



PREV

[Chapter 7 Designing Databases for Reliability, Scalability](#)

NEXT

[Chapter 9 Deploying Machine Learning Pipelines](#)

Chapter 8 Understanding Data Operations for Flexibility and Portability

Google Cloud Professional Data Engineer Exam objectives covered in this chapter include the following:

- ✓ **4-4 Ensuring flexibility and portability. Considerations include:**
 - Mapping to current and future business requirements
 - Designing for data and application portability (e.g., multi-cloud, data residency requirements)
 - Data staging, cataloging, and discovery



Data engineers are responsible for many aspects of the data lifecycle in addition to determining and designing storage systems. Data may be operated on in several ways:


- Cataloged
- Preprocessed
- Visualized
- Explored
- Processed with workflows

In this chapter, we will discuss how to use the Data Catalog, a metadata management service supporting the discovery and management of datasets in Google Cloud. Then we will turn our attention to Cloud Dataprep, a preprocessing tool for transforming and enriching data. Next, we will look at Data Studio for visualizing data and Cloud Datalab for interactive exploration and scripting. In each case, we will also discuss business requirements of typical use cases.

Cataloging and Discovery with Data Catalog

Enterprises accumulate vast amounts of data, and one of the challenges that comes with that is keeping track of information about datasets. For example, there may be hundreds of Cloud Storage buckets and folders that contain thousands of files. The people responsible for managing data need to keep track of information such as the contents of the data files, the version of the schema if the data is structured, how the data in one file relates to data in other files, who has access to the data, and so on. This kind of metadata about the datasets is crucial for understanding what data is available, what it means, and how it can be used.

Data Catalog is a GCP metadata service for data management. It is fully managed, so there are no servers to provision or configure. Its primary function is to provide a single, consolidated view of enterprise data. Metadata is collected automatically during ingest operations to BigQuery and Cloud Pub/Sub as well through APIs and third-party tools. BigQuery metadata is collected on datasets, tables, and views. Cloud Pub/Sub topic metadata is also automatically collected.



Data Catalog is currently in beta. Until it is

available for general release, it is unlikely that there will be questions about it on the Professional Data Engineer exam.

Nonetheless, data engineers should understand Data Catalog in general because metadata management is essential for compliance, lifecycle data management, and other data engineering tasks.

Before you can use Data Catalog to capture metadata, you need to enable the Data Catalog API in a project that contains the resources created or accessed via the API.

SEARCHING IN DATA CATALOG

The search capabilities in Data Catalog are based on the same search technology that Google uses with Gmail and Google Drive, so it should be familiar to most users of Google services. With the Data Catalog search capabilities, users can filter and find native metadata, which is captured from the underlying storage system that houses the subject data and user-generated metadata that is collected from tags. Tagging is discussed in the next section.

To be able to search metadata with Data Catalog, a user will need permissions to read metadata for the subject assets, such as a BigQuery dataset of a Pub/Sub topic. It is important to remember that Data Catalog is collecting and searching metadata, not the data in the dataset, table, topic, and so forth. Figure 8.1 shows an example overview page of Data Catalog.

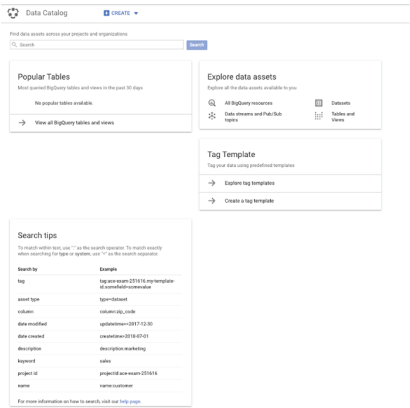


Figure 8.1 Users can search and browse data assets from the Data Catalog overview page.

