can only have 24 CPUs per region.

Given these quotas, you may be wondering, how do I spin up one of those 96-core VMs?

As your use of Google Cloud expands over time, your quotas may increase accordingly. If you expect a notable upcoming increase in usage, you can proactively request quota adjustments from the Quotas page in the Cloud Console. This page will also display your current quotas.

If quotas can be changed, why do they exist?

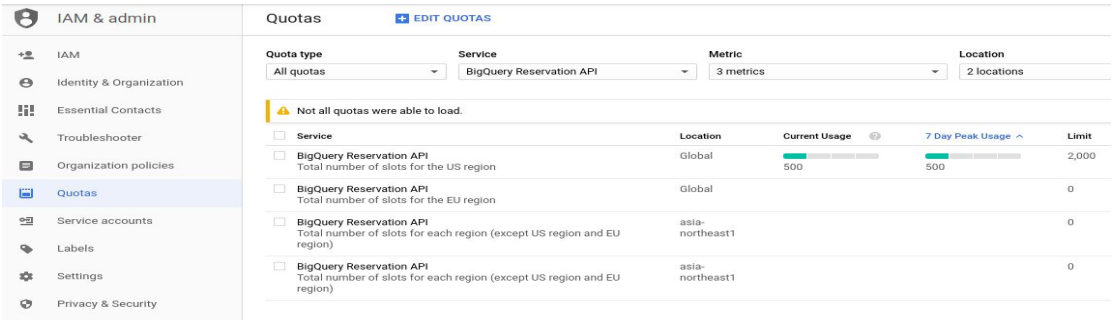
Resource quotas



- Quotas **provide protection** against
  - Cost overrun, and can be indicators of bad code
  - Other poorly behaved Google Cloud customers
- **Default quotas** may **increase as your use** of Google Cloud expands over time
- Most quotas are applied **per project**, based on **resource type and location**

Quotas - Flat Rate

Quota is the maximum allowable number of BigQuery slots you can purchase in the Cloud Console. You are not billed for quotas; you are only billed for purchased commitments. Quotas are defined per region.

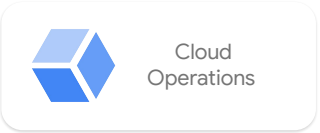


https://cloud.google.com/bigquery/pricing#flat_rate_pricing/

Monitoring quotas

Cloud Monitoring includes quota metrics for **many popular services** and can be used to configure **alerts**.

This excludes **Google Compute Engine**.



Leverage PSO-built tool (Compute Engine Quota Sync) to bridge this gap by ingesting **Google Compute Engine quota metrics** into **Cloud Operations**.

Resource quota increases

Quota increase requests are a **manual process** and are **not immediate**, but some have pre-approval.

Systematic requests

- Collaborative capacity planning strongly recommended
- Make quota increase requests well ahead of anticipated need

Emergency request

DO NOT DO THIS UNLESS CRITICAL

- Production that exceeds quota limits can be handled on an emergency basis Coordinated with account management

BigQuery Quotas and Rate Limits

Default Quotas and Rate Limits are in place for the following services:

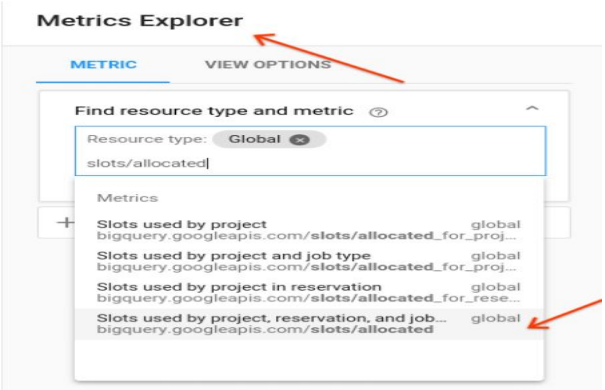
- BigQuery
- BigQuery ML
- BigQuery BI Engine
- BigQuery Data Transfer Engine

Quotas can be divided into Fixed Limit and Editable

https://cloud.google.com/bigquery/quotas

Monitoring Flat Rate

Information is available from the “slots allocated” metric. Gives a breakdown per project, per job breakdown.



Creating Custom Cost Controls

- It is not possible to assign a custom quota to a specific user or service account.
- Custom quotas are approximate. The custom quotas feature provides an additional safeguard against excessive spending, but is not designed to strictly limit bytes processed. BigQuery might occasionally run a query that exceeds a quota.
- Custom quotas are not enabled by default.

https://cloud.google.com/bigquery/docs/custom-quotas

Custom Quota Enforcement - Project

If you exceed a project-level custom quota, BigQuery returns the following usageQuotaExceeded error:

```
Custom quota exceeded: Your usage exceeded the custom quota for
QueryUsagePerDay, which is set by your administrator. For more information,
see https://cloud.google.com/bigquery/cost-controls
```

Custom Quota Enforcement - User

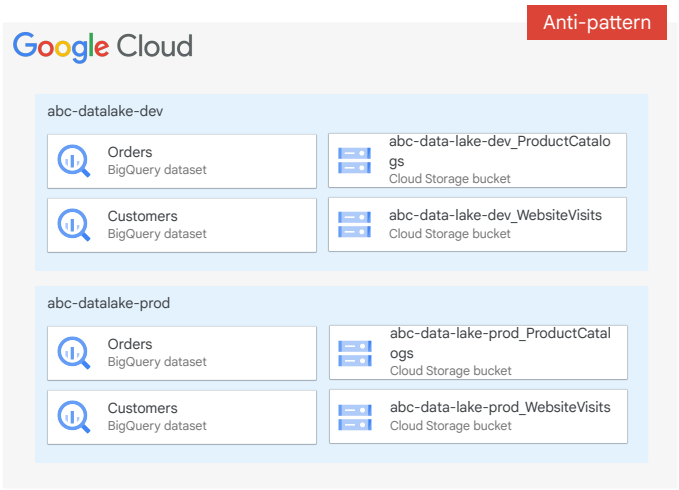
If you exceed a project-level custom quota, BigQuery returns the following usageQuotaExceeded error:

```
Custom quota exceeded: Your usage exceeded the custom quota for
QueryUsagePerUserPerDay, which is set by your administrator. For more
information, see https://cloud.google.com/bigquery/cost-controls
```

Example: Using a single project for each environment

Considerations for solution

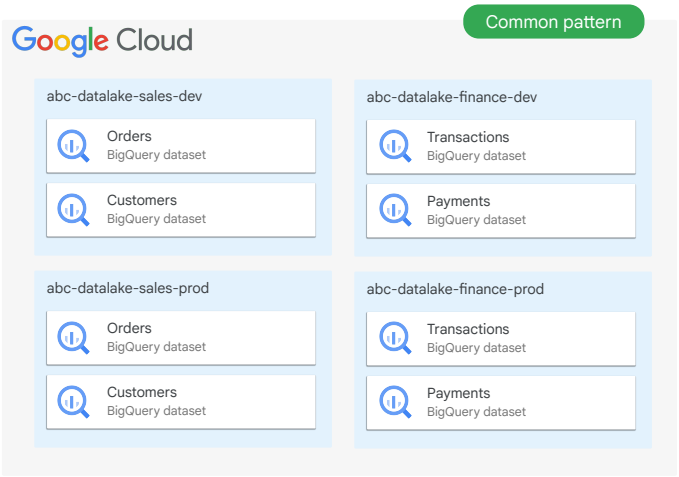
- **Billing isolation**: Possibilities for different billing accounts, environment-level cost visibility
- **Quotas and limits**: Separate by environment, however shared across workloads
- **Administrative complexity**: Access management done on environment level, shared by workloads
- **Blast radius**: Misconfiguration issues impacts a single environment and all of its workloads
- **Separation of duties**: Business units and data sensitivity classification mixed



Example: One project for each application and environment

Considerations for solution

- **Billing isolation**: Possibilities for different billing accounts, easy cost visibility per workload and environment
- **Quotas and limits**: Separate by environment and workloads
- **Administrative complexity**: Access management separate by environment and workload, with additional management overhead
- **Blast radius**: Misconfiguration issues only impact workloads specific environments
- **Separation of duties**: Business units and data sensitivity classification are separate



Q&A

When is my custom quota refilled?

