

PROJECT SEMESTER REPORT

NETWORK & SERVICE AUTOMATION USING GCP

Under the Guidance of

Faculty Mentor

Dr. Shashank Sheshar Singh
shashank.sheshar@thapar.edu

Industrial Mentor

Mr. Ashish Priydarshi
ashish.priydarshi@ericsson.com

Submitted By:

Name: Shubham Tiwari
Roll No: 101916126
Batch: 4CS12
stiwari_be19@thapar.edu



Submitted to the

Computer Science & Engineering Department
Thapar Institute of Engineering & Technology, Patiala

In Partial Fulfilment of the Requirements for the Degree of

Bachelor of Engineering in Computer Engineering

at

Thapar Institute of Engineering & Technology, Patiala

June 2023

Network & Service Automation using GCP

by Shubham Tiwari

Place of work: Ericsson Global India Pvt. Ltd., Noida

Submitted to the Computer Science & Engineering Department, Thapar Institute of Engineering & Technology

June 2023

In Partial Fulfilment of the Requirements for the Degree of Bachelor of Engineering in Computer Engineering

Abstract:

Cloud computing and cloud services offered by different providers is a very relevant topic nowadays. The need for this is being realised by several organisations. As everything is starting to move towards automation, this field is very interesting to dive into. In my project as a part of my internship, I was lucky enough to work on GCP (Google Cloud Platform) and learn useful skills that are required by the industry. So, I am involved in two projects, first is Data ingestion automation – There are some excel file which consist data of different vendor and their technology. It arrives in a bucket name is GCS bucket which needs to be ingested into BigQuery table using some methods or functions. This function executes when an excel file arrived in GCS bucket and then function pre-process the data and transform into BigQuery compatible format and ingest into the table. And the second one is Light weighting the architecture of Ericsson's internal tool which is using google cloud platform that tool has two high costly components, so our unit task was to eliminate both component and redesign the same architecture having same features and functionality. Namely both component is Google cloud Composer and Google Cloud Dataproc.

Author

Shubham Tiwari

Certified by signature Due to prevailing scenario email from Industrial mentor will serve as

*Take email from mentor → Convert in PDF → Paste that prior certificate
(Name & Signature) (Industrial Coordinator / mentor)*

Certified by Due to prevailing scenario email from Faculty mentor will serve as signature

*Take email from mentor → Convert in PDF → Paste that prior certificate
(Name & Signature) (Faculty mentor)*

Certificate (Project Semester Training) From The Company Or The Organization

As training will be complete on July 25th, 2023, only then training certificate will be available.

TABLE OF CONTENT

Contents

1. Company Profile	5
2. Introduction	7
3. Background	10
4. Objectives	13
5. Methodology	14
6. Observations and Findings	20
7. Limitations	22
8. Conclusions and Future Work	25
9. Bibliography/References	28

1. COMPANY PROFILE

1.1 ABOUT COMPANY:

Ericsson Global India Pvt. Ltd. is a subsidiary of Ericsson, a leading Swedish multinational telecommunications company. Established in India, Ericsson Global India Pvt. Ltd. (EGI) is responsible for **providing a wide range of innovative solutions and services in the field of information and communication technology (ICT)**. With a strong presence in the country, the company caters to the growing demands of the Indian market and plays a vital role in the development of the telecommunications industry.

EGI offers cutting-edge products, solutions, and services that enable communication service providers, enterprises, and governments to deliver advanced connectivity and digital services to their customers. The company's portfolio includes network infrastructure, software platforms, and various professional services aimed at transforming industries and empowering societies with seamless connectivity and technological advancements.

AREA OF EXPERTISE:

Networks: Ericsson is a leader in designing, deploying, and managing advanced communication networks, including 5G and beyond.

Digital Services: The company offers a range of digital solutions to help businesses and governments accelerate their digital transformation.

Managed Services: Ericsson provides comprehensive managed services to optimize network performance and operational efficiency.

Emerging Technologies: Ericsson explores emerging technologies such as Internet of Things (IoT), artificial intelligence, and cloud computing.

Overall, EGI plays a significant role in India's digital transformation journey by delivering advanced solutions and services that shape the future of communication technology, drive economic growth, and improve the lives of people across the country.

1.2 MY POSTION:

I am working in the unit of solution development (or service delivery) which is responsible for the E2E (end-to-end) solution architecture lifecycle management and ownership, the System solution and solution quality according to the requirements and strategy, compliance to Ericsson corporate policies, directives and instructions, and solution components, vendor and 3PP (3rd Party provider) management/selection and governance. And solution development unit is unit of Business Area Cloud Software and Services which provides industry leading solutions for Core Network and Automation, Managed Services, Services Orchestration and Telecom BSS (Business Support System).

Team Name: SDAP (Solution Development Application & Platform), Where I'm working as intern.

So, in solution development unit my contribution is distributed between two main tasks –

i) Ingestion Automation development:

This is about automation data ingestion on BigQuery using Excel files from Google Cloud Bucket involves automatically transferring and importing Excel data into BigQuery. It eliminates manual entry and uses Google Cloud services like Cloud Storage and Cloud Functions. Excel files are uploaded to a bucket, parsed, and transformed into a compatible format, and loaded into BigQuery. This streamlines data integration for analysis, reporting, and querying, enhancing efficiency and decision-making. I developed almost 70% of whole cloud function and deployed to google cloud.

In ingestion automation we have several steps i.e., automate the ingestion of Aggregation mapping, Dataset configuration and KPI (Key performance indicator) configuration

ii) Light weighting the architecture GCP by eliminating google cloud composer:

In this composer is a heavy/costly component of Google cloud platform (GCP) so our task was to eliminate this component with the use of another light components of GCP. So, we used cloud Evantarc, cloud scheduler, Pub/Sub, and cloud run.