When metadata is collected from underlying storage systems, Data Catalog is a read-only service. Any changes to the metadata must be made through the underlying storage system. Data Catalog will collect metadata automatically from several resources within a project, including the following:

- Cloud Storage
- Cloud Bigtable
- Google Sheets
- BigQuery
- Cloud Pub/Sub

Metadata can also be collected manually.

TAGGING IN DATA CATALOG

Tags are commonly used in GCP and other public clouds to store metadata about a resource. Tags are used for a wide range of metadata, such as assigning a department or team to all resources that they create in a project or specifying a data classification level to a Cloud Storage bucket or object.

Data Catalog uses templates to help manage user-defined metadata. Figure 8.2 shows an example tag template, which includes template details as well as tag attributes.

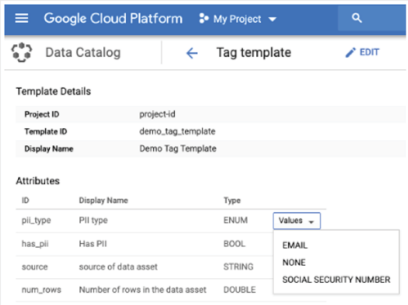



Figure 8.2 Example Data Catalog tag template

Source: <https://cloud.google.com/data-catalog/docs/quickstarts/quickstart-search-tag>

Data Preprocessing with Dataprep

Analysts and data engineers often spend significant amounts of time preparing data for analysis. Cloud Dataprep is a managed service designed to help reduce the time required to prepare data for analysis by providing tools to explore, cleanse, and transform data. There are no servers to provision or configure.

Typical business requirements that drive the use of Cloud Dataprep include the need to analyze datasets that may not be in a format suitable for analysis. This could be due to something as simple as the need for data in different formats as well as more complicated scenarios, such as the need to detect inconsistent or poor-quality data.



Cloud Dataprep is a partner service developed by Trifacta. Before you can use it, a project administrator will have to agree to its license and data access terms.

Cloud Dataprep is an interactive tool that relies on dragging and dropping components within a workflow rather than programming scripts. Users can read data directly from Cloud Storage and BigQuery as well as upload data from their local machines. When data is uploaded, Cloud Dataprep attempts to automatically detect the schema, the data types of values, the distribution of numeric data, and missing or anomalous values. Figure 8.3 shows the kind of summary data that users will see when working with Cloud Dataprep.



Figure 8.3 Cloud Dataprep shows statistics about the distribution of data in attributes.

Source: https://cloud.google.com/dataprep/docs/html/Job-Details-Page_57344846?hl=ES-MX

CLEANSING DATA

Cleansing data can be tedious job. It often requires careful attention to detail about data virtually anywhere in a dataset. In some cases, only a small number of values in a column are missing or incorrectly formatted, and sometimes every value in a column needs to be corrected. Also, there are many ways that data can be incorrect, and each way requires a different procedure to correct.

The main cleansing operations in Cloud Dataprep center around altering column names, reformatting strings, and working with numeric values. Here are some example cleansing tasks that can be performed with Cloud Dataprep:

- Renaming columns
- Changing the datatype of a column
- Copying from one column to another
- Removing and deduplicating data
- Modifying strings
- Extracting values from strings
- Formatting dates
- Applying conditional transformations

The cleansing phase of working with a dataset is often iterative. You may find some data formats that you want to change and then begin to explore the data only to realize that additional anomalies are in the dataset that

you need to correct. The interactive nature of Cloud Dataprep supports this kind of ad hoc, iterative sequence of steps.

DISCOVERING DATA

Another step in processing data for analysis and machine learning is identifying patterns and inconsistencies in your datasets. Cloud Dataprep supports this process by providing for the following:

- Filtering data
- Locating outliers
- Deriving aggregates, such as counts
- Calculating values across columns
- Comparing strings

In addition to performing data cleansing and discovery operations interactively, users can capture sequences of operations in a structure known as a *recipe*.

