

DATAFLOW

Project Title: Running WordCount Pipeline in Google Cloud Dataflow

Abstraction:

In this project, we aim to leverage the capabilities of Google Cloud Dataflow alongside Apache Beam SDK to build an efficient data transformation pipeline in the cloud environment. By enabling the Dataflow API and assigning appropriate IAM roles, we ensure seamless access and management of resources. The installation of Apache Beam SDK in Cloud Shell facilitates the development and execution of data processing tasks. Cloud Storage buckets are utilized for storing input and output data securely. The execution of an example WordCount pipeline showcases the functionality of Dataflow in transforming data at scale. Monitoring tools provided by Dataflow enable real-time tracking of job performance, ensuring efficient resource allocation. Upon job completion, the output data can be easily accessed and analyzed. Through this project, we demonstrate the power and flexibility of cloud-based data processing solutions for modern data-driven applications.

Introduction:

This project aims to harness the capabilities of Google Cloud Dataflow and Apache Beam SDK to build a robust data transformation pipeline in the cloud environment. By seamlessly integrating these technologies, we enable organizations to efficiently process large volumes of data while taking advantage of the scalability and flexibility offered by the cloud.

Tools and Technologies used:

- **Google Cloud Platform:** Google Cloud Platform (GCP) is a comprehensive suite of cloud computing services provided by Google, offering infrastructure, platform, and software services for businesses and developers. GCP provides scalable and reliable resources that can be dynamically provisioned to meet fluctuating demand, ensuring high availability and performance. Security is a top priority for GCP, with robust features such as encryption, identity and access management, and built-in compliance controls to protect customer data. GCP's global network infrastructure ensures low-latency and high-speed connectivity across regions, facilitating

efficient data transfer and application deployment. With a wide range of services for big data analytics, machine learning, and artificial intelligence, GCP empowers organizations to derive valuable insights and drive innovation from their data. GCP offers transparent and competitive pricing models, allowing customers to pay only for the resources they use and providing tools for cost optimization and budget management.

- **Dataflow:** Google Cloud Dataflow is a fully managed service for executing and managing data processing pipelines, offering both batch and stream processing capabilities. Developed by Google and based on Apache Beam SDK, Dataflow provides a unified programming model for building complex data processing workflows that can scale dynamically to handle massive datasets. With Dataflow, developers can focus on writing business logic without worrying about infrastructure management, as the service automatically handles resource provisioning, scaling, and optimization. Dataflow offers advanced features such as windowing, watermarking, and stateful processing, enabling the processing of real-time streaming data with high accuracy and efficiency. The service integrates seamlessly with other Google Cloud Platform services such as BigQuery, Pub/Sub, and Cloud Storage, facilitating data ingestion, transformation, and analysis across the entire data pipeline.
- **Apache Beam:** Apache Beam is an open-source unified programming model for defining both batch and streaming data processing pipelines. It provides a high-level API that abstracts the complexities of distributed computing, allowing developers to write data processing logic in a language-agnostic manner. Apache Beam supports multiple programming languages, including Java, Python, and Go. The core abstraction in Apache Beam is the "PCollection," which represents a distributed dataset that can be processed in parallel across multiple nodes in a cluster. Apache Beam's model is designed to be portable, allowing pipelines to run on various execution engines, including Apache Flink, Apache Spark, Google Cloud Dataflow, and others. Apache Beam provides a rich set of built-in transformations and connectors for common data processing tasks, such as filtering, aggregating, joining, and reading from/writing to different data sources. Apache Beam simplifies the development and deployment of data processing pipelines, accelerating time to insights and enabling organizations to leverage their data effectively for business intelligence and decision-making.

Description:

The project involves the following key steps:

- Enable Dataflow in a Google Cloud project.
- Add the necessary IAM roles to the Compute Engine default service account.
- Install the Apache Beam SDK in Cloud Shell to let Dataflow transform data for the pipelines.
- Set up a Cloud Storage bucket for the output data.
- Run the example WordCount pipeline in Dataflow as a job.
- Monitor your job in Dataflow.
- View the output of your job.
- Clean up to avoid billing.

Text File Description: This is how, the input text file looks like and it is named as “word-count.txt”

```
This is a Japanese doll
The team members were hard to tell apart since they all wore their hair in a ponytail
As the years pass by we all know owners look more and more like their dogs
If you don't like toenails you probably shouldn't look at your feet
He was disappointed when he found the beach to be so sandy and the sun so sunny
When he encountered maize for the first time he thought it incredibly corny
Situps are a terrible way to end your day
Toddlers feeding raccoons surprised even the seasoned park ranger
Edith could decide if she should paint her teeth or brush her nails
Her daily goal was to improve on yesterday
Tomorrow will bring something new so leave today as a memory
His son quipped that power bars were nothing more than adult candy bars
He wondered why at 18 he was old enough to go to war but not old enough to buy cigarettes
If my calculator had a history it would be more embarrassing than my browser history
The hummingbird's wings blurred while it eagerly sipped the sugar water from the feeder
He went on a whiskey diet and immediately lost three days
This is the last random sentence I will be writing and I am going to stop mid-sent
I come from a tribe of head-hunters so I will never need a shrink
The delicious aroma from the kitchen was ruined by cigarette smoke
Weather is not trivial - it's especially important when you're standing in it
She had a difficult time owning up to her own crazy self
Gary didn't understand why Doug went upstairs to get one dollar bills when he invited him to go cow tipping
He loved eating his bananas in hot dog buns
Peanut butter and jelly caused the elderly lady to think about her past
We have never been to Asia nor have we visited Africa
A dead duck doesn't fly backward
Having no hair made him look even hairier
The secret ingredient to his wonderful life was crime
He played the game as if his life depended on it and the truth was that it did
Baby wipes are made of chocolate stardust
He is no James Bond his name is Roger Moore
A quiet house is nice until you are ordered to stay in it for months
His thought process was on so many levels that he gave himself a phobia of heights
```

Project Workflow:

Step 1: Enabling Dataflow API

- Access GCP Console: Navigate to the GCP Console (<https://console.cloud.google.com/>).
- Open APIs & Services: In the navigation bar, select APIs & Services.
- Enable Dataflow API:
 - Click on Library.
 - Search for Dataflow API.
 - Enable the Dataflow API for your project.

Step 2: Granting IAM Permissions

- The Compute Engine default service account requires specific roles to run Dataflow jobs.
- Assign the following roles to the Compute Engine Default service account:
 - Dataflow Worker
 - Storage Object Admin

The screenshot shows the Google Cloud IAM Permissions interface. The left sidebar has a 'Permissions' icon selected. The main area title is 'Permissions for project "naidu-prasanth"'. It says 'These permissions affect this project and all of its resources.' Below this, there are tabs for 'VIEW BY PRINCIPALS' (selected) and 'VIEW BY ROLES'. Under 'GRANT ACCESS', a table lists a single entry: '79729997753-compute@developer.gserviceaccount.com' with the role 'Editor'. A 'Filter' input field is present. On the right, there are 'Security insights' and a pencil icon for editing. A checkbox for 'Include Google-provided role grants' is checked. The top navigation bar includes 'Google Cloud', a dropdown for 'naidu-prasanth', a search bar, and various icons.

Type	Principal	Name	Role	Security insights
<input type="checkbox"/>	79729997753-compute@developer.gserviceaccount.com	Compute Engine default service account	Editor	?

Step 3: Installing Apache Beam SDK (Cloud Shell)

- Open Cloud Shell: In the GCP Console, navigate to the top right corner and click on the "Activate Cloud Shell" button.
- Install Beam SDK: Use the package manager for your chosen language to install the Apache Beam SDK (e.g., Python: pip install apache-beam[gcp]).

Step 4: Creating a Cloud Storage Bucket

A **bucket** is a fundamental concept in **Google Cloud Storage**. Here are the key points:

Purpose of Buckets:

- Buckets serve as the **basic containers** for storing data in Cloud Storage.
- **Everything** you store in Cloud Storage must be contained within a bucket.
- Unlike directories or folders, **buckets cannot be nested** inside one another.
- There is **no limit** to the number of buckets you can have in a project or location.

Bucket Properties:

- When you create a bucket, you give it a **globally-unique name** and specify a **geographic location** where the bucket and its contents will be stored.
- You cannot change the name or location of an existing bucket directly. Instead, you can create a new bucket with the desired properties and move the contents from the old bucket to the new one.
- The **pricing** for buckets depends on factors such as location and storage classes of objects within them.

1. Access Control:

- You can use **Identity and Access Management (IAM)** to control access to individual buckets.
- IAM allows you to manage permissions for users and services accessing the bucket.

2. Bucket Naming Rules:

- Bucket names must meet specific requirements:
 - Only **lowercase letters, numeric characters, dashes, underscores, and dots** are allowed.
 - Spaces are **not allowed**.
 - Names containing dots require **verification**.
 - Bucket names must start and end with a **number or letter**.
 - Bucket names must contain **3-63 characters**.
 - Names containing dots can have up to **222 characters**, but each dot-separated component cannot exceed 63 characters.
 - Bucket names cannot be represented as an **IP address**.

- Bucket names cannot begin with the prefix "**goog**".
- Avoid using "**google**" or similar misspellings.
- Bucket names are **publicly visible**, so avoid using personally identifiable information (PII) or sensitive data.

- Access Storage Section: In the navigation bar, select Storage.
- Create Bucket: Click on "Create bucket".
- Configure Bucket: Choose a unique bucket name and region, then click "Create".

Name your bucket

Pick a globally unique, permanent name. [Naming guidelines](#)

Tip: Don't include any sensitive information

LABELS (OPTIONAL)

CONTINUE

Good to know

Location pricing

Storage rates vary depending on the storage class of your data and location of your bucket. [Pricing details](#)

Current configuration: Region / Standard

Item	Cost
asia-south1 (Mumbai)	\$0.023 per GB-month

ESTIMATE YOUR MONTHLY COST

Here we are creating input bucket named “wordcount-input” to store the text file on which we are going to perform wordcount.