Note - This is an internal tool of organization so information about this product will be very limited.

2. INTRODUCTION

2.1 INTRODUCTION

- In this chapter, we will briefly discuss the area of work, which is broadly related to cloud computing, Google Cloud Platform and its component i.e. BigQuery, GCS Bucket and SQL etc. We will then explore the present-day scenario of this work area and what motivates us to work towards this direction and what are the benefits that we enjoy after implementing these solutions.
- We shall also discuss the ideas for our project, the importance of our results and the impact that it has on a broader scale.
- Cloud computing and cloud services offered by different providers is a very relevant topic nowadays. The need for this is being realised by several organisations. As everything is starting to move towards automation, this field is very interesting to dive into. In my project as a part of my internship, I was lucky enough to work on GCP (Google Cloud Platform) and learn useful skills that are required by the industry.
- Currently, a lot of organizations are using cloud computing services such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS) and so on are different levels of services in which the amount data, middleware, OS etc. that we manage varies. There are also different types of deployment models such as on public, private and hybrid clouds.

2.2 MOTIVATION FOR THE PROJECT

- ✓ At the end, we will be able to develop solutions which will be more cost effective and efficient by reducing the number of tools and functions that were used on cloud before.
- ✓ This project will help in enhancing the pace and reducing costs for providing telecom services.
- ✓ It will provide an automated flow of a large collection of data in different formats to the cloud for further analysis.
- ✓ Reducing TCO by analyzing CapEx vs OpEx

As I mentioned above the whole 6 months internship is distributed in two tasks which need to be achieve. Let's talk about in detail on both project what the relevance and outcome of the projects are.

i) **Data Ingestion Automation development:**

Automation data ingestion on BigQuery using Excel files from Google Cloud Bucket offers organizations a streamlined and efficient way to integrate Excel data into their data analysis workflows. It reduces manual effort, improves data accuracy, and enables faster insights and decision-making based on the transformed and loaded data in BigQuery.

Namely Aggregation mappings, Dataset Configuration and KPI configuration three automation integrated into single cloud function. This cloud function has some more features like deleting data, updating data, inserting new data, or appending new data in existing data into BigQuery table. My work was to develop these functions from scratch under supervision of my mentor who assisted me to achieve the goal.

These 3 automations have different excel files as input and having different schema for table. Data of excel files are vendor, technology, or customer specific. **This automation reduces 80% efforts and around 95% times which leads to cost reduction to company.**

ii) **Light weighting the architecture GCP by eliminating google cloud composer:**

The objective of this project is to eliminate the Cloud Composer and Cloud Dataproc from the system by using some different set of rules or mechanisms, because it is heavy and expensive part of whole architecture. (See Fig. 1.)

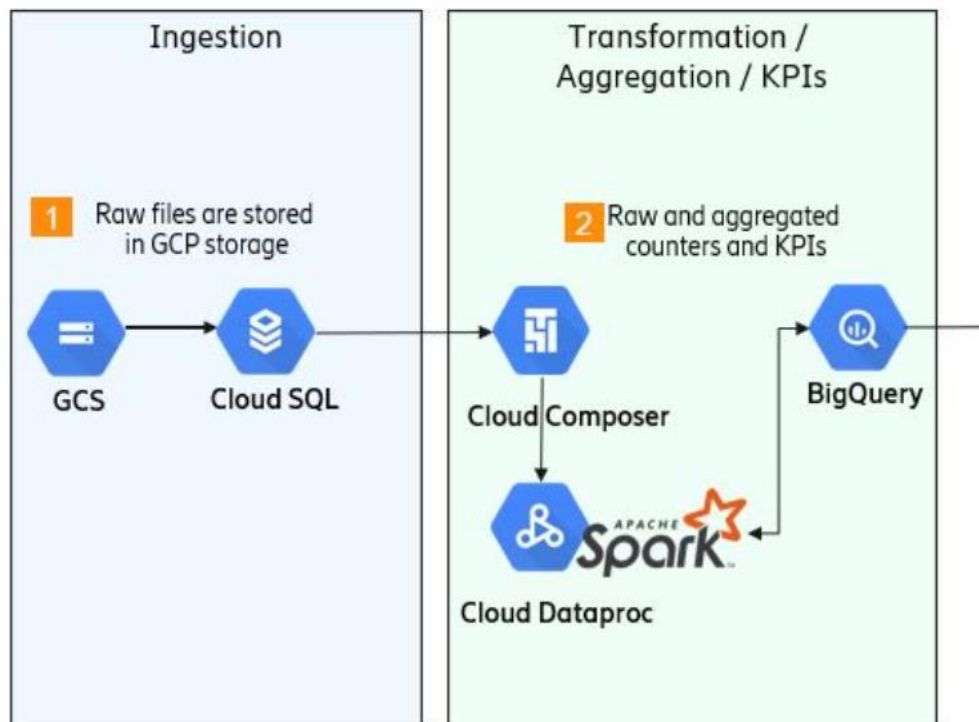


Fig. 1. Part of complete architecture of internal tool of GCP

This task is harder than previous because it has numerous things to be implemented. You will know more about the amended architecture later. My involvement in this project is develop such cloud function which can do same tasks as existing architecture was doing.

2.3 CONCLUSION:

So, in this chapter we have an eagle view of the 6-month internship with two projects. In first one we are automating data ingestion using GCP and in second one we are targeting to reduce cost by eliminating use of cloud Dataproc and Composer from the architecture.

3. BACKGROUND

3.1 INTRODUCTION

In this chapter, we shall discuss the Introduction to the project title, a literature review which includes the present state and developments that have taken place in area of cloud computing, a brief theory on the services that were used on GCP and some background theory on the same. And then we will also dive into both project and components used.