ENRICHING DATA

Sometimes, datasets need to be augmented with additional columns. For example, datasets may need to be joined or appended together before using them for analysis or machine learning model building. Cloud Dataprep supports several types of enrichment operations, including the following:

- Adding two columns
- Generating primary keys
- Adding lookup data
- Appending datasets
- Joining datasets
- Adding metadata

In the case of adding metadata, Cloud Dataprep can work with data outside the data in datasets. For example, you can reference source file path and filename, creation data, date of importing, and other metadata attributes.

IMPORTING AND EXPORTING DATA

Cloud Dataprep can import a number of flat file formats, including

- Microsoft Excel format (XLS/XLSX)
- CSV
- JSON, including nested
- Plain text
- Tab-separated values (TSV)
- Parquet

The service can also read CSV and JSON files compressed with GZIP, BZIP, and Snappy. Avro files compressed with Snappy can also be imported.

Cloud Dataprep does not change source data, but it is possible to export data after preparing it. Data can be exported to the following:

- CSV
- JSON
- Avro
- BigQuery tables

Users can write compressed CSV and JSON files using GZIP or BZIP.

When data is imported, Cloud Dataprep creates a reference to the dataset, except when data is uploaded from a local device, in which case a copy of the dataset is created.

STRUCTURING AND VALIDATING DATA

Cloud Dataprep has functionality for more advanced transformations, including the following:

- Reshaping data
- Splitting columns
- Creating aggregations
- Pivoting data
- Manipulating arrays
- Manipulating JSON

There are also tools for validating data, including profiling source data. Profiles include information about

- Mismatched values in columns
- Missing values in columns
- Statistical breakout by quartile

Once data has been prepared using Cloud Dataprep, you can then move on to visualize the data with Data Studio.

Visualizing with Data Studio

Data Studio is a reporting and visualization tool. The tool is organized around reports, and it reads data from data sources and formats the data into tables and charts. Figure 8.4 shows an example report generated by Data Studio.

Many business use cases will require the use of Data Studio, including data warehouse reporting and monitoring with dashboards. The three basic tasks in Data Studio are connecting to data sources, visualizing data, and sharing reports.

CONNECTING TO DATA SOURCES

Data Studio uses the concept of a connector for working with datasets. Datasets can come in a variety of forms, including a relational database table, a Google Sheet, or a BigQuery table. Connectors provide access to all or a subset of columns in a data source. Connectors typically require you to authorize access to data.

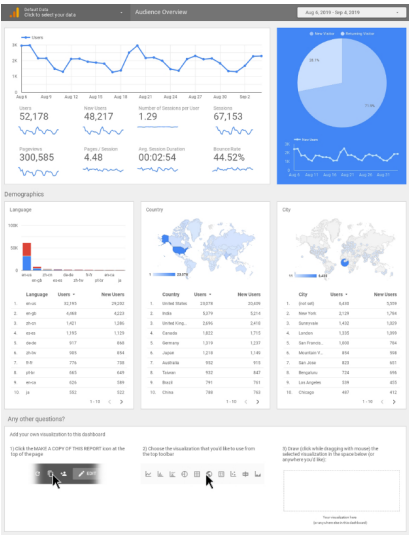


Figure 8.4 An example report showing Google Analytics data

Source: <https://datastudio.google.com/gallery?category=marketing>

There are three kinds of connectors:

Google Connectors These are provided by Google for accessing data from other Google services, including Analytics, Ads, Sheets, and BigQuery.

Partner Connectors These connectors are developed by third parties, and they provide access to non-Google data such as Facebook, GitHub, and Reddit data sources.

Community Connectors These connectors are developed by anyone with a need to access a data source.

There are also three types of data sources:

Live Connection Data Sources These data sources are automatically updated with changes to the underlying data source. Data is stored in the source system. Most connectors work with live data sources.

Extracted Data Sources These data sources work with a static snapshot of a dataset, which can be updated on demand. These may give better performance than live connection data sources.

Blended Data Sources These data sources are the result of combining data from up to five data sources.