Your data is always protected with Cloud Storage but you can also choose from these additional data protection options to add extra layers of security.

Data protection

Soft delete policy (For data recovery)
When enabled, deleted objects will be kept for a specified period after they're deleted and can be restored during this time. [Learn more](#)

Object versioning (For version control)
For restoring deleted or overwritten objects. To minimize the cost of storing versions, we recommend limiting the number of nonconcurrent versions per object and scheduling them to expire after a number of days. [Learn more](#)

Retention (For compliance)
For preventing the deletion or modification of the bucket's objects for a specified period of time.

DATA ENCRYPTION

CREATE **CANCEL**

Now we are going to upload text file named “word-count.txt” to the “wordcount-input” bucket.

The screenshot shows the Google Cloud Storage interface. The top navigation bar includes 'Google Cloud', a user dropdown for 'naidu-prasanth', a search bar, and various navigation icons. Below the header, the title 'Bucket details' is shown next to a back arrow. The main section is titled 'wordcount-input'. It displays basic bucket metadata: Location (asia-south1 (Mumbai)), Storage class (Standard), Public access (Not public), and Protection (None). A sidebar on the left contains icons for buckets, metrics, and settings. The main content area has tabs for 'OBJECTS', 'CONFIGURATION', 'PERMISSIONS', 'PROTECTION', 'LIFECYCLE', 'OBSERVABILITY', 'INVENTORY REPORTS', and 'OPERATIONS'. Under the 'OBJECTS' tab, there is a table showing one object: 'Word-count.txt'. The table columns include Name, Size, Type, Created, Storage class, Last modified, and Public access. The object 'Word-count.txt' has a size of 12.6 KB, is a text/plain type, was created on Apr 22, 2024, at 12:27:36 PM, is in the Standard storage class, last modified on Apr 22, 2024, at 12:27:36 PM, and has Not public public access.

Now we are going to create output bucket named “wordcount-output1” to store the result.

The screenshot shows the 'Create a bucket' wizard. The top navigation bar is identical to the previous screenshot. The main form has several steps completed with checkmarks: 'Name your bucket' (Name: wordcount-output1), 'Choose where to store your data' (Location: asia-south1 (Mumbai), Location type: Region), 'Choose a storage class for your data' (Default storage class: Standard), and 'Choose how to control access to objects' (Public access prevention: Off, Access control: Uniform). A 'Good to know' sidebar provides information about location pricing, mentioning storage rates vary depending on the storage class and location of the bucket, with a link to 'Pricing details'. It also shows the current configuration as Region / Standard. A table below lists the item 'asia-south1 (Mumbai)' with a cost of '\$0.023 per GB-month'. At the bottom, there is a section titled 'ESTIMATE YOUR MONTHLY COST'.

Now we can verify both the input and output buckets which we created earlier.

The screenshot shows the Google Cloud Storage Buckets page. At the top, there's a navigation bar with 'Google Cloud' and a dropdown for 'naidu-prasanth'. A search bar says 'Search (/) for resources, docs, products, and more' with a 'Search' button. To the right are icons for refresh, 4 notifications, help, and profile. Below the bar, a banner informs about Security Command Center Premium requirements. The main area has a sidebar with icons for buckets, logs, metrics, and settings. It shows a table of buckets:

Name	Created	Location type	Location	Default storage class	Last modified	Public access	Access
wordcount-input	Apr 22, 2024, 12:27:17 PM	Region	asia-south1	Standard	Apr 22, 2024, 12:27:17 PM	Not public	Unif...
wordcount-output1	Apr 22, 2024, 12:28:58 PM	Region	asia-south1	Standard	Apr 22, 2024, 12:28:58 PM	Not public	Unif...

Step 5: Running the WordCount Pipeline

There are two options to run the WordCount pipeline:

Option A: Using a Dataflow Template

- Access Dataflow Section: In the navigation bar, select Dataflow

The screenshot shows the Google Cloud Dataflow section. The navigation bar includes 'Google Cloud', 'naidu-prasanth', a search bar ('dataflow'), and standard navigation icons. The main area has a sidebar with icons for jobs, logs, metrics, and settings. It displays a table of jobs and a detailed view of the 'Dataflow' template:

JOBS

Name	Type	End time	Elapsed time

PRODUCTS & PAGES

- Dataflow - Streaming analytics service
- Jobs - Dataflow
- Monitoring - Dataflow
- Overview - Dataflow

DOCUMENTATION & TUTORIALS

- Dataflow - Dataflow is a fully managed streaming analytics service that...
- Dataflow documentation - A fully-managed service for transforming and enriching data...
- Run pipelines with Dataflow and Java - Interactive Tutorial
- Run pipelines with Dataflow and Python

- Create Job: Click on "Create job".
- Select Template: Choose "Create job from template".
- Choose WordCount Template: Select "WordCount" from the template dropdown menu.
- Choose the input text file to perform WordCount.

The screenshot shows two overlapping windows. The main window is titled 'Create job from template' and has a sidebar with various icons. It contains fields for 'Job name' (set to 'wordcount'), 'Regional endpoint' (set to 'us-central1 (Iowa)'), 'Dataflow template' (set to 'Word Count'), and 'Input file(s) in Cloud Storage' (set to 'gs:// wordcount-input/Word-count.txt'). Below these are sections for 'Required Parameters' and 'Encryption'. The second window, titled 'Select object', is overlaid on the main window. It has a search bar with 'wordcount-input' and a results list containing 'Word-count.txt'. At the bottom of the 'Select object' window are 'SELECT' and 'CANCEL' buttons.

- Configure Parameters: Configure the template parameters (e.g., input and output locations).

This screenshot shows the 'Create job from template' interface with several configuration steps visible. In the 'Output Cloud Storage file prefix' field, 'gs:// wordcount-output1/wordcount-output' is entered. In the 'Temporary location' field, 'gs:// wordcount-output1/wctemp' is entered. Under 'Encryption', the 'Google-managed encryption key' option is selected. A section for 'Dataflow Prime' is present at the bottom, with a checkbox for 'Enable Dataflow Prime' and a detailed description below it.

- Here we customized the optional parameters.

Google Cloud naidu-prasanth dataflow

Create job from template

Machine types for common workloads, optimized for cost and flexibility

Series: E2

Machine type: e2-medium (2 vCPU, 1 core, 4 GB memory)

vCPU	1-2 vCPU (1 shared core)	Memory
		4 GB

Service account email: Compute Engine default service account

- Run the Job: Click "Run job".

JOB GRAPH EXECUTION DETAILS JOB METRICS COST RECOMMENDATIONS

Job steps view: Graph view

Job name: wordcount

Job ID: 2024-04-22_00_06_33-6929683504585203680

Job type: Batch

Job status: Running

SDK version: Apache Beam SDK for Java 2.54.0

Job region: us-central1

Worker location: us-central1

Current workers: 1

Latest worker status: Worker pool started.

Start time: April 22, 2024 at 12:36:36 PM

- Here we can see that, the Dataflow job get succeeded.

JOB GRAPH EXECUTION DETAILS JOB METRICS COST RECOMMENDATIONS

Job steps view: Graph view

Job name: wordcount

Job ID: 2024-04-22_00_06_33-6929683504585203680

Job type: Batch

Job status: Running

SDK version: Apache Beam SDK for Java 2.54.0

Job region: us-central1

Worker location: us-central1

Current workers: -

Latest worker status: Stopping worker pool.

Start time: April 22, 2024 at 12:36:36 PM

- We can see the output of the above Dataflow job in “wordcount-output1” bucket.

The screenshot shows the 'Bucket details' page for 'wordcount-output1'. The bucket's location is 'asia-south1 (Mumbai)', storage class is 'Standard', public access is 'Not public', and protection is 'Soft Delete'. The 'OBJECTS' tab is selected, showing a list of objects. One object, 'wordcount-output-00000-of-00001', is listed with a size of 3.3 KB, type 'text/plain', and creation date of Apr 22, 2024, 12:40:14 PM.

The screenshot shows the 'Object details' page for 'wordcount-output-00000-of-00001'. It provides an overview of the object, including its type (text/plain), size (3.3 KB), creation date (Apr 22, 2024, 12:40:14 PM), and storage class (Standard). It also shows the object's URL: <https://storage.cloud.google.com/wordcount-output1/wordcount-output-00000-of-00001>.