3.2 INTRODUCTION TO PROJECT TITLE

As the project title suggests, I will be working on different tools offered by GCP platform for automating network and services. We are using different GCP component like GCS bucket, BigQuery etc. In BigQuery table we ingest a large amount of data by running Python codes and dynamic SQL queries. We load the data on a table in the cloud. Also, a cloud function is triggered by PubSub which runs our code to provide analysis.

I will also work on Cloud Run, Cloud Dataproc, Composer and Evantarc which uses applications such as PySpark and Scala.

3.3 PROJECT INTRODUCTION:

I worked on two projects till now Data ingestion automation and Light weight architecture. First one in is under testing phase and second one is in developing state. Here I am going to explain about both project's background and how the things is going to be happen.

3.3.1 Data Ingestion Automation development:

The data ingestion process to BigQuery using GCS bucket, Cloud Function, and Pub/Sub involves a series of steps that enable the seamless flow of data from source to destination. Here's a brief working background of this process:

1. Data Source: The process begins with a data source, which can be any system or application that generates or collects data. This data could be in various formats such as CSV, JSON, or Avro.

2. GCS Bucket: Google Cloud Storage (GCS) provides a scalable and durable object storage solution. A GCS bucket is created to store the data files temporarily before they are ingested into BigQuery. The bucket acts as a landing zone for incoming data.

3. Cloud Function: A Cloud Function is a serverless compute platform that allows you to run event-driven code. In this case, a Cloud Function is created and configured as a trigger for the GCS bucket. It is set to be triggered whenever a new file is uploaded or added to the bucket.

4. File Upload: When a new data file is uploaded to the GCS bucket, the Cloud Function is automatically triggered. It receives an event notification indicating the arrival of a new file.

5. Pub/Sub: Google Cloud Pub/Sub is a messaging service that enables asynchronous communication between applications. The Cloud Function publishes a message to a Pub/Sub topic, which acts as a central messaging hub.

6. Subscriber: A subscriber application is set up to receive messages from the Pub/Sub topic. It can be a custom application or another service within the GCP ecosystem.

7. Data Processing: The subscriber application consumes the message and retrieves the details of the newly uploaded file, including its location in the GCS bucket. It can then perform any required data processing or transformation on the file before ingestion into BigQuery.

8. BigQuery Load Job: Once the data processing is complete, the subscriber application initiates a load job in BigQuery. It specifies the GCS file location and the target table in BigQuery where the data should be ingested.

9. Data Ingestion: BigQuery API retrieves the specified file from the GCS bucket and loads it into the specified table. The ingestion process sometimes automatically handles schema detection, but here we are ingesting an existing dynamic schema, data type inference, and other optimizations to ensure efficient data loading.

10. Query and Analysis: Once the data is ingested into BigQuery, it becomes available for querying and analysis. Users can write SQL queries or use other

analytical tools to perform complex analytics, generate insights, and gain valuable information from the ingested data.

By utilizing GCS bucket, Cloud Function, and Pub/Sub in conjunction with BigQuery, the data ingestion process ensures a streamlined flow of data from the source to the destination, enabling real-time or near-real-time analytics and insights.

3.3.2 Light weighting the architecture of GCP:

The main task of this project is removing two heavy components from the project having same feature and functionality. So, Let's have a brief intro of both components.

Google Cloud Dataproc is a fully managed and highly scalable cloud service for running Apache Hadoop, Apache Spark, Apache Hive, and other big data frameworks. It simplifies the process of deploying, managing, and scaling clusters for big data processing and analytics. Dataproc enables you to process large datasets quickly and efficiently, leveraging the power of distributed computing.

Google Cloud Composer is a fully managed workflow orchestration service based on the Apache Airflow framework. It allows you to define, schedule, and monitor complex workflows as directed acyclic graphs (DAGs). Cloud Composer provides a user-friendly interface for designing and managing workflows, automating tasks, and integrating with other Google Cloud services. It simplifies the development and deployment of data pipelines, allowing you to orchestrate the execution of tasks across different services and environments.

Finally, we are removing both component from the project we shall learn how it is possible in detail in methodology section. This project consists of a variety of tasks for example aggregation, key performance factor (KPI) calculation, event triggering, dataset mapping, enriching the data etc. That are not useful for report point of view.

3.4 CONCLUSION:

Therefore, we got an introduction to the project title and its two different projects, which includes GCP services such as BQ, Composer, Dataproc and Cloud Function. We also saw how cloud computing is providing services on the internet but there still are some challenges to tackle specially for second project.

4. OBJECTIVES

4.1 PROJECT OBJECTIVES:

- i.) Using GCP as a service to reduce the burden and increase the performance in the telecom industry so organization can offer services in low budget to customers.
- ii.) Better utilization of available cloud resources i.e., computation power, storage etc.
- iii.) Data ingestion automation which reduces costs, error, time and manual effort.
- iv.) Light weighting the existing architecture of an internal tool which involve GCP

4.2 MY OBJECTIVES:

My objectives while fulfilling the project objectives are following:

- i.) Enhancing my development, debugging and other technical skill as well as teamwork and other soft skills.
- ii.) Hands on experience with cloud platform like GCP and different component of GCP
- iii.) Learn professionalism, work Ethic, networking, building Relationships, Project and Time Management

5. METHODOLOGY

5.1 INTRODUCTION:

In this chapter, we will know about how we implemented logics and methods to achieve the forementioned objectives using GCP and its other components. First, we discuss about pathway for data ingestion automation and then we proceed for light weight GCP.

Development Need:

Programming language – Python 3.10

Modules – Pandas, JSON, Google cloud BigQuery, google cloud storage, Flask, Jinja2, Sqlite3

Database – Google SQL (Because it supports broadest domain

Tools: Excel, VS code, Docker-Container

GCP Components – BigQuery, GCS Bucket, Cloud Function, Cloud Run etc.