- Daily quotas reset at midnight Pacific Time.

Is custom quota proactive or reactive?

- Custom quota is proactive, so you won't be able to run an 11 TB query if you have a 10 TB quota.

Can custom quotas span multiple projects?

- No, custom quotas are project specific.

https://cloud.google.com/bigquery/quotas

Pub/Sub, Pub/Sub Lite and Cloud Tasks

Pub/Sub

Pub/Sub is an asynchronous messaging service that decouples services that produce events from services that process events.

- Fully Managed, No-Ops real-time messaging service
- Reliable messaging and streaming data
- Deep integration with other GCP products
- Low latency even at high scale

Core Concepts
- Topic
- Subscription
- Message
- Message attribute

https://cloud.google.com/pubsub/docs/overview

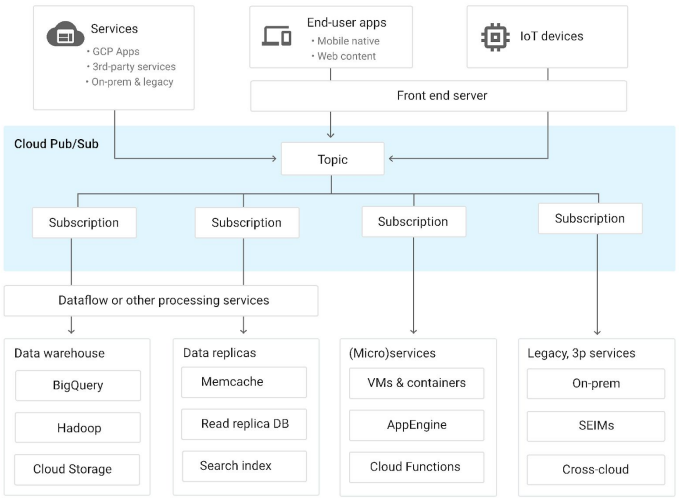
Core Concepts
https://cloud.google.com/pubsub/docs/overview#data_model

- **Topic**: A named resource to which messages are sent by publishers.
- **Subscription**: A named resource representing the stream of messages from a single, specific topic, to be delivered to the subscribing application. For more details about subscriptions and message delivery semantics, see the Subscriber Guide.
- **Message**: The combination of data and (optional) attributes that a publisher sends to a topic and is eventually delivered to subscribers.
- **Message attribute**: A key-value pair that a publisher can define for a message. For example, key iana.org/language_tag and value en could be added to messages to mark them as readable by an English-speaking subscriber.

Other items to mention:
- Global, fully managed noops/serverless
- “Shock absorber”
- “Glue” between services

- It considers the message as a “bag of bytes” as opposed to other products/services will inspect your data
    - Can send send a jpeg or even break it up and on the other side another service to assemble it back together
    - So it doesn’t care about the content of your message
    - But will care about the attributes you apply to your message
- Push vs Pull
    - Pull is the default
    - Push is faster than pull

Integrating Pub/Sub with Other Services



Here is an example of Pub/Sub working with other services.

The middle portion of the diagram, shaded in blue, is the Cloud Pub/Sub service.

Above Pub/Sub, are the producers of the message.

Below Pub/Sub, are the subscribers.

A producer will deliver a message to Pub/Sub. A producer can be a Google Cloud service, an end user application, or an IoT device.

The messages are delivered to Pub/Sub and are are stored in Topics. Subscriptions are created to assist in delivering the messages.

The subscribers get their messages from the subscription. Subscribers can be Google Cloud services, storage, or applications.

Topics and Subscriptions

- Messages in Pub/Sub are sent to a Topic
  - Messages can contain data and attributes
- Topics are named endpoints where messages are sent
  - Topic names are in the form: projects/<project-id>/topics/<topic-name>
- Subscriptions represent a stream of messages within a topic
  - Topics can contain multiple subscriptions
  - Each subscription belongs to one topic
  - Subscribers get messages from subscriptions

In Cloud Pub/Sub, messages are delivered using Topics and Subscriptions.

Messages are sent to a Topic. The messages contain data and attributes.

Topics are named endpoints. Topic names have a format: `Projects/<ProjectID>/Topics/<Topic-name>`

Subscripts is a stream of messages within a topic.

Topics can have multiple subscriptions. A subscription belongs to one topic. Subscribers get messages based on the subscriptions they are subscribed to.

Subscribers

- Subscribers are applications that process Pub/Sub messages
  - Subscribers get messages from a subscription
- Two types of subscriptions, push and pull
- Push messages are automatically sent to the subscriber via an endpoint
  - Acknowledgement of the message is implied by a response code 200
- Pull messages must be requested by the subscriber
  - Subscriber calls the pull() method of the Pub/Sub API
  - If a message exists, it is sent
  - Subscriber then calls the acknowledge() method

Subscribers are applications that process Pub/Sub messages. Subscribers get messages from a subscription

There are two types of subscriptions, push and pull.

Push messages are automatically sent to subscriber via an endpoint. Acknowledgement of the messages is implied by a response code 200

Pull messages must be requested by the subscriber

Subscriber calls the pull() method of the pubsub API. If the message exists, it is sent. The subscriber then calls the acknowledge() method.

If an acknowledgement is not sent, then the message will stay in

the queue and you run the risk of the message being sent again.

Push or Pull Subscribers

- Push subscribers must be web servers that support HTTPS
  - Must expose an endpoint to receive the message (a webhook)
  - Delivery is immediate unless throttled
  - Can be load balanced
- Pull subscribers can be any type of application
  - Must be able to use the Pub/Sub REST API
- App Engine applications are ideal push subscribers
- Dataflow jobs are pull subscribers

A few other things about push or pull subscribers.

Push subscribers must be web servers that support HTTPS. They have to expose an endpoint to receive the messages, which is called a webhook. Delivery can be immediate but it can also be throttled. The messages can be load balanced.

Pull subscribers can be any type of application. You have to be able to use the Pub/Sub rest-based API for your pool subscribers.

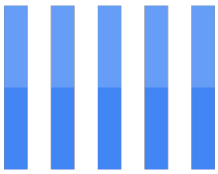
App Engine applications are designed to work with Pub/Sub and make great push subscribers.

Dataflow jobs are pull subscribers. As the messages come in, Dataflow will poll the que, look for a message, and deliver the message to the pipeline.