- This is how the output looks like.

```

she: 8
yesterday: 4
mean: 4
didn't: 4
feeder: 4
Japanese: 4
nor: 4
from: 16
said: 4
been: 4
it's: 4
tipping: 4
The: 32
Africa: 4
created: 4
shrink: 4
outside: 4
lost: 4
book: 4
wipes: 4
crackers: 4
Please: 4
could: 8
whether: 4
Weather: 4
depended: 4
find: 4
in: 24
consider: 4
into: 4
preoccupied: 4
enough: 8
phone: 4
doll: 4
hopes: 4
feet: 4
stop: 8

```

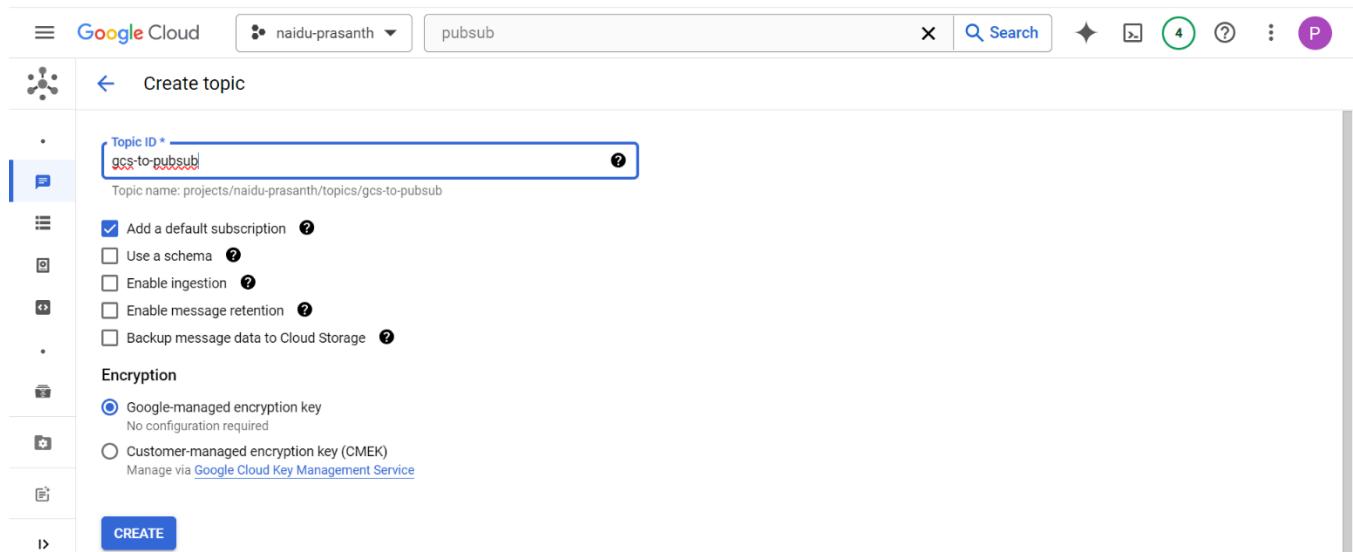
Step 6: Going to run the Pub/Sub Pipeline (Additional work)

Now we are going to connect the wordcount output to Pub/Sub.

Pub/Sub, short for publish-subscribe, is an asynchronous messaging system used for communication between different parts of an application.

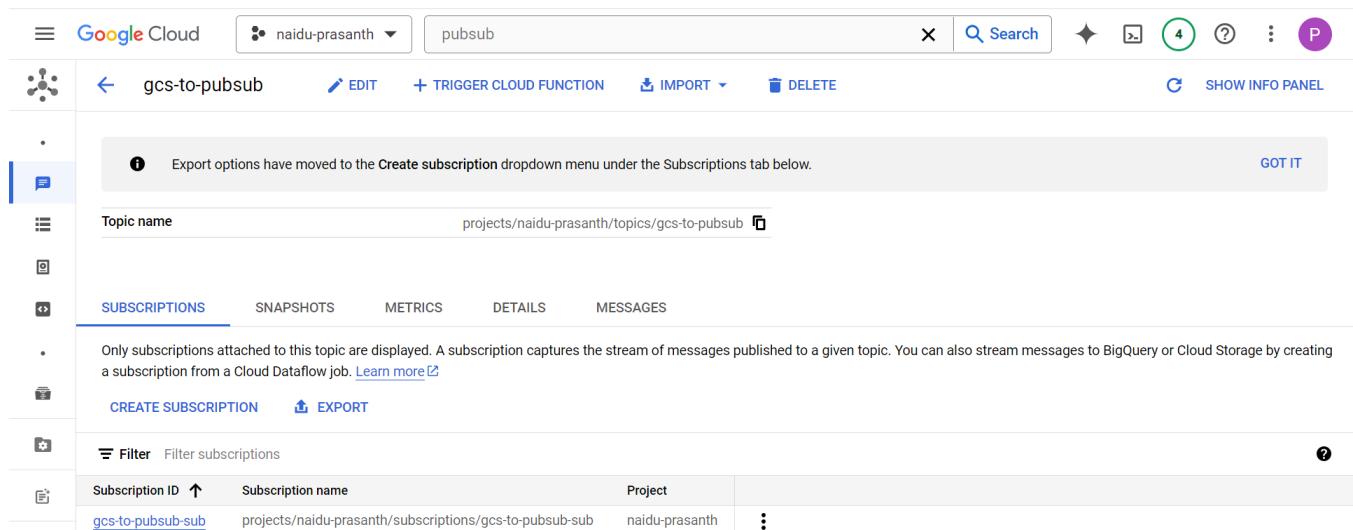
Pub/Sub provides a flexible and reliable way for different parts of a distributed system to communicate asynchronously, without tight coupling or waiting times.

- For that, first we need to create a “Topic” and “Subscription” in Pub/Sub



The screenshot shows the 'Create topic' page in the Google Cloud interface. The top navigation bar includes 'Google Cloud', a user dropdown for 'naidu-prasanth', a search bar with 'pubsub', and various icons. The main form has a 'Topic ID' field containing 'gcs-to-pubsub' with a red box highlighting it. Below the field, a tooltip says 'Topic name: projects/naidu-prasanth/topics/gcs-to-pubsub'. There are several configuration options: a checked checkbox for 'Add a default subscription', and unchecked checkboxes for 'Use a schema', 'Enable ingestion', 'Enable message retention', and 'Backup message data to Cloud Storage'. Under 'Encryption', the 'Google-managed encryption key' option is selected. A 'CREATE' button is at the bottom. The left sidebar has a tree view with 'Topics' selected.

- Here our Topic ID is “gcs-to-pubsub” and Subscription ID is “gcs-to-pubsub-sub” and we can verify that below.



The screenshot shows the 'Topics' page in the Google Cloud interface. The top navigation bar includes 'Google Cloud', a user dropdown for 'naidu-prasanth', a search bar with 'pubsub', and various icons. The main area shows a topic named 'gcs-to-pubsub'. Below the topic name, there are buttons for 'EDIT', '+ TRIGGER CLOUD FUNCTION', 'IMPORT', and 'DELETE'. A 'SHOW INFO PANEL' button is also present. A message box says 'Export options have moved to the Create subscription dropdown menu under the Subscriptions tab below.' with a 'GOT IT' button. The 'SUBSCRIPTIONS' tab is active, showing one subscription: 'gcs-to-pubsub-sub'. Other tabs include 'SNAPSHOTS', 'METRICS', 'DETAILS', and 'MESSAGES'. At the bottom, there's a 'CREATE SUBSCRIPTION' button and a 'EXPORT' button. A 'Filter' dropdown is shown, and a table lists the subscription details: 'Subscription ID' is 'gcs-to-pubsub-sub', 'Subscription name' is 'projects/naidu-prasanth/subscriptions/gcs-to-pubsub-sub', and 'Project' is 'naidu-prasanth'. A three-dot menu icon is at the end of the row.

- Now we have to go to the output file of the wordcount and click on “Export to cloud Pub/Sub”

Bucket details for wordcount-output1

Name	Type	Created	Storage class	Last modified
wctemp/	Folder	—	—	—
wordcount-output-00000-of-00001	text/plain	Apr 22, 2024, 12:40:14 PM	Standard	Apr 22, 2024, 12:40:14 PM

Context menu options include: Download, Copy Authenticated URL, Copy gsutil URI, Edit metadata, Edit access, Edit retention, Copy, Move, Rename, Export to Cloud Pub/Sub, and Scan with Sensitive Data Protection.

- In next step it will automatically redirect to Dataflow service and it will select the Dataflow template by default.

Create job from template

Create a Dataflow job using template "Cloud Storage Text File to Pub/Sub (Batch)"

Job name *
gcs-to-pubsub

Must be unique among running jobs

Regional endpoint *
us-central1 (Iowa)

Choose a Dataflow regional endpoint to deploy worker instances and store job metadata. You can optionally deploy worker instances to any available Google Cloud region or zone by using the worker region or worker zone parameters. Job metadata is always stored in the Dataflow regional endpoint. [Learn more](#)

Dataflow template *
Cloud Storage Text File to Pub/Sub (Batch)

This template creates a batch pipeline that reads records from text files stored in Cloud Storage and publishes them to a Pub/Sub topic. The template can be used to publish

Additional information

Once you run this job, you can view its status on the next screen to confirm that no errors occurred and all data exported successfully. You can also stop it at any time. The costs of this batch pipeline will depend on the amount of data you will process.

SHOW MORE

```

graph TD
    A[Read Text Data] --> B[Write to PubSub]
  
```

- Now we have to give the Pub/Sub topic which we created earlier and give the temporary location.

Required Parameters

Cloud Storage Input File(s) * [BROWSE](#)
Path of the file pattern glob to read from. (Example: gs://your-bucket/path/*.txt)

Output Pub/Sub topic * [▼](#)

Temporary location * [BROWSE](#)
Path and filename prefix for writing temporary files. Ex: gs://your-bucket/temp

Encryption

Google-managed encryption key
No configuration required

Customer-managed encryption key (CMK)
Manage via [Google Cloud Key Management Service](#)

Dataflow Prime

- Now customize the optional parameters.

Machine types for common workloads, optimized for cost and flexibility

Series [▼](#)
CPU platform selection based on availability

Machine type [▼](#)

	vCPU 1-2 vCPU (1 shared core)	Memory 4 GB
--	----------------------------------	----------------

Service account email [▼](#)
The email address of the service account to run the job as

- Click on run job and verify whether Dataflow job get succeeded or not.

Job info

Job name	gcs-to-pubsub
Job ID	2024-04-22_00_16_57-3452508192825604832
Job type	Batch
Job status	Running
SDK version	Apache Beam SDK for Java 2.54.0
A newer version of the SDK family exists and updating is recommended. Learn more	
Job region	us-central1
Worker location	us-central1
Current workers	1
Latest worker status	Worker pool started.
Start time	April 22, 2024 at 12:46:58 PM GMT+5

JOB GRAPH

Job steps view [▼](#)

CLEAR SELECTION

Read Text Data
Succeeded
7 sec
2 of 2 stages succeeded

Write to PubSub
Succeeded
0 sec
1 of 1 stage succeeded

Logs [SHOW](#)

- We can verify it by navigating to Pub/Sub subscription and Pull the message.