And some Ericsson's internal tool

5.2 DATA INGESTION AUTOMATION DEVELOPMENT:

Automation data ingestion on BigQuery using Excel files from Google Cloud Bucket involves the process of automatically transferring and importing Excel data into BigQuery, a powerful cloud-based data warehouse offered by Google. This process **eliminates the need for manual data entry and enables seamless integration of Excel data into BigQuery for further analysis and processing.**

S1) The automation workflow starts with uploading Excel files to a Google Cloud Bucket, which serves as the storage location for the data. Once a new file is added to the bucket, a trigger is set up using Pub/Sub, a messaging service provided by Google Cloud.

Cloud Functions, a serverless compute service, is then utilized to respond to the Pub/Sub trigger. The Cloud Function is configured to execute a predefined code or

script that extracts data from the Excel file and transforms it into a compatible format for ingestion into BigQuery.

S2) Using libraries such as pandas, JSON, BigQuery etc., the Cloud Function parses the Excel file, performs any necessary data transformations or cleansing, and prepares the data for loading into BigQuery.

S3) The transformed data is then loaded into BigQuery using the appropriate APIs or SDKs provided by Google Cloud. BigQuery's scalable and high-performance capabilities enable efficient handling of large volumes of data and facilitate fast querying and analysis. (See Fig. 2)

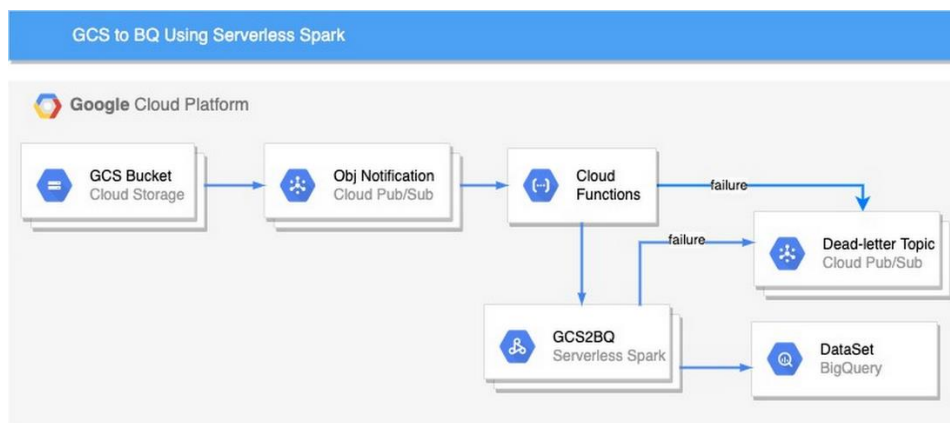


Fig. 2) Data ingestion Automation Architecture

By automating the data ingestion process, organizations can **save time and effort, eliminate manual tasks, and ensure consistent and timely data updates in BigQuery**. This automation enables faster data-driven decision-making, enhances data accuracy, and improves overall operational efficiency.

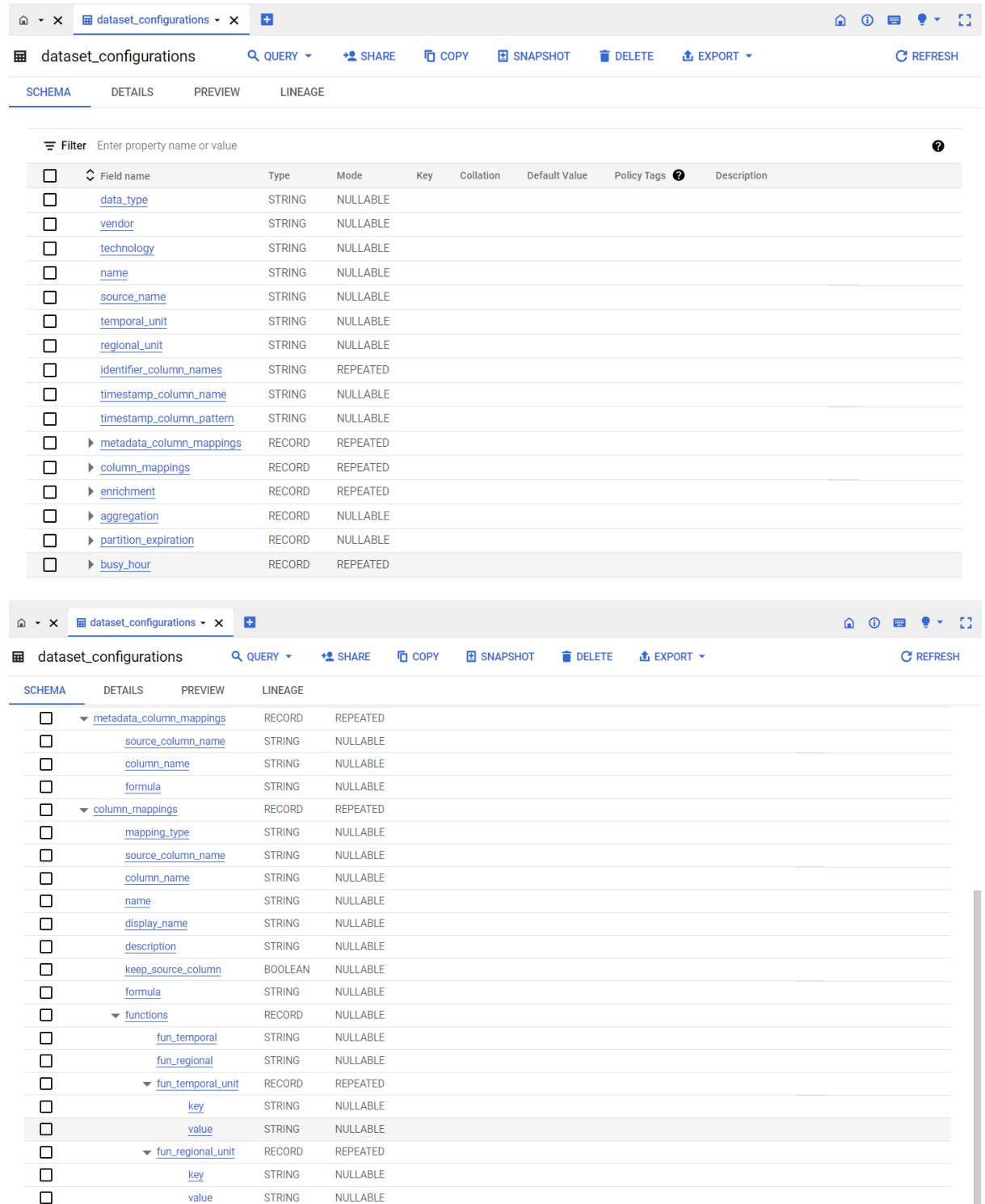
Once the data is ingested into BigQuery, it becomes available for analysis, reporting, and querying using SQL-based queries. BigQuery's scalability and performance capabilities enable handling large volumes of data and executing complex analytical tasks efficiently.