Pub/Sub Lite

Pub/Sub

Cloud Tasks



Pub/Sub noteworthy resource limits

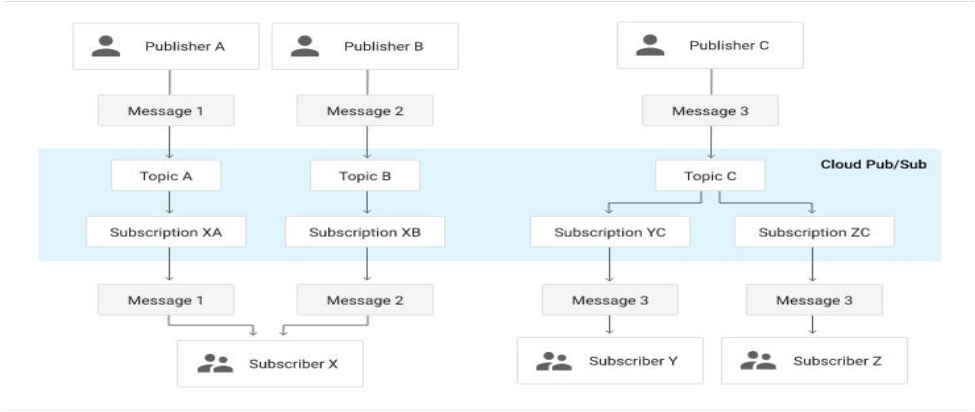
- **Project**
  - 10,000 topics, 10,000 attached or detached subscriptions
  - 5,000 snapshots
- **Topic**
  - 10,000 attached subscriptions
  - 5,000 attached snapshots
- **Message**
  - 10MB - message size (data field)
  - 100 attributes per message
- **Subscription**
  - Unacknowledged messages persistent for 7 days
- **Publish requests**
  - 10MB (total size)

Resource Limits
https://cloud.google.com/pubsub/quotas#resource_limits

At this time these can not increase these.

Publisher-Subscriber Relationships

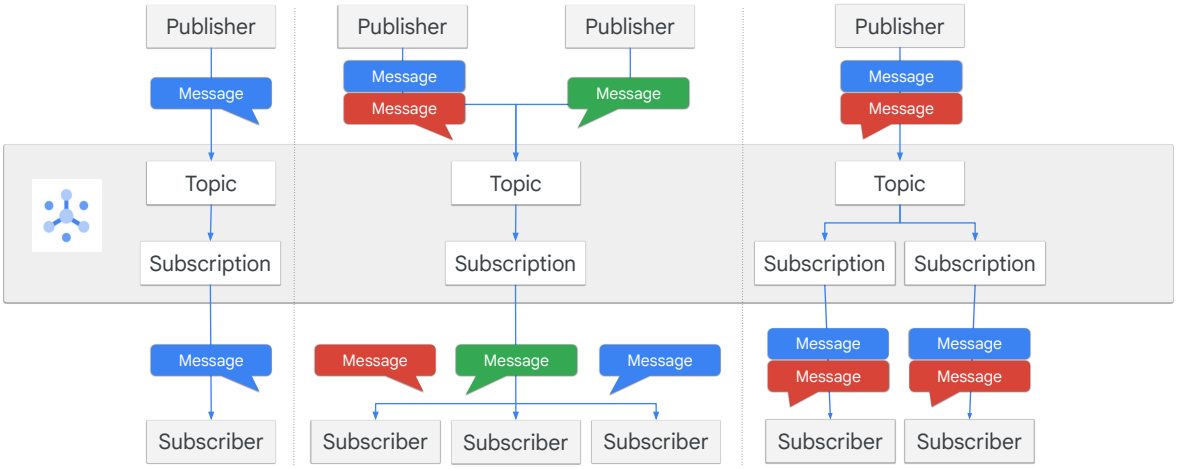
A publisher application creates and sends messages to a *topic*. Subscriber applications create a *subscription* to a topic to receive messages from it. Communication can be one-to-many (fan-out), many-to-one (fan-in), and many-to-many, as the following diagram shows.



Publisher-Subscriber Relationships
https://cloud.google.com/pubsub/docs/overview#publisher-subscriber-relationships

- Publisher can be outside or inside of Google Cloud

Publisher-Subscriber patterns



Basic pattern just a straight through, which is a queue.
To receive messages published to a topic, you must create a subscription to that topic. Only messages published to the topic after the subscription is created are available to subscriber applications. The subscription connects the topic to a subscriber application that receives and processes messages published to the topic

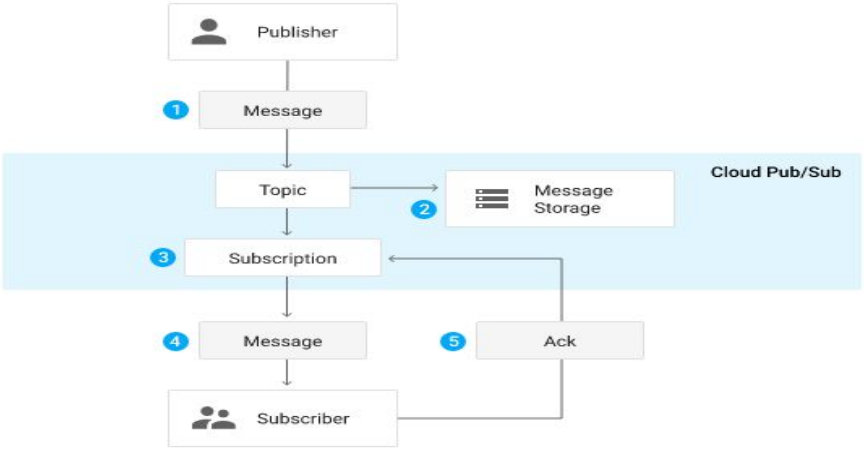
(Fan in / Load Balancing) Multiple Publishers publishing to the same topic
Also have Multiple subscribers pulling from same subscriptions, basic parallelization of processing. dataflow,
one subscription multiple consumers and each Subscriber receives a subset of messages from the subscription.

(Fan out) Multiple subscribers, where you have multiple use case for same piece of data, and all data is sent to multiple different subscribers.

By providing many-to-many, asynchronous messaging that decouples senders and receivers, it allows for secure and highly available

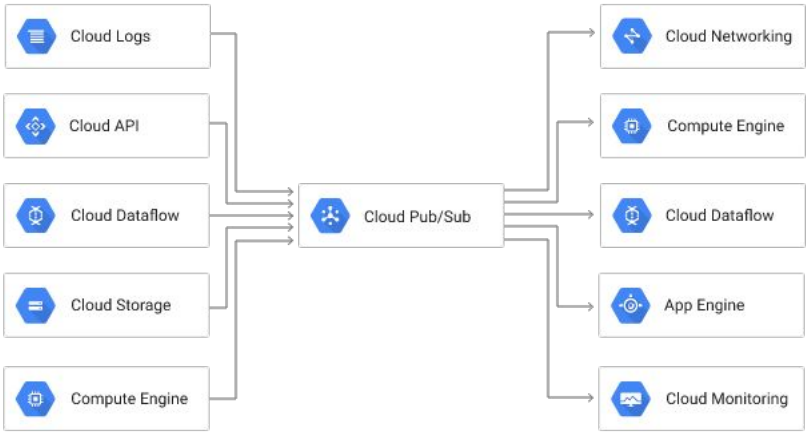
communication among independently written applications

Pub/Sub Message Flow



https://cloud.google.com/pubsub/docs/overview#concepts

Pub/Sub Integrations



Pub/Sub Integrations
https://cloud.google.com/pubsub/docs/overview#pubsub-integrations