The screenshot shows the Google Cloud Pub/Sub interface. At the top, there's a navigation bar with 'Google Cloud' and a dropdown for 'naidu-prasanth'. The search bar contains 'pubsub'. On the right side of the header are various icons for search, refresh, and account management. Below the header, the main content area shows a subscription named 'gcs-to-pubsub-sub'. The subscription details are as follows:

- Subscription name: projects/naidu-prasanth/subscriptions/gcs-to-pubsub-sub
- Subscription state: active (indicated by a green checkmark)
- Topic name: projects/naidu-prasanth/topics/gcs-to-pubsub

Below the subscription details, there are three tabs: 'METRICS', 'DETAILS', and 'MESSAGES'. The 'MESSAGES' tab is selected, showing the following instructions: "Click Pull to view messages and temporarily delay message delivery to other subscribers. Select Enable ACK messages and then click ACK next to the message to permanently prevent message delivery to other subscribers." There are also buttons for 'PULL' and 'Enable ack messages'. A 'Filter' button is available to 'Filter messages'. The message list table has columns: Publish time, Attribute keys, Message body, Ordering key, and Ack. The 'Ack' column has an upward arrow icon. The table currently displays no messages.

- Finally, we can verify the output.

This screenshot shows the same Google Cloud Pub/Sub interface as the previous one, but now it displays a list of messages in the 'MESSAGES' tab. The subscription details remain the same:

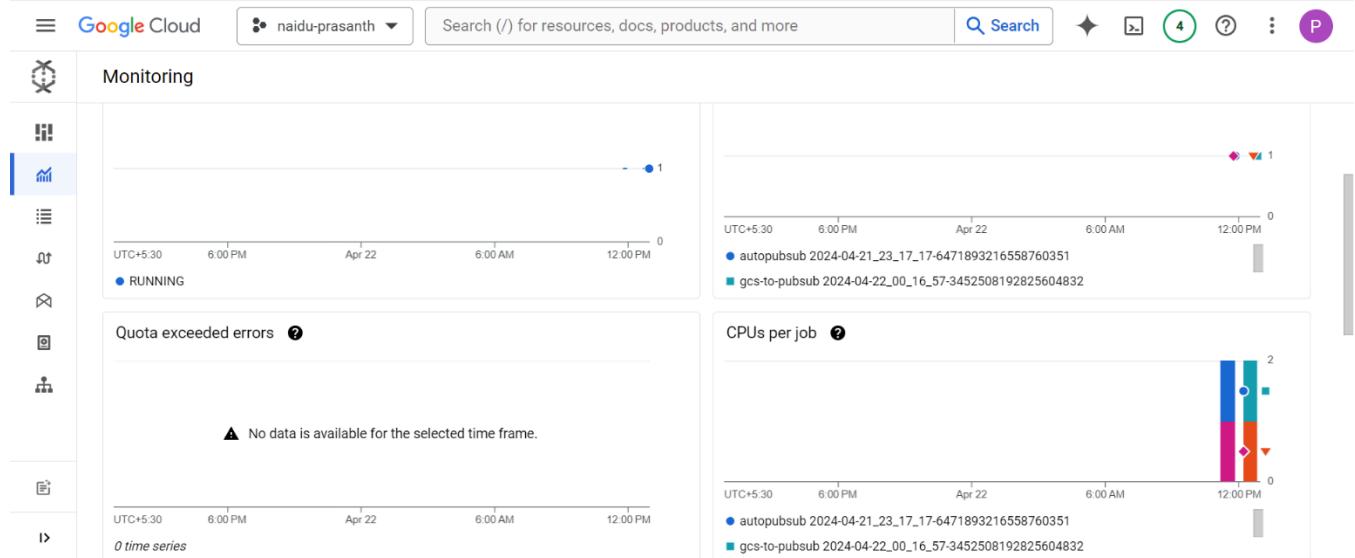
- Subscription name: projects/naidu-prasanth/subscriptions/gcs-to-pubsub-sub
- Subscription state: active
- Topic name: projects/naidu-prasanth/topics/gcs-to-pubsub

The message list table is populated with the following data:

Publish time	Attribute keys	Message body	Ordering key	Ack
Apr 22, 2024, 12:49:45 PM	—	months: 4	—	Deadline exceeded
Apr 22, 2024, 12:49:45 PM	—	past: 4	—	Deadline exceeded
Apr 22, 2024, 12:49:45 PM	—	buns: 4	—	Deadline exceeded
Apr 22, 2024, 12:49:45 PM	—	would: 8	—	Deadline exceeded
Apr 22, 2024, 12:49:45 PM	—	way: 8	—	Deadline exceeded
Apr 22, 2024, 12:49:45 PM	—	Toddlers: 4	—	Deadline exceeded

Step 7: Monitoring Your Job

- Access Dataflow Section: In the navigation bar, select Dataflow.
- View Jobs List: You'll see a list of your Dataflow jobs.
- View Job Details: Click on the specific WordCount job to view details like status, logs, and metrics.



Deleting buckets and Pub/Sub topics after use serves several important purposes:

Resource Management and Cost Optimization:

- **Buckets:** When you delete a **Cloud Storage bucket**, it frees up the resources associated with it. This helps manage your cloud resources efficiently and reduces costs.
- **Pub/Sub Topics:** Deleting unused Pub/Sub topics ensures that you are not paying for unnecessary message handling. It's a way to optimize costs by cleaning up resources that are no longer needed.

Security and Data Privacy:

- **Buckets:** Deleting buckets prevents unauthorized access to data. Even if a bucket is no longer actively used, it may still contain sensitive information. Deleting it ensures that no one can accidentally or intentionally access the data.
- **Pub/Sub Topics:** Removing unused topics reduces the attack surface. If a topic is no longer in use, it's best to delete it to minimize potential security risks.

Maintaining a Clean Environment:

- **Buckets:** Regularly deleting unused buckets keeps your cloud environment tidy. It avoids clutter and makes it easier to manage and navigate your resources.
- **Pub/Sub Topics:** Similar to buckets, cleaning up unused topics ensures a streamlined Pub/Sub setup.

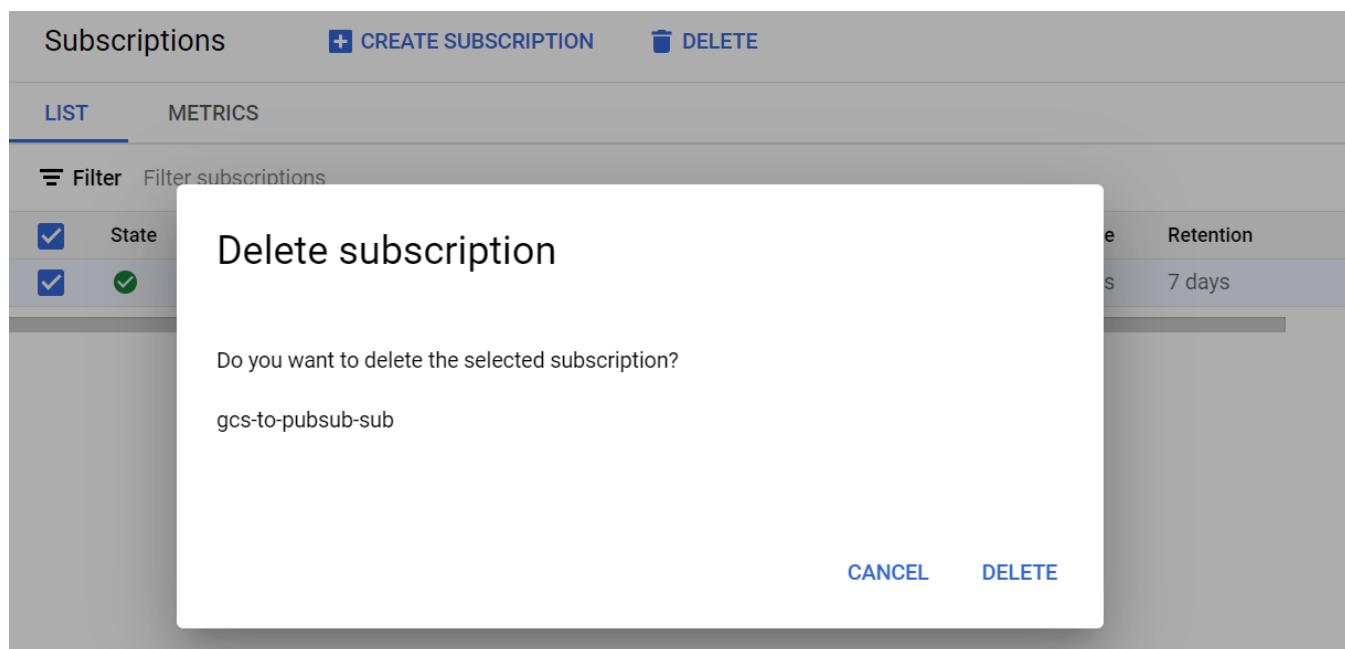
Avoiding Confusion and Accidental Usage:

- **Buckets:** If you have multiple buckets, leaving unused ones can lead to confusion. Developers might accidentally use the wrong bucket, causing unexpected behavior.
- **Pub/Sub Topics:** Unused topics can also confuse developers or result in unintended message delivery.