Ground level working:

In this project, we have used BigQuery and Cloud SQL for ingestion of data. We write a Python script for extracting data from different types of excel sheets etc. and parse it into a JSON format. Next step is to write BigQuery operations with given schemas to ingest data into the table on the cloud. After this we utilise the Cloud function to deploy our codes with a triggering topic and publisher. When any excel file arrives in google

cloud bucket publisher publishes a topic and cloud function as subscriber invoke with relevant topic and in logs, we can check the performance of our function.

Here I am attaching an example of schema of dataset configuration so we can have a good understanding about the outcome. (See Fig. 3)



The image shows two screenshots of the BigQuery dataset configuration schema for 'dataset_configurations'. The top screenshot shows the full schema with 18 fields, and the bottom screenshot shows a detailed view of the 'column_mappings' and 'functions' sections.

Top Screenshot: Full Schema

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
data_type	STRING	NULLABLE					
vendor	STRING	NULLABLE					
technology	STRING	NULLABLE					
name	STRING	NULLABLE					
source_name	STRING	NULLABLE					
temporal_unit	STRING	NULLABLE					
regional_unit	STRING	NULLABLE					
identifier_column_names	STRING	REPEATED					
timestamp_column_name	STRING	NULLABLE					
timestamp_column_pattern	STRING	NULLABLE					
metadata_column_mappings	RECORD	REPEATED					
column_mappings	RECORD	REPEATED					
enrichment	RECORD	REPEATED					
aggregation	RECORD	NULLABLE					
partition_expiration	RECORD	NULLABLE					
busy_hour	RECORD	REPEATED					

Bottom Screenshot: Detailed View of column_mappings and functions

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
source_column_name	STRING	NULLABLE					
column_name	STRING	NULLABLE					
formula	STRING	NULLABLE					
mapping_type	STRING	NULLABLE					
source_column_name	STRING	NULLABLE					
column_name	STRING	NULLABLE					
name	STRING	NULLABLE					
display_name	STRING	NULLABLE					
description	STRING	NULLABLE					
keep_source_column	BOOLEAN	NULLABLE					
formula	STRING	NULLABLE					
fun_temporal	STRING	NULLABLE					
fun_regional	STRING	NULLABLE					
fun_temporal_unit	RECORD	REPEATED					
key	STRING	NULLABLE					
value	STRING	NULLABLE					
fun_regional_unit	RECORD	REPEATED					
key	STRING	NULLABLE					
value	STRING	NULLABLE					

SCHEMA	DETAILS	PREVIEW	LINEAGE
<input type="checkbox"/>	column_mappings		RECORD REPEATED
<input type="checkbox"/>	▼ enrichment		RECORD REPEATED
<input type="checkbox"/>	name		STRING NULLABLE
<input type="checkbox"/>	data_type		STRING NULLABLE
<input type="checkbox"/>	vendor		STRING NULLABLE
<input type="checkbox"/>	technology		STRING NULLABLE
<input type="checkbox"/>	▼ identifier_column_names		RECORD REPEATED
<input type="checkbox"/>	source_column_name		STRING NULLABLE
<input type="checkbox"/>	column_name		STRING NULLABLE
<input type="checkbox"/>	▼ column_names		RECORD REPEATED
<input type="checkbox"/>	source_column_name		STRING NULLABLE
<input type="checkbox"/>	column_name		STRING NULLABLE
<input type="checkbox"/>	formula		STRING NULLABLE
<input type="checkbox"/>	completeness_basis		BOOLEAN NULLABLE
<input type="checkbox"/>	▼ aggregation		RECORD NULLABLE
<input type="checkbox"/>	▼ identifier_column_names		RECORD REPEATED
<input type="checkbox"/>	regional_unit		STRING NULLABLE
<input type="checkbox"/>	parent_regional_unit		STRING NULLABLE
<input type="checkbox"/>	identifier_column_names		STRING REPEATED
<input type="checkbox"/>	▼ partition_expiration		RECORD NULLABLE
<input type="checkbox"/>	ingest_days		INTEGER NULLABLE
<input type="checkbox"/>	▼ aggregations		RECORD REPEATED
<input type="checkbox"/>	temporal_unit		STRING NULLABLE
<input type="checkbox"/>	agg_days		INTEGER NULLABLE
<input type="checkbox"/>	kpi_days		INTEGER NULLABLE
<input type="checkbox"/>	▼ busy_hour		RECORD REPEATED
<input type="checkbox"/>	temporal_unit		STRING NULLABLE
<input type="checkbox"/>	agg_days		INTEGER NULLABLE
<input type="checkbox"/>	kpi_days		INTEGER NULLABLE
<input type="checkbox"/>	▼ busy_hour		RECORD REPEATED
<input type="checkbox"/>	vendor		STRING NULLABLE
<input type="checkbox"/>	technology		STRING NULLABLE
<input type="checkbox"/>	kpi_source_name		STRING NULLABLE
<input type="checkbox"/>	label		STRING NULLABLE
<input type="checkbox"/>	bh_temporal_units		STRING REPEATED