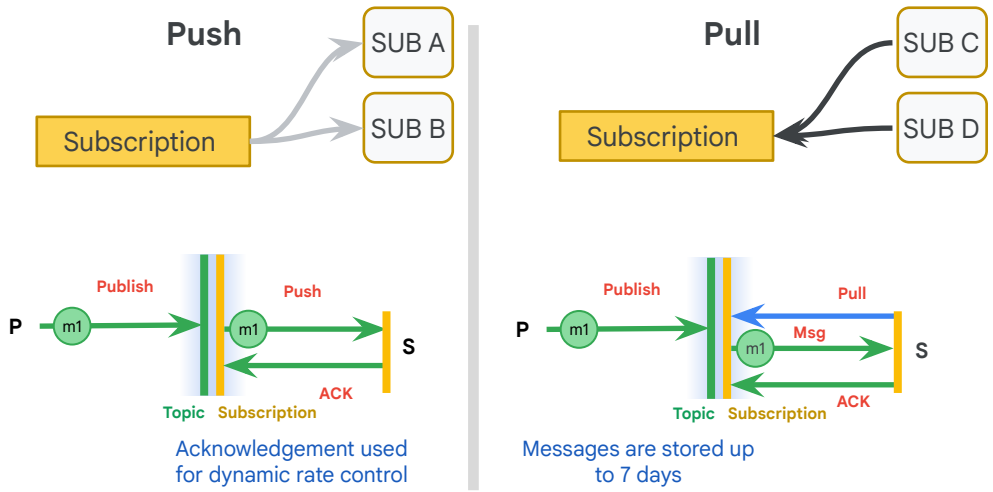
It's a well-positioned service that can be integrated with many Google Cloud Services

- Has a REST API programming model
- Can write your own pub/sub integration if needed
- Security via IAM role-based security across these services can drastically improve security, management and visibility

Other mentions

- "Shock absorber"
- "Glue" between services

Cloud Pub/Sub provides both Push and Pull delivery

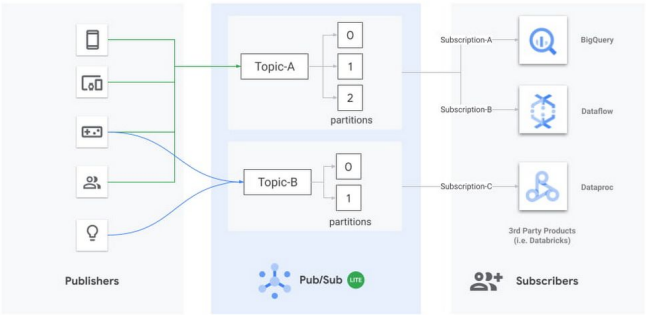


In the Push model, it actually uses an HTTP endpoint. You register an webhook as your subscription, and Pub/Sub infrastructure itself will call you with the latest messages. In the case of Push, you just respond with ‘status 200 ok’ for the HTTP call, and that tells Pub/Sub the message delivery was successful.

It will actually use the rate of your success responses to self limit so that it doesn’t overload your worker.

Pub/Sub Lite

- A zonal service
- Run publisher, subscriber and topics in the same zone
- Designed to minimize networking egress cost and latency
- High speed replacement for Kafka and Spark structured streaming
- GA Oct 9, 2020



Check out Pub/Sub Lite for your streaming applications | Google Cloud Blog (Mar 5, 2021)
https://cloud.google.com/blog/products/data-analytics/pubsub-lite-for-your-streaming-applications

What is Pub/Sub Lite?
Pub/Sub Lite is a recently released, horizontally scalable messaging service that lets you send and receive messages asynchronously between independent applications. Publisher applications publish messages to a Pub/Sub Lite topic, and subscriber applications (like Apache Spark) read the messages from the topic.

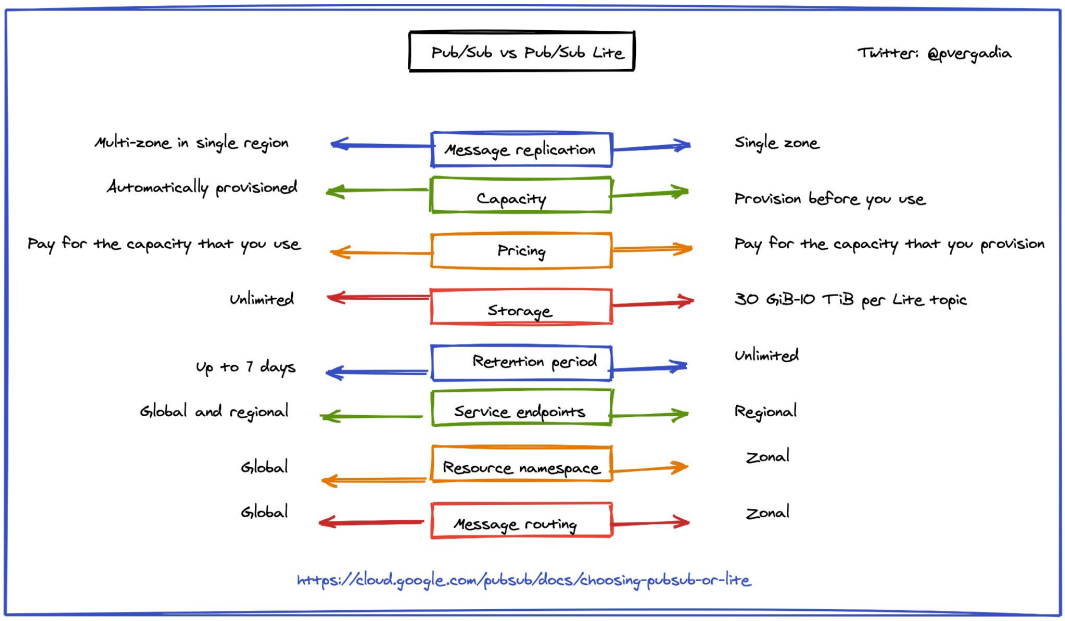
Pub/Sub Lite is a zonal service. While you can connect to Pub/Sub Lite from anywhere on the internet, running publisher and subscriber applications in the same zone as the topic they connect to will help minimize networking egress cost and latency.

GA Oct 9, 2020
https://cloud.google.com/pubsub/docs/release-notes#October_09_2020

Why did Pub/Sub Lite come about?

- Kafka is a regional or zonal solution
  - This on-prem
  - Or Co-lo
- Think of kafka as “zonal” b/c it’s located somewhere for a customer

- Think of Pub/Sub Lite as a head-to-head competitor for Kafka to meet “zonal” and charge model of what customers are used to
    - Can help in migration scenarios and the like
    - Less costly
        - egress is zonal → focused area for services etc ...



https://thecloudgirl.dev/gcpsketchnote4.html

Pub/Sub vs Pub/Sub Lite

Feature	Pub/Sub	Pub/Sub Lite
Message replication	Multi-zone in single region	Single zone
Capacity	Automatically provisioned	Provision before you use
Pricing	Pay for the capacity that you use	Pay for the capacity that you provision
Storage	Unlimited	30 GiB-10 TiB per Lite topic
Retention period	Up to 7 days	Unlimited
Service endpoints	Global and regional	Regional
Resource namespace	Global	Zonal
Message routing	Global	Zonal

[Choosing Pub/Sub or Pub/Sub Lite | Cloud Pub/Sub Documentation (google.com)]

Pub/Sub and Pub/Sub Lite are both horizontally scalable, managed messaging services.