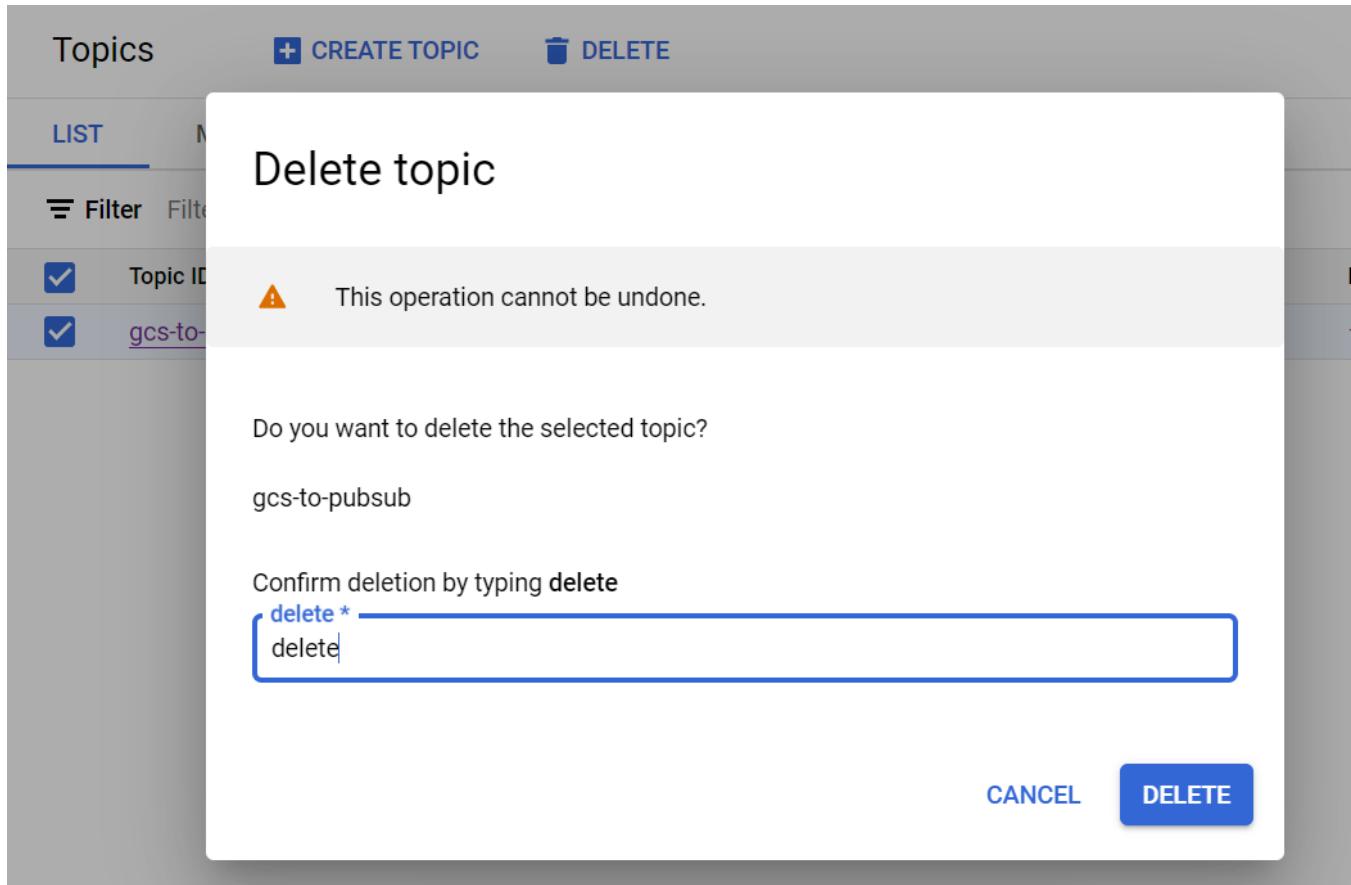
Best Practices:

- **Buckets:** Following best practices, such as deleting buckets after their lifecycle ends, ensures a well-organized and secure cloud storage environment.
- **Pub/Sub Topics:** Regularly reviewing and deleting topics aligns with good resource management practices.

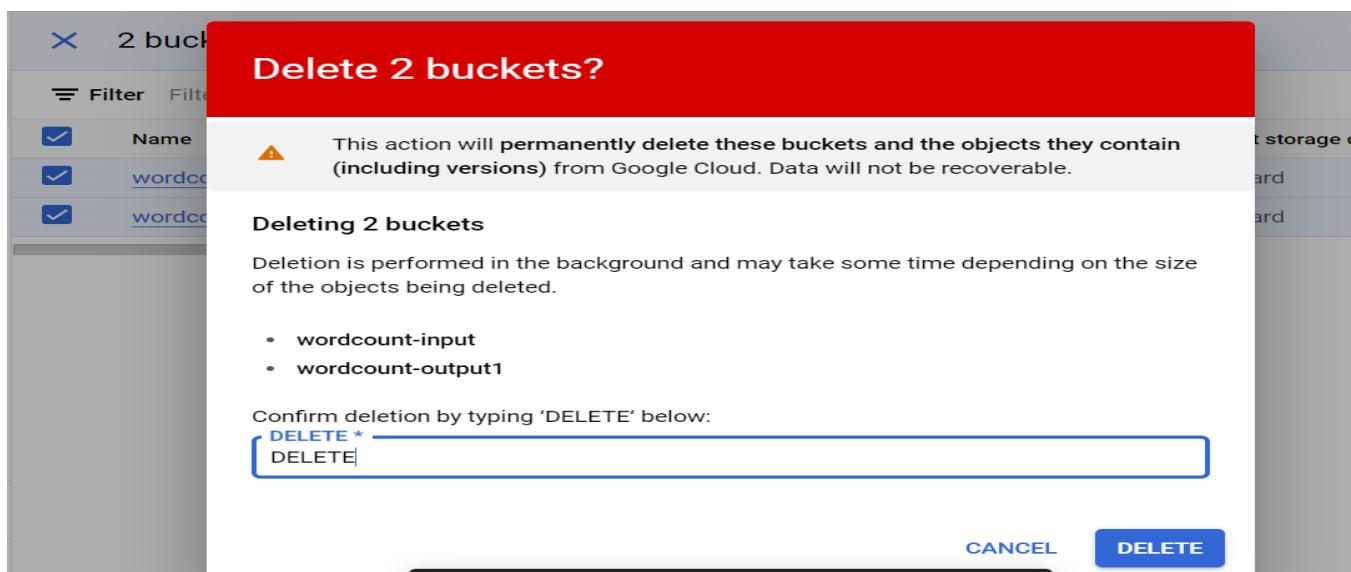
Deletion of subscription in pub/sub:-



Deletion of topic in pub/sub:-



Deletion of buckets:-



Using command shell

A **bucket** is a fundamental concept in **Google Cloud Storage**. Here are the key points:

Purpose of Buckets:

- Buckets serve as the **basic containers** for storing data in Cloud Storage.
- **Everything** you store in Cloud Storage must be contained within a bucket.
- Unlike directories or folders, **buckets cannot be nested** inside one another.
- There is **no limit** to the number of buckets you can have in a project or location.

Inserting file into bucket wordcount_data

The screenshot shows the 'Bucket details' page for the 'wordcount_data' bucket. The bucket is located in 'asia-south1 (Mumbai)' with 'Standard' storage class and 'Not public' access. The 'OBJECTS' tab is selected, showing a folder browser. Inside the 'wordcount_data' folder, there is a single file named 'constitution.txt'. The file details are: Size 11.1 KB, Type text/plain, and Created Apr 24, 2024, 9:59. There are also buttons for UPLOAD FILES, UPLOAD FOLDER, CREATE FOLDER, TRANSFER DATA, MANAGE HOLDS, EDIT RETENTION, DOWNLOAD, and DELETE.

Created output bucket wordcount_data_op for storing output data.

The screenshot shows the 'Buckets' page. It lists two buckets: 'wordcount_data' and 'wordcount_data_op'. Both buckets were created on Apr 24, 2024, at 9:58:26 AM, are located in 'Region' type, and have 'Standard' as their default storage class. The 'wordcount_data' bucket has a last modified time of Apr 24, 2024, 9:58:26 AM, while the 'wordcount_data_op' bucket has a last modified time of Apr 24, 2024, 9:59:22 AM. The page includes filters, a create button, and refresh, go to path, and learn links.

```
pip3 install virtualenv
```

- This part of the command uses `pip3`, which is a package manager for Python.
- `pip3` is used to install Python packages and libraries.
- `virtualenv` is a tool that creates isolated Python environments.

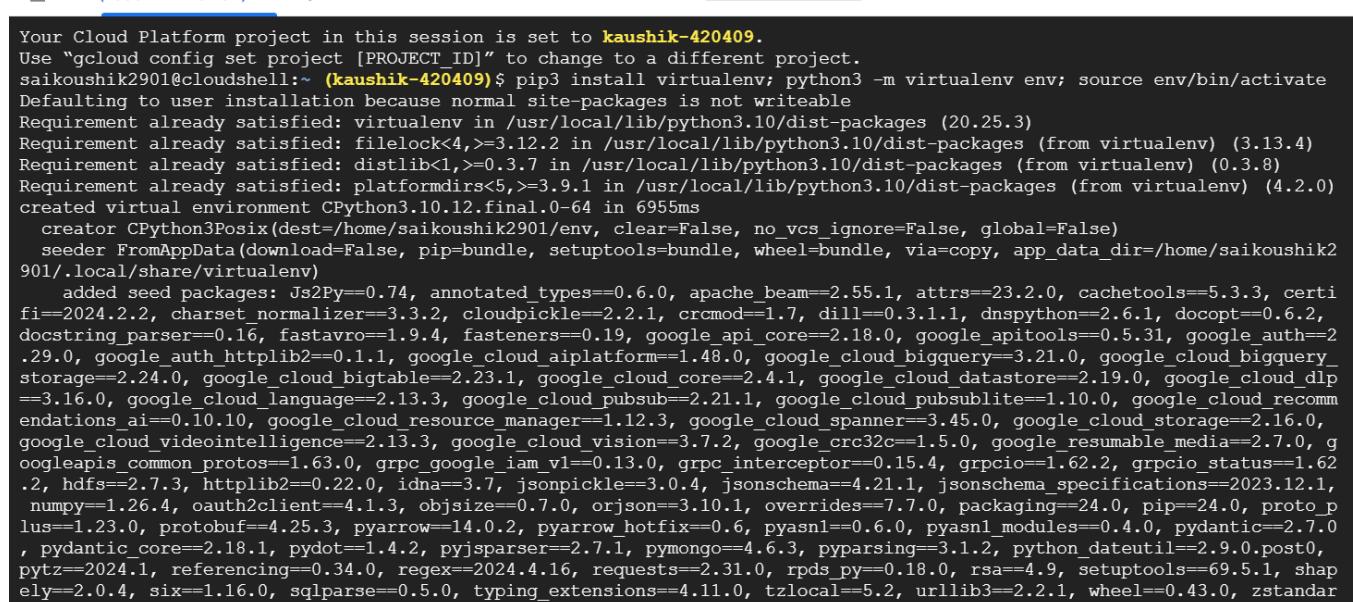
- When you run this command, it installs the `virtualenv` package globally on your system.

```
python3 -m virtualenv env:
```