Fig. 3) Schema for Dataset Configuration table

In summary, the automation of data ingestion on BigQuery using Cloud Functions and Pub/Sub simplifies the process of importing Excel files from a Google Cloud Bucket, transforming the data, and loading it into BigQuery for analysis, thereby enabling organizations to leverage their data more effectively. **It reduces manual effort, improves data accuracy, and enables faster insights and decision-making based on the transformed and loaded data in BigQuery.**

5.2 LIGHT WEIGHTING THE ARCHITECTURE:

The existing architecture of the tools has two heavy component Cloud composer and Cloud Dataproc as you can see Fig. 4. These components were using heavy chunk of whole cost so our unit was working how we can eliminate these two components which could help in overall cost reduction.

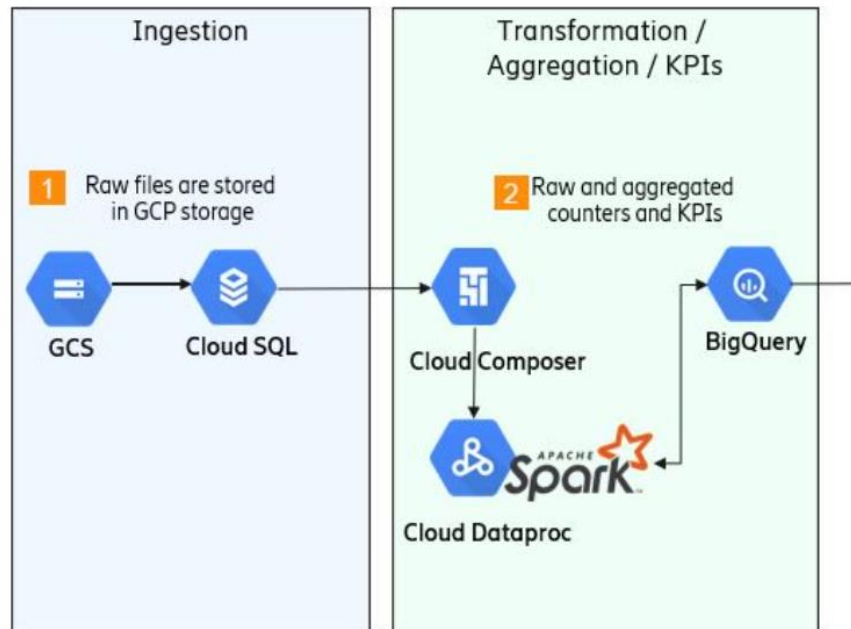


Fig. 4) Old Architecture of GCP

For eliminating Google Cloud Composer and Cloud Dataproc by utilizing other services like Eventarc, Cloud Functions, Cloud Run, and Cloud Scheduler, we are following the following steps (See Fig. 5):

5.2.1. Migrate DAGs (Directed Acyclic Graphs): Extract the DAGs from Google Cloud Composer and convert them into individual Cloud Functions or Cloud Run services. Each DAG can be transformed into a separate service.

5.2.2. Event Trigger Configuration: We are using Eventarc, a serverless eventing platform on Google Cloud, to configure triggers for events that will initiate the execution of your Cloud Functions or Cloud Run services. For example, we can set up triggers based on changes to a Cloud Storage bucket or Pub/Sub messages.

5.2.3. Implement Business Logic: Rewriting and reimplementing the business logic of our DAGs as standalone functions or microservices using Cloud Functions and

Cloud Run. By Ensuring that the functionality and dependencies from the DAGs are replicated in the new services.

5.2.4. Deployment: Deploy your Cloud Functions and Cloud Run services to Google Cloud. We can use the respective deployment methods for each service, such as deploying Cloud Functions using the `gcloud` command-line tool or using Cloud Build for Cloud Run services. But here we are using container image and deploying it into the google cloud.

5.2.5. Scheduling: For scheduling tasks, we can leverage Cloud Scheduler, which allows us to define cron job-like schedules to trigger your Cloud Functions or Cloud Run services at specific intervals.

5.2.6. Monitoring and Logging: Configure monitoring and logging for your new services using Google Cloud Monitoring and Cloud Logging to gain insights into the execution and performance of your functions or microservices.

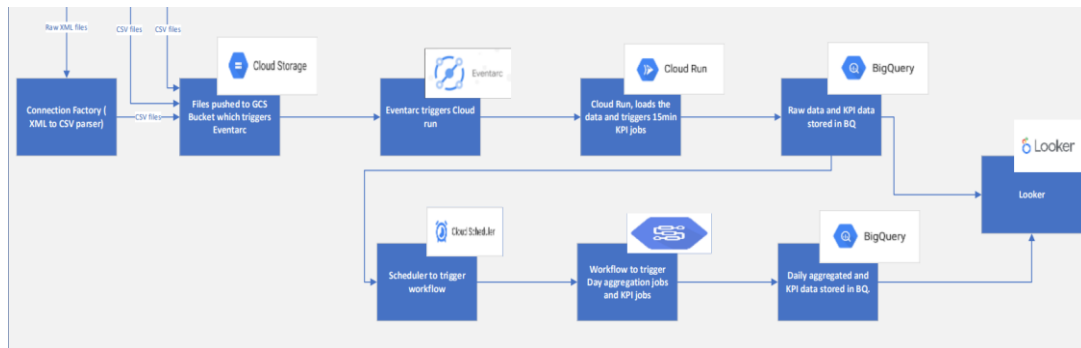


Fig. 5). Lightweight architecture of the platform

By following these steps, we can eliminate the dependency on Google Cloud Composer & Google cloud Dataproc and leveraging Eventarc, Cloud Functions, Cloud Run, and Cloud Scheduler to orchestrate and execute our tasks in a more lightweight and modular manner. This approach provides more flexibility, scalability, and cost-efficiency for our workflows.