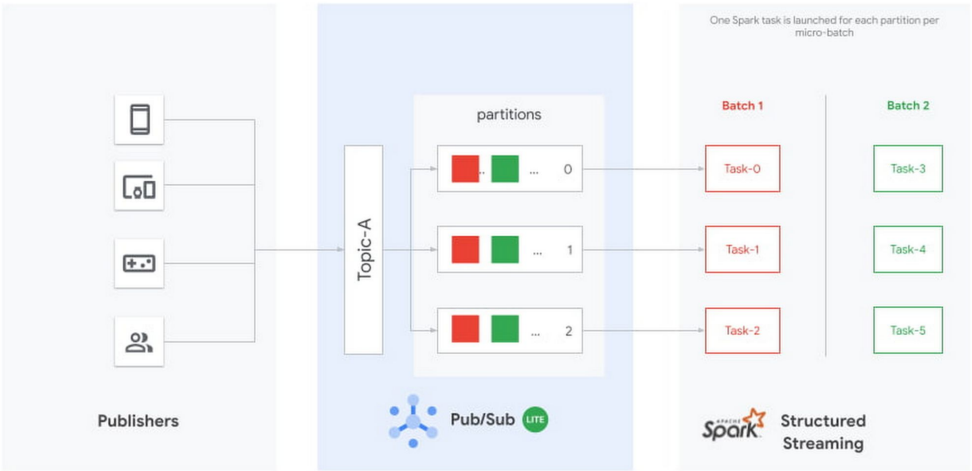
- Pub/Sub should be the default solution for most application integration and analytics use cases.
- Pub/Sub Lite is only recommended for applications where achieving extremely low cost justifies some additional operational work.

Pub/Sub offers a broader range of features, per-message parallelism, global routing, and automatically scaling resource capacity.

Pub/Sub Lite can be as much as an order of magnitude less expensive, but offers lower availability and durability. In addition, Pub/Sub Lite requires you to manually reserve and manage resource capacity.

Pricing
https://cloud.google.com/pubsub/pricing#pubsub-pricing
https://cloud.google.com/pubsub/pricing#comparing_pricing

Spark Structured Streaming connector for Pub/Sub Lite



Introducing Apache Spark Structured Streaming connector for Pub/Sub Lite (Mar 5, 2021)
https://cloud.google.com/blog/products/data-analytics/pubsub-lite-for-your-streaming-applications

Three-tier architecture showing Publishers writing to Pub/Sub Lite's Topic-A, which contains three partitions that are read by Spark's Structured Streaming.

Today we're excited to announce the release of an open source connector to read streams of messages from Pub/Sub Lite into Apache Spark. Pub/Sub Lite is a scalable, managed messaging service for Spark users on GCP who are looking for an exceptionally low-cost ingestion solution. The connector allows you to use Pub/Sub Lite as a replayable source for Structured Streaming's processing engine with exactly-once guarantees1 and ~100ms processing latencies.

The connector works in all Apache Spark 2.4.X distributions, including Dataproc, Databricks, or manual Spark installations.

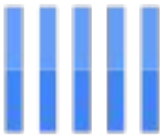
Pub/Sub Lite is only a part of a stream processing system. While Pub/Sub Lite solves the problem of message ingestion and delivery, you'll still need a message processing component.

Apache Spark is a popular processing framework that's commonly used as a batch processing system.

Streaming processing was introduced in Spark 2.0 using a micro-batch engine. The Spark micro-batch engine processes data streams as small batch jobs that periodically read new data from the streaming source, then run a query or computation on it. The time period for each micro-batch can be configured via triggers to run at fixed intervals. The number of tasks in each Spark job will be equal to the number of partitions in the subscribed Pub/Sub Lite topic. Each Spark task will read the new data from one Pub/Sub Lite partition, and together create a streaming DataFrame or Dataset.

Cloud Tasks

- Asynchronous task execution - a fully managed service
- Task Deduplication
- Guaranteed Delivery
- Schedule when a task is run
- Distributed Task Queues
- Decouple and scale micro services
- Manage resource utilization
- HTTP Targets in GCP or On-Premise
  - Cloud Functions
  - Cloud Run
  - GKE
  - Compute Engine
  - On-Premise web server



Cloud Tasks overview | Cloud Tasks Documentation | Google Cloud
Cloud Tasks Service For Asynchronous Execution | Google Cloud

https://cloud.google.com/tasks/docs
Cloud Tasks is a fully managed service that allows you to manage the execution, dispatch and delivery of a large number of distributed tasks. You can asynchronously perform work outside of a user request. Your tasks can be executed on App Engine or any arbitrary HTTP endpoint.

So not just Google Cloud, can be anything that has an HTTP endpoint

It falls under Developer Tools

GA Apr 9, 2019

- https://cloud.google.com/tasks/docs/release-notes#April_09_2019
- Initially for AppEngine

Cloud Tasks with HTTP Queues

Cloud Task Service forwards request to any generic HTTP target
Target must manage scaling workers and cleaning up tasks once they are complete

[Cloud Tasks overview | Cloud Tasks Documentation | Google Cloud]
[Cloud Tasks Service For Asynchronous Execution | Google Cloud]

Cloud Tasks with App Engine Targets

Cloud Task Service forwards request to the worker
Cloud Task Service can handle much of the process management for the task, scaling workers and deleting completed tasks.

[Cloud Tasks overview | Cloud Tasks Documentation | Google Cloud]
[Cloud Tasks Service For Asynchronous Execution | Google Cloud]

Pub/Sub vs Cloud Tasks

Pub/Sub

- Publishers do not need to know anything about their subscribers
- Publishers have no control over the delivery of the messages
- Referred to as **implicit** invocation

Cloud Tasks

- Publisher retains full control of execution
- Publisher specifies an endpoint where each message is to be delivered
- Referred to as **explicit** invocation

Choosing between Cloud Tasks and Pub/Sub | Cloud Tasks Documentation (google.com)

Both Cloud Tasks and Pub/Sub may be used to implement message passing and asynchronous integration, but while they function in similar ways, they are not identical. This page helps you choose the right product for your use case.

Key Differences
The core difference between Pub/Sub and Cloud Tasks is the notion of implicit vs explicit invocation.

Pub/Sub aims to **decouple** publishers of events and subscribers to those events. Publishers do not need to know anything about their subscribers. As a result, Pub/Sub gives publishers no control over the delivery of the messages ~~save for~~ just the guarantee of delivery. In this way, Pub/Sub supports implicit invocation: a publisher implicitly causes the subscribers to execute by publishing an event.

By contrast, Cloud Tasks is aimed at explicit invocation where the publisher retains full control of execution. In particular, a publisher specifies an endpoint where each message is to be delivered.

In addition, Cloud Tasks provides tools for queue and task management unavailable to Pub/Sub publishers, including:

- Scheduling specific delivery times

- Delivery rate controls
- Configurable retries
- Access and management of individual tasks in a queue
- Task/message creation deduplication

Cloud Tasks queue & task management

- Scheduling specific delivery times
- Delivery rate controls
- Configurable retries
- Access and management of individual tasks in a queue
- Task/message creation deduplication

In addition, Cloud Tasks provides tools for queue and task management unavailable to Pub/Sub publishers, including:

- Scheduling specific delivery times
- Delivery rate controls
- Configurable retries
- Access and management of individual tasks in a queue
- Task/message creation deduplication