- This part of the command creates a new **virtual environment** named `env`.
- A virtual environment is a self-contained directory where you can install Python packages independently of the system-wide Python installation.
- The `-m` flag specifies that we want to run the `virtualenv` module as a script.
- After executing this command, you'll have a new directory called `nv` containing a separate Python environment.

```
source env/bin/activate:
```

- This part of the command **activates** the virtual environment.
- When you activate the environment, it modifies your shell's environment variables to use the Python interpreter and packages from the virtual environment.
- You'll notice that your shell prompt changes to indicate that you are now working within the `env` virtual environment.
- This step is essential because it ensures that any Python packages you install or scripts you run will use the isolated environment rather than the system-wide Python installation.



```
Your Cloud Platform project in this session is set to kaushik-420409.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
saikoushik2901@cloudshell:~ (kaushik-420409)$ pip3 install virtualenv; python3 -m virtualenv env; source env/bin/activate
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: virtualenv in /usr/local/lib/python3.10/dist-packages (20.25.3)
Requirement already satisfied: filelock<4,>=3.12.2 in /usr/local/lib/python3.10/dist-packages (from virtualenv) (3.13.4)
Requirement already satisfied: distlib<1,>=0.3.7 in /usr/local/lib/python3.10/dist-packages (from virtualenv) (0.3.8)
Requirement already satisfied: platformdirs<5,>=3.9.1 in /usr/local/lib/python3.10/dist-packages (from virtualenv) (4.2.0)
created virtual environment CPython3.10.12.final.0-64 in 695ms
  creator CPython3Posix(dest=/home/saikoushik2901/env, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/saikoushik2901/.local/share/virtualenv)
  added seed packages: Js2Py==0.74, annotated_types==0.6.0, apache_beam==2.55.1, attrs==23.2.0, cachetools==5.3.3, certifi==2024.2.2, charset_normalizer==3.3.2,云pickle==2.2.1, crcmod==1.7, dill==0.3.1.1, dnsPYTHON==2.6.1, docopt==0.6.2, docstring_parser==0.16, fastavro==1.9.4, fasteners==0.19, google_api_core==2.18.0, google_apitools==0.5.31, google_auth==2.29.0, google_auth_httplib2==0.1.1, google_cloud_aiplatform==1.48.0, google_cloud_bigquery==3.21.0, google_cloud_bigquery_storage==2.24.0, google_cloud_bigtable==2.23.1, google_cloud_core==2.4.1, google_cloud_datastore==2.19.0, google_cloud_dlp==3.16.0, google_cloud_language==2.13.3, google_cloud_pubsub==2.21.1, google_cloud_pubsublite==1.10.0, google_cloud_recommender==2.10.10, google_cloud_resource_manager==1.12.3, google_cloud_spanner==3.45.0, google_cloud_storage==2.16.0, google_cloud_videointelligence==2.13.3, google_cloud_vision==3.7.2, google_crc32c==1.5.0, google_resumable_media==2.7.0, googleapis_common_protos==1.63.0, grpc_google iam_v1==0.13.0, grpc_interceptor==0.15.4, grpcio==1.62.2, grpcio_status==1.62.2, hdfs==2.7.3, httpplib2==0.22.0, idna==3.7, jsonpickle==3.0.4, jsonschema==4.21.1, jsonschema_specifications==2023.12.1, numpy==1.26.4, oauth2client==4.1.3, objsize==0.7.0, orjson==3.10.1, overrides==7.7.0, packaging==24.0, pip==24.0, proto_pbus==1.23.0, protobuf==4.25.3, pyarrow==14.0.2, pyarrow_hotfix==0.6, pyasn1==0.6.0, pyasn1_modules==0.4.0, pydantic==2.7.0, pydantic_core==2.18.1, pydot==1.4.2, pyjpsparser==2.7.1, pymongo==4.6.3, pyparsing==3.1.2, python_dateutil==2.9.0.post0, pytz==2024.1, referencing==0.34.0, regex==2024.4.16, requests==2.31.0, rpds_py==0.18.0, rsa==4.9, setuptools==69.5.1, shapely==2.0.4, six==1.16.0, sqlparse==0.5.0, typing_extensions==4.11.0, tzlocal==5.2, urllib3==2.2.1, wheel==0.43.0, zstandard==0.15.0
```

`pip3`:

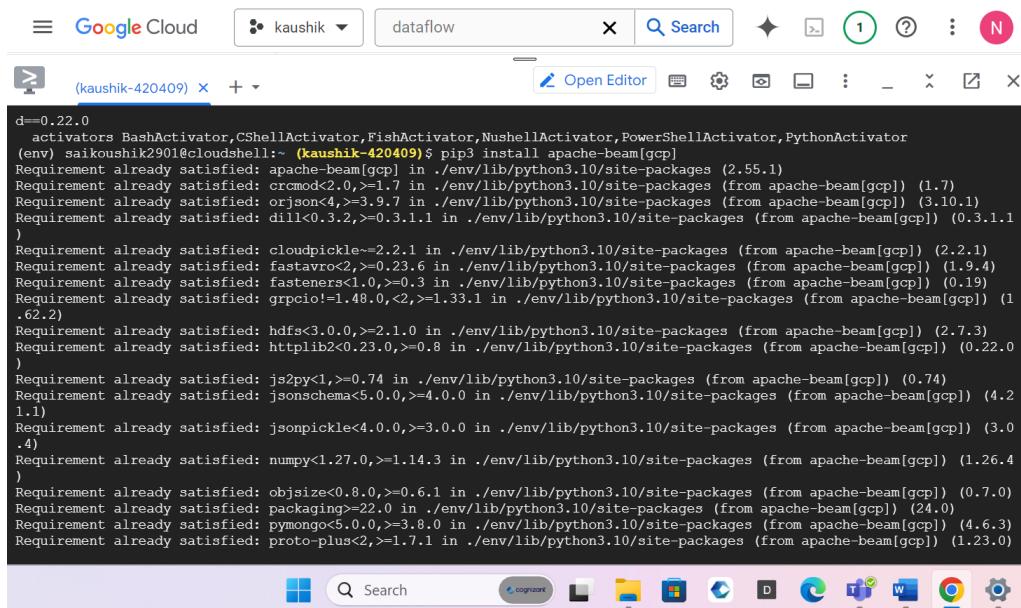
- `pip3` is a package manager for Python.
- It allows you to **install, upgrade, and manage** Python packages and libraries.

install:

- The `install` command is used to **install** Python packages.
- When you run `pip3 install <package-name>`, it fetches the specified package from the Python Package Index (PyPI) and installs it on your system.

apache-beam:

- Apache Beam is an open-source project that provides a unified programming model for both batch and streaming data processing.
- It enables efficient execution across diverse distributed execution engines (such as Apache Flink, Apache Spark, and Google Cloud Dataflow) and provides extensibility points for connecting to different technologies and user communities.
- In the context of Python, `apache-beam` refers to the Python SDK for Apache Beam.



The screenshot shows a terminal window within a Google Cloud Dataflow interface. The terminal title is '(kaushik-420409)'. The command entered is `pip3 install apache-beam[gcp]`. The output of the command is displayed, showing the installation of various dependencies. The output starts with `d==0.22.0` and lists requirements for BashActivator, CShellActivator, FishActivator, NoshellActivator, PowerShellActivator, PythonActivator, and others, including apache-beam[gcp], crcmod, orjson, dill, cloudpickle, fastavro, fasteners, grpcio, hdfs, httplib2, js2py<1, jsonschema, numpy, objsize, packaging, pymongo, proto-plus, and several versions of Python libraries like requests, six, and typing.

```
d==0.22.0
activators BashActivator,CShellActivator,FishActivator,NoshellActivator,PowerShellActivator,PythonActivator
(env) saikoushik2901@cloudshell:~ (kaushik-420409)$ pip3 install apache-beam[gcp]
Requirement already satisfied: apache-beam[gcp] in ./env/lib/python3.10/site-packages (2.55.1)
Requirement already satisfied: crcmod<2.0,>=1.7 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (1.7)
Requirement already satisfied: orjson<4,>=3.9.7 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (3.10.1)
Requirement already satisfied: dill<0.3.2,>=0.3.1.1 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (0.3.1.1)
Requirement already satisfied: cloudpickle<=2.2.1 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (2.2.1)
Requirement already satisfied: fastavro<2,>=0.23.6 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (1.9.4)
Requirement already satisfied: fasteners<1.0,>=0.3 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (0.19)
Requirement already satisfied: grpcio!=1.48.0,<2,>=1.33.1 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (1.62.2)
Requirement already satisfied: hdfs<3.0.0,>=2.1.0 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (2.7.3)
Requirement already satisfied: httplib2<0.23.0,>=0.8 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (0.22.0)
Requirement already satisfied: js2py<1,>=0.74 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (0.74)
Requirement already satisfied: jsonschema<5.0.0,>=4.0.0 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (4.2.1.1)
Requirement already satisfied: jsonpickle<4.0.0,>=3.0.0 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (3.0.4)
Requirement already satisfied: numpy<1.27.0,>=1.14.3 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (1.26.4)
Requirement already satisfied: objsize<0.8.0,>=0.6.1 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (0.7.0)
Requirement already satisfied: packaging<22.0,>=22.0 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (24.0)
Requirement already satisfied: pymongo<5.0.0,>=3.8.0 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (4.6.3)
Requirement already satisfied: proto-plus<2,>=1.7.1 in ./env/lib/python3.10/site-packages (from apache-beam[gcp]) (1.23.0)
```

```
python3 -m apache_beam.examples.wordcount:
```

- This part of the command runs a Python module using the `-m` flag.
- The module being executed is `apache_beam.examples.wordcount`.