6. OBSERVATIONS AND FINDINGS

6.1 INTRODUCTION:

As a developer working on a real-time project that involves Google Cloud Platform (GCP) and its various components like BigQuery, GCS bucket, Cloud Functions, and Cloud Run, I have several benefits and findings in following ways:

1. Familiarity with Modern Cloud Technologies: Working with GCP exposes us to modern cloud technologies and practices. I gain experience in building scalable, serverless architectures and working with managed services for data storage, processing, and analysis. This expertise is highly valuable in today's cloud-centric industry.

2. Enhanced Development Productivity: GCP's managed services and serverless offerings significantly reduce the operational overhead and infrastructure management tasks for developers. With services like Cloud Functions and Cloud Run, I can focus on writing code and building applications rather than dealing with infrastructure provisioning, scaling, and maintenance. This improves my development productivity and speeds up the time-to-market for our projects.

3. Seamless Integration and Interoperability: GCP provides a wide range of services and APIs that seamlessly integrate with each other. As a developer, this enables me to leverage the strengths of different GCP components and build cohesive, end-to-end solutions. The ability to integrate services like BigQuery, GCS bucket, Cloud Functions, and Cloud Run allows me to create complex real-time workflows and leverage the power of multiple services in a cohesive manner.

4. Learning Opportunities: Working with GCP and its components exposes me to a vast ecosystem of technologies and tools. This presents ample learning opportunities to expand my skill set and deepen my understanding of cloud computing, data analytics, serverless architectures, and more. GCP provides comprehensive documentation, tutorials, and online resources that can further enhance my knowledge and expertise.

5. Scalability and Performance Optimization: GCP's services are designed to handle massive workloads and scale seamlessly. I learn how to design efficient data processing pipelines, leverage caching mechanisms, and make use of GCP's auto-scaling capabilities to ensure our real-time project performs optimally even under high load.

6. Data Analytics and Insights: GCP's BigQuery offers powerful data analytics capabilities, allowing me to gain insights from large datasets in real-time. I get the opportunity to work with BigQuery and develop expertise in querying, analyzing, and visualizing data. This knowledge is highly valuable in today's data-driven world, where organizations heavily rely on real-time analytics to make informed decisions.

7. Collaboration and Teamwork: GCP provides collaboration tools and features that facilitate teamwork and streamlined development processes. Services like Cloud Source Repositories, Cloud Build, and Cloud Deployment Manager enable version control, continuous integration, and automated deployments, fostering efficient collaboration among developers. This allows me to work effectively in a team environment and adhere to modern software development practices.

8. Industry Relevance and Demand: GCP is one of the leading cloud platforms in the industry, and companies across various domains are adopting it for their real-time projects. By gaining expertise in GCP and its components, I position myself as a skilled professional in high demand. This can lead to exciting career opportunities and increased market value as organizations seek developers with cloud expertise.

6.2 CONCLUSION:

Overall, working on a real-time project with GCP and its components equips me with valuable skills, enhances my development productivity, and opens doors to diverse career opportunities in the cloud computing domain.

7. LIMITATIONS

7.1 INTRODUCTION

In this chapter, we will learn some limitations or criteria of both project where project might fail or may crash. First, we proceed with first and then light weight GCP.

7.2 LIMITATIONS OF DATA INGESTION AUTOMATION:

While data ingestion to BigQuery using GCS bucket, Cloud Function, and Pub/Sub is a reliable and robust process, there are some scenarios where failures can occur. Here are a few brief reasons where the data ingestion process may encounter issues:

1. Network Connectivity: Network connectivity issues between the data source, GCS bucket, Cloud Function, Pub/Sub, or BigQuery can cause data ingestion failures. Intermittent or unstable network connections can disrupt the flow of data and prevent successful transmission or processing.

2. Authentication and Access Permissions: Improper authentication or misconfigured access permissions can lead to data ingestion failures. If the Cloud Function, Pub/Sub, or BigQuery does not have the necessary credentials or permissions to access the required resources, it will be unable to perform the required operations.

3. Data Format or Structure: Data ingestion can fail if the format or structure of the data being uploaded to the GCS bucket does not conform to the expected schema in BigQuery. For example, if the data file has missing or mismatched fields, incompatible data types, or invalid formatting, it may result in ingestion errors or data loss.

4. Incomplete Data Upload: If the data upload process to the GCS bucket is interrupted or incomplete, it can lead to ingestion failures. This can happen due to network disruptions, timeouts, or premature termination of the upload process. In such cases, the Cloud Function may not receive the expected trigger event, and the data may not be processed further.

5. Service Limitations or Quotas: Data ingestion can fail if any of the involved services, such as GCS bucket, Cloud Function, Pub/Sub, or BigQuery, encounter limitations or exceed usage quotas. For example, if the GCS bucket reaches its storage limit, the Cloud Function exceeds its execution time limit, or Pub/Sub exceeds its message throughput capacity, the ingestion process may be disrupted or halted.

6. Data Transformation Errors: If the subscriber application, responsible for data processing or transformation before ingestion, encounters errors or issues, it can cause data ingestion failures. This can happen if there are bugs or logic errors in the data processing code, leading to incorrect data transformations or incompatible data formats for ingestion into BigQuery.