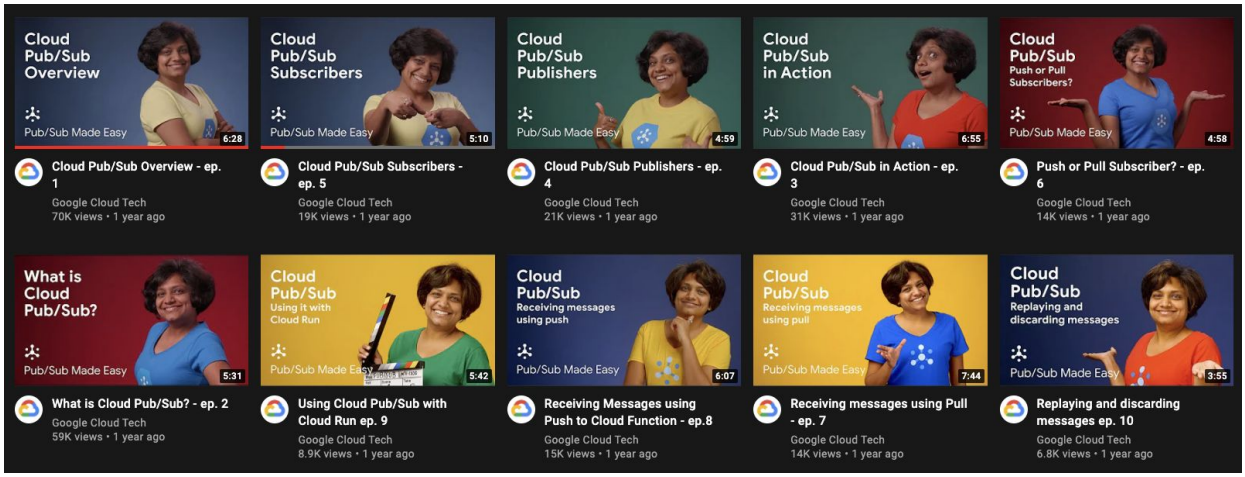
Pub/Sub vs Cloud Tasks

Feature	Cloud Tasks	Cloud Pub/Sub
Push via webhooks	**Yes**	**Yes**
At least once delivery guarantee	**Yes**	**Yes**
Task creation deduplication	**Yes**	No
Configurable retries	**Yes**	No
Scheduled delivery	**Yes**	No
Explicit rate controls	**Yes**	No (Subscriber clients can implement flow control)
Pull via API	No	**Yes**
Batch insert	No	**Yes**
Multiple handlers/subscribers per message	No	**Yes**
Task/message retention	30 days	Up to 7 days
Max size of task/message	1MB	10MB
Max delivery rate	500 qps/queue	No upper limit
Geographic availability	Regional	Global
Maximum push handler/subscriber processing duration	30 minutes (HTTP) 10 minutes (App Engine Standard automatic scaling) 24 hours (App Engine Standard manual or basic scaling) 60 minutes (App Engine Flexible)	10 minutes for push operations
Number of queues/subscriptions per project	1,000/project, more available via quota increase request	10,000/project

Google Cloud

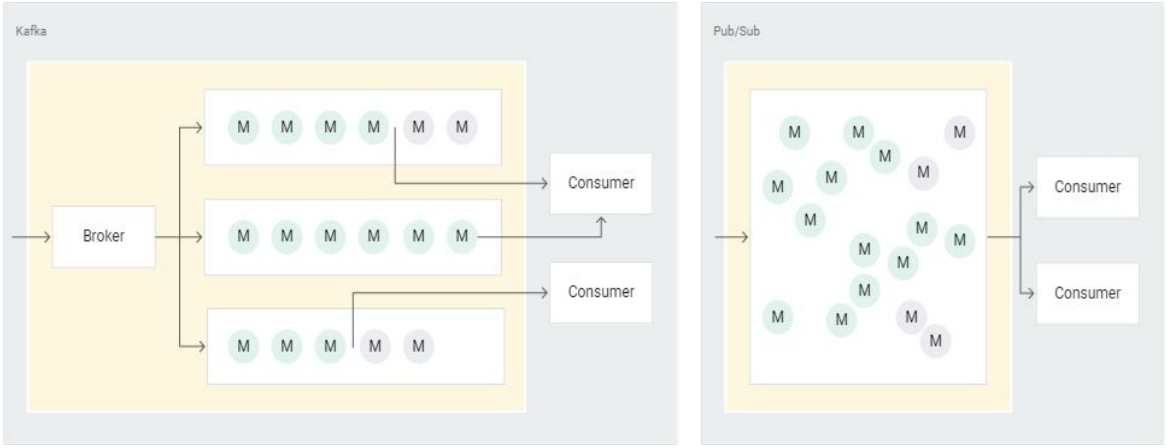
[Choosing between Cloud Tasks and Pub/Sub | Cloud Tasks Documentation (google.com)]