The `wordcount` example is a common introductory example in Apache Beam, demonstrating how to count words in a text document.

1. --region us-central1:

- Specifies the **region** where the Dataflow job will run.
- In this case, it's set to `us-central1`, which corresponds to the central United States.

2. --input gs://wordcount_data/constitution.txt:

- Specifies the **input file** for the word count job.

- The input file is located at `gs://wordcount_data/constitution.txt`, which is a Google Cloud Storage (GCS) path.
 - The job will read the content of this file to perform word counting.
- 3. `--output gs://wordcount_data_op:`**
- Specifies the **output location** for the word count results.
 - The output will be written to a GCS bucket at `gs://wordcount_data_op`.
- 4. `--runner DataflowRunner:`**
- Specifies the **execution runner** for the job.
 - In this case, it's set to `DataflowRunner`, which means the job will run on Google Cloud Dataflow.
 - Dataflow is a managed service for executing Apache Beam pipelines.
- 5. `--project kaushik-420409:`**
- Specifies the **Google Cloud project ID** where the Dataflow job will run.
 - Replace `kaushik-420409` with your actual project ID.
- 6. `--temp_location gs://wordcount_data_op/temp:`**
- Specifies the **temporary location** for storing intermediate data during the job execution.
 - The temporary files generated by the job will be stored in the GCS bucket at `gs://wordcount_data_op/temp`.

```
(env) saikoushik2901@cloudshell:~ (kaushik-420409)$ python3 -m apache_beam.examples.wordcount --region us-central1
--input gs://wordcount_data/constitution.txt --output gs://wordcount_data_op --runner DataflowRunner
--project kaushik-420409 --temp_location gs://wordcount_data_op/temp/
INFO:apache_beam.internal.gcp.auth:Setting socket default timeout to 60 seconds.
INFO:apache_beam.internal.gcp.auth:socket default timeout is 60.0 seconds.
INFO:apache_beam.runners.dataflow.dataflow_runner:Pipeline has additional dependencies to be installed in SDK worker container, consider using the SDK container image pre-building workflow to avoid repetitive installations. Learn more on https://cloud.google.com/dataflow/docs/guides/using-custom-containers#prebuild
INFO:root:Using provided Python SDK container image: gcr.io/cloud-dataflow/v1beta3/beam_python3.10_sdk:2.55.1
INFO:root:Python SDK container image set to "gcr.io/cloud-dataflow/v1beta3/beam_python3.10_sdk:2.55.1" for Docker environment
INFO:apache_beam.runners.portability.fn_api_runner.translations:===== <function pack_combiners at 0x7a83c33a3b50> =====
INFO:apache_beam.runners.portability.fn_api_runner.translations:===== <function sort_stages at 0x7a83c33ac3a0> =====
INFO:apache_beam.runners.dataflow.internal.apiclient:Starting GCS upload to gs://wordcount_data_op/temp/beamapp-saikoushik2901-0424044656-754472-juju4d3x.1713934016.754827/pickled_main_session...
INFO:apache_beam.runners.dataflow.internal.apiclient:Completed GCS upload to gs://wordcount_data_op/temp/beamapp-saikoushik2901-0424044656-754472-juju4d3x.1713934016.754827/pickled_main_session in 1 seconds.
INFO:apache_beam.runners.dataflow.internal.apiclient:Starting GCS upload to gs://wordcount_data_op/temp/beamapp-saikoushik2901-0424044656-754472-juju4d3x.1713934016.754827/pipeline.pb...
INFO:apache_beam.runners.dataflow.internal.apiclient:Completed GCS upload to gs://wordcount_data_op/temp/beamapp-saikoushik2901-0424044656-754472-juju4d3x.1713934016.754827/pipeline.pb...
```

The screenshot shows the Google Cloud Shell interface. At the top, there's a navigation bar with 'Google Cloud' and a dropdown for 'kaushik'. Below it is a search bar and a toolbar with icons for 'CREATE', 'REFRESH', 'GO TO PATH', and 'LEARN'. The main area is a terminal window titled '(kaushik-420409)'. It displays log output from a Dataflow job, including messages about writing to a side input and finalizing the job. The terminal ends with the command 'saikushik2901@cloudshell:~ (kaushik-420409)\$'. Below the terminal is a taskbar with various icons.

```

write/WriteImpl/FinalizeWrite/View-python_side_input1
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-04-24T04:59:38.376Z: JOB_MESSAGE_BASIC: Executing operation Write/W
rite/WriteImpl/PreFinalize/View-python_side_input1
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-04-24T04:59:39.479Z: JOB_MESSAGE_BASIC: Finished operation Write/Wr
ite/WriteImpl/FinalizeWrite/View-python_side_input1
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-04-24T04:59:39.479Z: JOB_MESSAGE_BASIC: Finished operation Write/Wr
ite/WriteImpl/PreFinalize/View-python_side_input1
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-04-24T04:59:39.581Z: JOB_MESSAGE_BASIC: Executing operation Write/W
rite/WriteImpl/PreFinalize
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-04-24T05:00:31.910Z: JOB_MESSAGE_BASIC: Finished operation Write/Wr
ite/WriteImpl/PreFinalize
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-04-24T05:00:32.027Z: JOB_MESSAGE_BASIC: Executing operation Write/W
rite/WriteImpl/View-python_side_input2
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-04-24T05:00:33.153Z: JOB_MESSAGE_BASIC: Finished operation Write/Wr
ite/WriteImpl/View-python_side_input2
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-04-24T05:00:33.312Z: JOB_MESSAGE_BASIC: Executing operation Write/W
rite/WriteImpl/FinalizeWrite
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-04-24T05:01:02.682Z: JOB_MESSAGE_BASIC: Finished operation Write/Wr
ite/WriteImpl/FinalizeWrite
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-04-24T05:01:07.540Z: JOB_MESSAGE_BASIC: Stopping worker pool...
INFO:apache_beam.runners.dataflow.dataflow_runner:2024-04-24T05:01:47.019Z: JOB_MESSAGE_BASIC: Worker pool stopped.
INFO:apache_beam.runners.dataflow.dataflow_runner:Job 2024-04-23_21_54_29-1481249222811756739 is in state JOB_STATE_DONE
(env) saikushik2901@cloudshell:~ (kaushik-420409)$

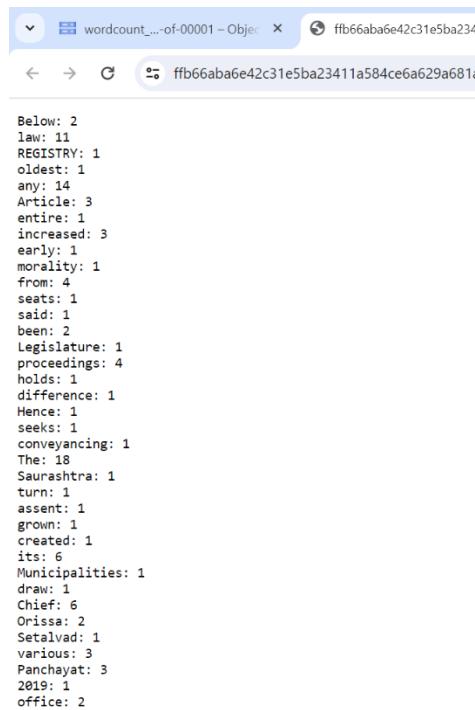
```

Now we can observe the result in output bucket.

This screenshot shows the 'Bucket details' page for the 'wordcount_data_op' bucket. The left sidebar has 'Buckets' selected. The main area shows a 'Folder browser' with two folders: 'op/' and 'temp/'. There are buttons for 'UPLOAD FILES', 'UPLOAD FOLDER', 'CREATE FOLDER', 'TRANSFER DATA', 'MANAGE HOLDS', 'EDIT RETENTION', 'DOWNLOAD', and 'DELETE'. A filter bar at the bottom allows filtering by name prefix, type, and storage class. The results table shows the 'op/' folder with one entry: '-00000-of-00001'.