7. Schema Evolution: If the schema of the target BigQuery table evolves over time, data ingestion can fail if the incoming data does not match the updated schema. This can occur when there is a mismatch in field names, data types, or additional required

fields in the updated schema. In such cases, data may need to be transformed or mapped appropriately before ingestion.

7.3 LIMITATIONS IN LIGHT WEIGHT GCP:

As we are using set of components of GCP so there are several limitations with each component. So here are some possible reasons for failures while working with Google Cloud services:

- Cloud Run:
 - Deployment issues: Incorrect configuration or missing dependencies.
 - Resource limitations: Inadequate resources allocated to a service.
 - Network connectivity problems.
- Eventarc:
 - Misconfigured triggers.
 - Permission issues.
 - Event payload formatting problems.
- Cloud Scheduler:
 - Misconfigured schedules or time zones.
 - Permission and authentication misconfigurations.
 - Target service failures.
- BigQuery:
 - Data ingestion errors.
 - Query mistakes or limitations.
 - Data access and permission problems.
- GCS Bucket:
 - Permission misconfigurations.
 - Networking or connectivity issues.
 - Exceeding storage limits.

7.4 COCNLUSION:

To mitigate these potential issues, it is important to ensure proper error handling, monitoring, and logging mechanisms are in place. Implementing retry mechanisms, performing thorough data validation, and conducting regular testing can help identify and address any issues in the data ingestion process to ensure reliable and successful ingestion into BigQuery.

Understanding these potential failure points can help address and troubleshoot issues effectively while working with these Google Cloud services. And we are using GCP functionality and using all considerable debugging process while implanting the whole.

8. CONCLUSIONS AND FUTURE WORK

8.1 INTRODUCTION:

In 6 months of internship, we completed Data Ingestion Automation project from scratch to deployment with continuous integration and continuous deployment on GCP. So firstly, I am concluding the automation part of project followed by Lightweight GCP tool which is used in organization.

8.2 DATA INGESTION AUTOMATION:

The automated ingestion of BigQuery data into a Google Cloud Bucket by using Cloud Functions and Pub/Sub simplifies the process of importing Excel files from a Google Cloud Bucket, transforming the data, and loading it into BigQuery so that organizations can optimize their data usage.

- As a result, greater efficiency is achieved, data accuracy is improved, and insights and decisions can be made more quickly based on the transformed and loaded data in BigQuery.
- As there are a lot of manual tasks for ingesting data with slightly different parameters that can be automated using this experience and idea.

Automating data ingestion brings a multitude of benefits to organizations.

Firstly, it saves time and reduces labour costs by eliminating manual entry and processing. Automation ensures consistent and accurate data capture, enhancing overall data quality. Additionally, automated processes can handle large data volumes and integrate data from diverse sources and formats, providing scalability and flexibility. By seamlessly integrating data ingestion with processing and analytics, automation streamlines data processing pipelines, improving efficiency and decision-making. Overall, automating data ingestion empowers organizations to optimize their data management, drive operational efficiency, and stay competitive in the data-driven landscape.

8.3 LIGHTWEIGHT GCP:

For light weighting architecture, we are eliminating the dependency on Google Cloud Composer and Google Cloud Dataproc and leverage Eventarc, Cloud Functions, Cloud Run, and Cloud Scheduler to orchestrate and execute our tasks in a more lightweight

and modular manner. **For our workflows, this approach provides greater flexibility, scalability, and cost-efficiency.**

Obviously, there are some limitations which can be mitigated by ensuring proper error handling, monitoring, and logging mechanisms are in place. To ensure reliable and successful ingestion of data into BigQuery, retry mechanisms, thorough data validation, and regular testing can help identify and address any issues.

Working with these Google Cloud services can be easier if we are aware of these potential failure points. The entire implanting process is based on GCP functionality and extensive debugging.

8.4 MY EXPERIENCE:

My experience working on a real-time project with GCP, and its components has equipped me with valuable skills, enhanced my productivity as a developer, and opened up many career opportunities.

Working in an MNC like Ericsson provides global opportunities, career development, competitive compensation, a diverse work environment, exposure to cutting-edge technologies, professional growth, brand recognition, cross-cultural collaboration, stability, and work-life balance. These factors contribute to a rewarding and fulfilling my career experience.

If I get a chance of doing same project again then I will incorporate the following changes –

Firstly, I will get a clearer idea of the requirements necessary for the project.

- I will implement robust error handling mechanisms and logging capabilities. This will help in identifying and resolving issues quickly during development and in production environments.
- Design the project with a modular and scalable architecture to accommodate future growth and changes. Break down the project into smaller, reusable components that can be easily maintained and expanded.
- Maintain detailed documentation throughout the project, including design decisions, configurations, and code explanations. This documentation will be

invaluable for future reference and troubleshooting. I was not maintaining documentation of tasks, but it is must.

- Take the time to thoroughly plan and design the project before starting development. Clearly define the project requirements, data flow, and expected outcomes. This will help us to avoid unnecessary rework and I will try to ensure a more efficient implementation.

8. BIBLIOGRAPHY/REFERENCES

- [1]. Google Cloud, <https://cloud.google.com/>
- [2]. Rakibul Hassan, “Cloud Computing: Literature Review”
- [3]. TechTarget, <https://www.techtarget.com/searchdatamanagement/definition/Google-BigQuery#:~:text=Google%20BigQuery%20is%20a%20cloud,using%20a%20SQL%20Dlike%20syntax>
- [4]. Medium - <https://medium.com/>
- [5]. <https://en.wikipedia.org/>
- [6]. <https://www.ericsson.com/en>
- [7]. <https://console.cloud.google.com/run>
- [8]. <https://console.cloud.google.com/storage>
- [9]. <https://console.cloud.google.com/compute>
- [10]. <https://console.cloud.google.com/functions>