Pub/Sub made easy



#pubsubmadeeasy
https://www.youtube.com/hashtag/pubsubmadeeasy

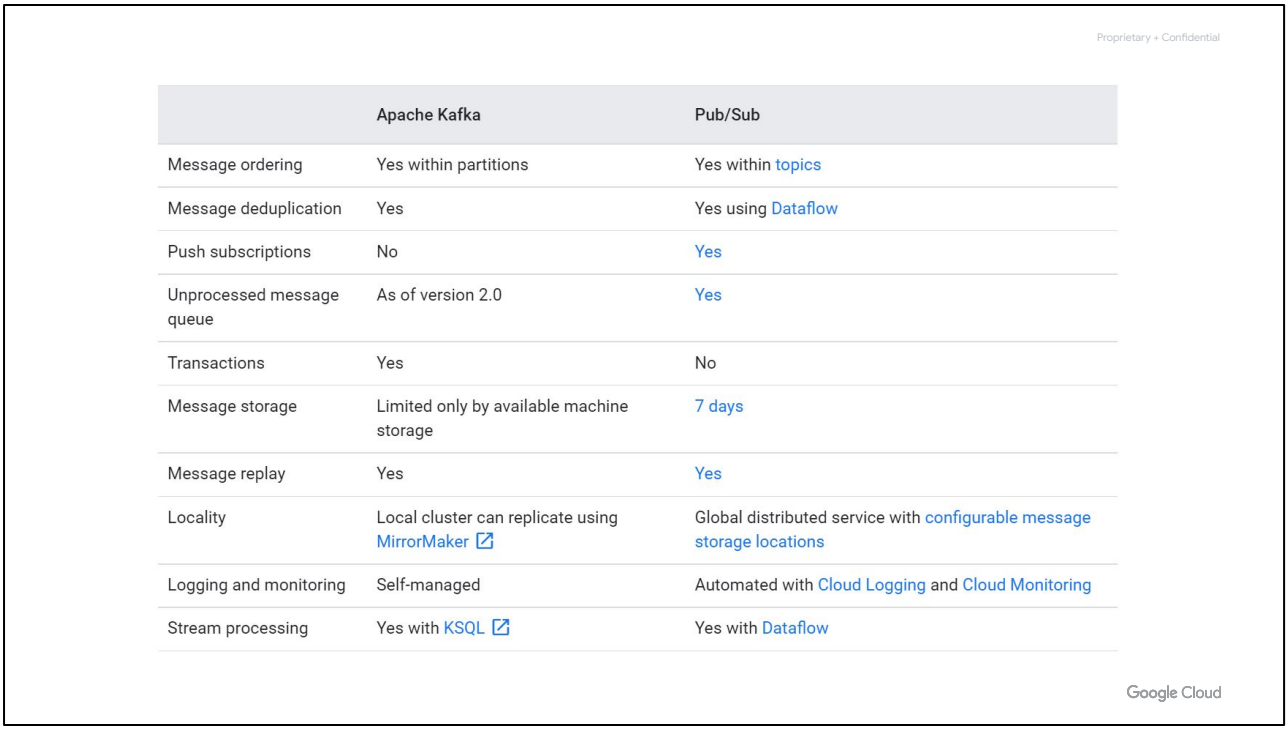
Migrate Kafka to Pub/Sub



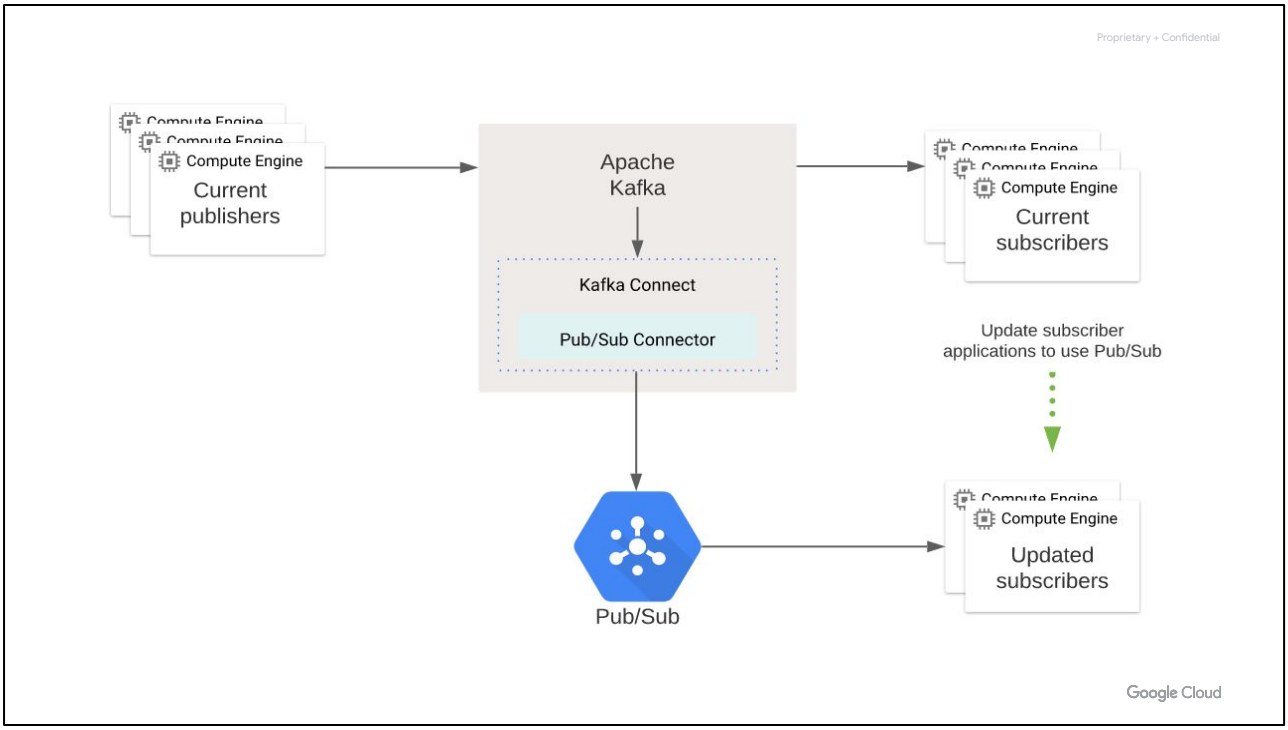
M stands for message

Migration from Kafka to Pub/Sub
https://cloud.google.com/architecture/migrating-from-kafka-to-pubsub

Disaster Recovery for Multi-Region Kafka at Uber
https://eng.uber.com/?s=pub%2Fsub



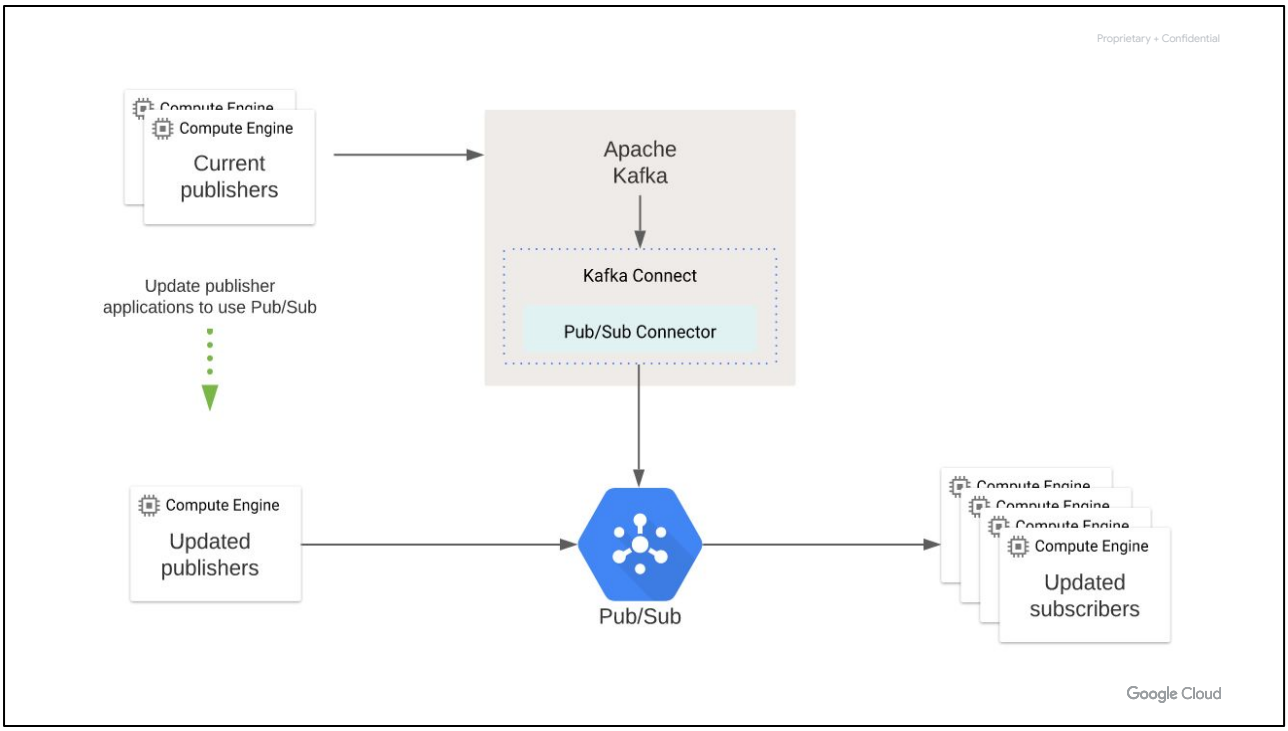
Comparing Features
https://cloud.google.com/architecture/migrating-from-kafka-to-pubsub#comparing_features



Comparing Features
https://cloud.google.com/architecture/migrating-from-kafka-to-pubsub#comparing_features

Phase 1

- Connect Kafka Connect to Pub/Sub Connector
- Migrate subscribers to Pub/Sub
- https://cloud.google.com/pubsub/docs/connect_kafka
- https://cloud.google.com/pubsub/lite/docs/lite_connect_kafka



Comparing Features
https://cloud.google.com/architecture/migrating-from-kafka-to-pubsub#comparing_features

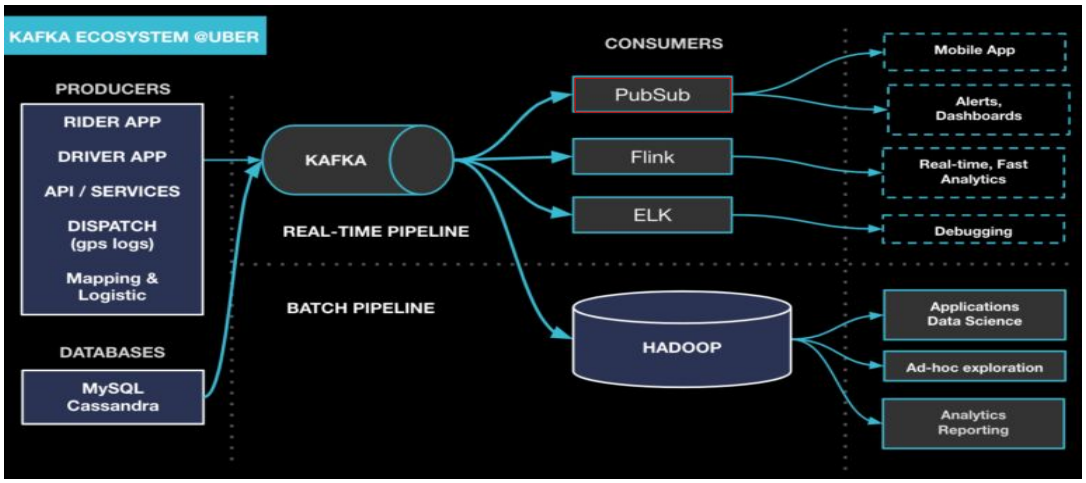
Phase 2

- Once subscribers are migrated to Pub/Bub point publishers from Kafka to Pub/Sub