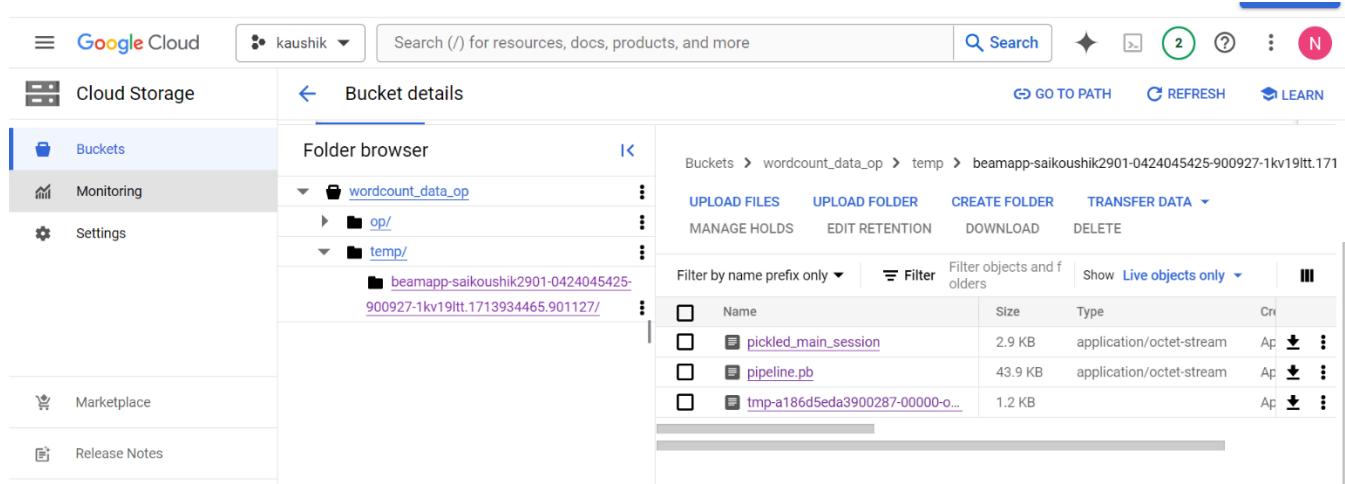
This screenshot shows the 'Bucket details' page for the 'op/' folder within the 'wordcount_data_op' bucket. The left sidebar has 'Monitoring' selected. The main area shows a 'Folder browser' with two sub-folders: 'op/' and 'temp/'. There are buttons for 'UPLOAD FILES', 'UPLOAD FOLDER', 'CREATE FOLDER', 'TRANSFER DATA', 'MANAGE HOLDS', 'EDIT RETENTION', 'DOWNLOAD', and 'DELETE'. A filter bar at the bottom allows filtering by name prefix, type, and storage class. The results table shows the 'op/' folder with one entry: '-00000-of-00001'.

It's the output bucket

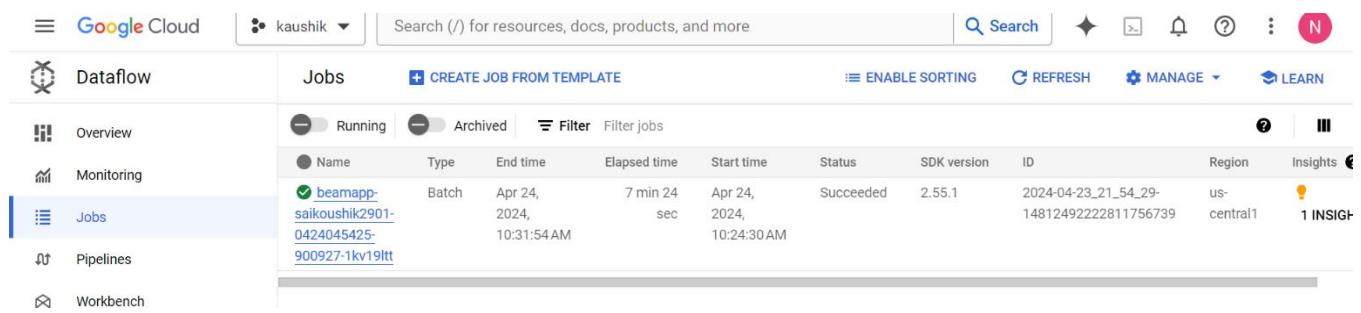


The screenshot shows a browser window with the URL `ff66aba6e42c31e5ba23411a584ce6a629a681a`. The page displays a list of words and their counts from a word count operation. The counts are as follows:

```
Below: 2
law: 11
REGISTRY: 1
oldest: 1
any: 14
Article: 3
entire: 1
increased: 3
early: 1
morality: 1
from: 4
seats: 1
said: 1
been: 2
Legislature: 1
proceedings: 4
holds: 1
difference: 1
Hence: 1
seeks: 1
conveyancing: 1
The: 18
Saurashtra: 1
turn: 1
assent: 1
grown: 1
created: 1
its: 6
Municipalities: 1
draw: 1
Chief: 6
Orissa: 2
Setalvad: 1
various: 3
Panchayat: 3
2019: 1
office: 2
```



The screenshot shows the 'Bucket details' page for the 'wordcount_data_op' bucket. The left sidebar shows 'Cloud Storage' with 'Buckets' selected. The main area shows the 'temp' folder under 'wordcount_data_op'. The folder contains several files: 'beamapp-saikoushik2901-0424045425-900927-1kv19ltt.1713934465.901127/' (which is expanded), 'beamapp-saikoushik2901-0424045425-900927-1kv19ltt.1713934465.901127/_SUCCESS', 'beamapp-saikoushik2901-0424045425-900927-1kv19ltt.1713934465.901127/_LOG', 'beamapp-saikoushik2901-0424045425-900927-1kv19ltt.1713934465.901127/_PARTITIONED_LOG', 'beamapp-saikoushik2901-0424045425-900927-1kv19ltt.1713934465.901127/_PIPELINE_INFO', 'beamapp-saikoushik2901-0424045425-900927-1kv19ltt.1713934465.901127/_SESSION_INFO', 'beamapp-saikoushik2901-0424045425-900927-1kv19ltt.1713934465.901127/_TEMP', 'beamapp-saikoushik2901-0424045425-900927-1kv19ltt.1713934465.901127/_TEMP_PARTITIONED_LOG', 'beamapp-saikoushik2901-0424045425-900927-1kv19ltt.1713934465.901127/_TEMP_SESSION_INFO', and 'beamapp-saikoushik2901-0424045425-900927-1kv19ltt.1713934465.901127/_TEMP_TABLET_INFO'. The table below lists these files with columns for Name, Size, Type, and Actions.



The screenshot shows the 'Jobs' page in the Dataflow interface. The left sidebar shows 'Dataflow' with 'Jobs' selected. The main area shows a single job named 'beamapp-saikoushik2901-0424045425-900927-1kv19ltt'. The job details table includes columns for Name, Type, End time, Elapsed time, Start time, Status, SDK version, ID, Region, and Insights. The job status is 'Succeeded'.

Name	Type	End time	Elapsed time	Start time	Status	SDK version	ID	Region	Insights
beamapp-saikoushik2901-0424045425-900927-1kv19ltt	Batch	Apr 24, 2024, 10:31:54 AM	7 min 24 sec	Apr 24, 2024, 10:24:30 AM	Succeeded	2.55.1	2024-04-23_21_54_29-1481249222811756739	us-central1	1 INSIGHT

The screenshot shows the Google Cloud Dataflow job monitoring interface. On the left, a sidebar lists options like Overview, Monitoring, Jobs (which is selected), Pipelines, Workbench, Snapshots, and SQL Workspace. The main area displays a 'JOB GRAPH' showing three stages: 'Read' (Succeeded, 7 sec, 2 of 2 stages succeeded), 'Split' (Succeeded, 0 sec, 1 of 1 stage succeeded), and 'PairWithOne'. To the right is a 'Job info' panel with detailed information about the job, including its name, ID, type, status, region, location, workers, latest worker status, and start time.

Job name	beamapp-saikoushik2901-0424045425-900927-1kv19ltt
Job ID	2024-04-23_21_54_29-1481249222811756739
Job type	Batch
Job status	Succeeded
SDK version	Apache Beam Python 3.10 SDK 2.55.1
Job region	us-central1
Worker location	us-central1
Current workers	0
Latest worker status	Worker pool stopped.
Start time	April 24, 2024 at 10:24:30 AM

Here we can monitor the job in dataflow

The screenshot shows the Google Cloud Dataflow monitoring interface. The sidebar includes Overview, Monitoring (selected), Jobs, Pipelines, Workbench, Snapshots, and SQL Workspace. The main area features four cards: 'Running jobs' (2), 'Workers per job' (2), 'Quota exceeded errors' (No data available for the selected time frame), and 'CPUs per job' (5). Each card includes a timeline from UTC+5:30 to 12:00 PM on April 24, 2024.

The screenshot shows the Google Cloud Dataflow interface. On the left, a sidebar lists options: Overview, Monitoring, Jobs (selected), Pipelines, Workbench, Snapshots, and SQL Workspace. The main area displays a job named "beamapp-saikoushik2901-0424045425-900927-1kv19ltt". The "JOB LOGS" tab is active, showing log entries from April 24, 2024, at 10:24:40 IST. The logs detail the execution of operations like Read, Write, and EmitSource. The "Job info" panel on the right provides detailed information about the job, including its name, ID, type (Batch), status (Succeeded), SDK version (Apache Beam Python 3.10 SDK 2.55.1), region (us-central1), location (us-central1), and start time (April 24, 2024 at 10:24:30 AM). The logs table has columns for Severity, Timestamp, and Summary.

Severity	Timestamp	Summary
Info	2024-04-24 10:24:40.673 IST	Worker configuration: n1-standard-1 in us-central1-f.
Info	2024-04-24 10:24:41.985 IST	Executing operation Read/Read/Impulse+Read/Read/EmitSource+...
Info	2024-04-24 10:24:42.003 IST	Executing operation Write/Write/WriteImpl/DoOnce/Impulse+Wr...
Info	2024-04-24 10:24:42.048 IST	Starting 1 workers in us-central1-f...
Info	2024-04-24 10:28:15.690 IST	All workers have finished the startup processes and began t...
Info	2024-04-24 10:28:35.184 IST	Finished operation Write/Write/WriteImpl/DoOnce/Impulse+Wri...

Conclusion:- by using dataflow we successfully executed wordcount in google cloud platform.