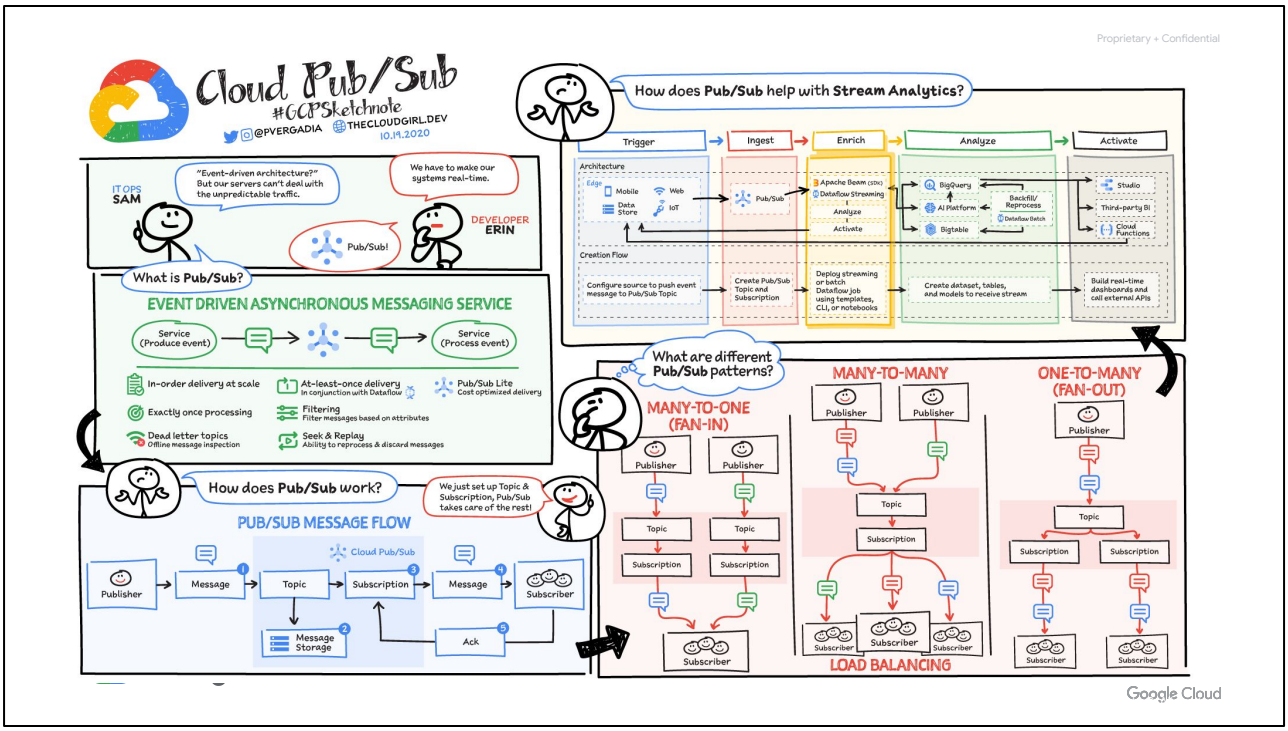
Migration from Kafka to Pub/Sub
https://cloud.google.com/architecture/migrating-from-kafka-to-pubsub

Disaster Recovery for Multi-Region Kafka at Uber
https://eng.uber.com/?s=pub%2Fsub

Migrate Kafka to Pub/Sub



Uber Engineering
https://eng.uber.com/?s=pub%2Fsub



https://thecloudgirl.dev/pubsub.html

https://cloud.google.com/pubsub/docs/overview

Pub/Sub Overview
Pub/Sub is an asynchronous messaging service that decouples services that produce events from services that process events.

Core Concepts
https://cloud.google.com/pubsub/docs/overview#data_model

- **Topic**: A named resource to which messages are sent by publishers.
- **Subscription**: A named resource representing the stream of messages from a single, specific topic, to be delivered to the subscribing application. For more details about subscriptions and message delivery semantics, see the Subscriber Guide.
- **Message**: The combination of data and (optional) attributes that a publisher sends to a topic and is eventually delivered to subscribers.
- **Message attribute**: A key-value pair that a publisher can define for a message. For example, key iana.org/language_tag and value en could be added to messages to mark them as readable by an English-speaking

- subscriber.

This Week's Recommended Activities

1. Review the **exam guide** to assess your own level of expertise and readiness
2. Familiarize yourself with exam **Sample Questions**
3. Labs and Quests for this week:
   a. **Course:** Building Resilient Streaming Analytics Systems on GCP
   b. **Course:** Smart Analytics, Machine Learning, and AI on GCP
4. Labs and Quests for next week: (or review the content):
   a. **Course:** Serverless Data Processing with Dataflow: Foundations
   b. **Course:** Serverless Data Processing with Dataflow: Develop Pipelines

Week 1
Course: Google Cloud Big Data and Machine Learning Fundamentals
https://partner.cloudskillsboost.google/course_templates/3

Quest: Create and Manage Cloud Resources (this is an introductory quest) - Skill Badge
https://partner.cloudskillsboost.google/quests/120

Course: Modernizing Data Lakes & Data Warehouses with Google Cloud
https://partner.cloudskillsboost.google/course_templates/54

Week 2
Course: Building Batch Data Pipelines on Google Cloud
https://partner.cloudskillsboost.google/course_templates/53

Week 3
Course: Building Resilient Streaming Analytics Systems on GCP
https://partner.cloudskillsboost.google/course_templates/52

Course: Smart Analytics, Machine Learning, and AI on GCP
https://partner.cloudskillsboost.google/course_templates/55

Week 4
Course: Serverless Data Processing with Dataflow: Foundations

https://partner.cloudskillsboost.google/course_templates/218

Course: Serverless Data Processing with Dataflow: Develop Pipelines
https://partner.cloudskillsboost.google/course_templates/229

Week 5
Course: Serverless Data Processing with Dataflow: Operations
https://partner.cloudskillsboost.google/course_templates/264

Quest: Perform Foundational Data, ML and AI Tasks - Skill Badge
https://partner.cloudskillsboost.google/quests/117

Week 6
Lab: Optimizing BigQuery for Cost and Performance v1.5
https://partner.cloudskillsboost.google/focuses/18091?parent=catalog

Quest: Build and Optimize Data Warehouses with BigQuery - Skill Badge
https://partner.cloudskillsboost.google/quests/147

Lab: ETL Processing on Google Cloud Using Dataflow and BigQuery
https://partner.cloudskillsboost.google/focuses/11581?parent=catalog

Quest: Engineer data in Google Cloud - Skill Badge
https://partner.cloudskillsboost.google/quests/132

Week 7

Course: Preparing for the Google Cloud Professional Data Engineer Exam
https://partner.cloudskillsboost.google/course_templates/72

Practice: Professional Data Engineer Sample Questions
https://cloud.google.com/certification/practice-exam/data-engineer

Thank you

Google Cloud