Once you have connected to a data source in Data Studio, you can start visualizing data.

VISUALIZING DATA

Data Studio provides components that can be deployed in a drag-and-drop manner to create reports. Data Studio reports are collections of tables and visualizations. The visualization components include the following:

- Line charts
- Bar charts
- Pie charts
- Geo maps
- Area and bubble graphs
- Paginated data tables
- Pivot tables

Users can use filters and data ranges to restrict the set of data included in report tables and charts.

SHARING DATA

Developers of reports can share the report with others, who can then view or edit the report. Reports can also be made available to non-Google users with link sharing.

Data Studio provides the option to schedule the running of a report and the generation of a PDF file, which can be emailed to recipients.

Exploring Data with Cloud Datalab

Cloud Datalab is an interactive tool for exploring and transforming data. Cloud Datalab runs as an instance of a container. Users of Cloud Datalab create a Compute Engine instance, run the container, and then connect from a browser to a Cloud Datalab notebook, as shown in Figure 8.5

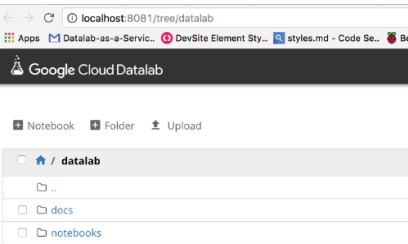
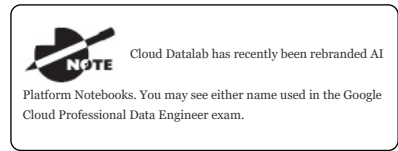


Figure 8.5 An example Cloud Datalab notebook

Source: <https://cloud.google.com/datalab/docs/quickstart>

Cloud Datalab containers run an instance of a Jupyter Notebook.



JUPYTER NOTEBOOKS

Jupyter Notebooks are documents that can contain code as well as text. Code and text are located in cells. All text and code within a cell are treated as a single unit. For example, when a cell is executed, all code in the cell is executed.

Jupyter Notebooks are widely used in data science, machine learning, and other tasks that lend themselves to interactive, iterative development. Jupyter Notebooks support a variety of programming languages, including Python and SQL.

MANAGING CLOUD DATALAB INSTANCES

It is a relatively simple task to create and use Cloud Datalab instances. With the Cloud software development kit (SDK) already installed, including the optional Datalab component, a user can create a Cloud Datalab instance with the `datalab create` command. For example:

```
datalab create --machine-type n1-highmem-2 my-datalab-instance-1
```

Once the instance is created, the user can connect using the `datalab connect` command. The default port for connecting is 8081, but that can be changed by specifying the `--port` option in the `datalab create` command. The `datalab list` command provides a list of all running instances.

A Datalab instance can be deleted using the `datalab delete` command. By default, this command does not delete the persistent disk attached to the instance, assuming that the disk's configuration is also set to the default. To delete the persistent instance as well, users need to specify the `--delete-disk` option.

ADDING LIBRARIES TO CLOUD DATALAB INSTANCES

Data scientists and machine learning engineers often need to import libraries when working with Python. Some of the most commonly used libraries are as follows:

- **Numpy:** A high-performance scientific computing package
- **Scipy:** An open source package for science and engineering that uses numpy
- **Pandas:** An open source package for working with tabular data
- **Scikit Learn:** An open source machine learning package
- **TensorFlow:** An open source machine learning package for deep learning

Many of the commonly used packages are available in Cloud Datalab, but when a user needs to add others, this is done by using either the `conda`

`install` command or the `pip install` command. For example, to install the data analysis package `scikit-data`, a user would specify the following command in a Jupyter Notebook cell:

```
!conda install scikit-data
```

This command runs the `conda` installer to download the `scikit-data` package. Some libraries are not available through the `conda` installer; in that case, the `pip` installer can be used. For example, the topological data analysis tool is currently not available through the `conda` installer, so `pip` should be used to install that library:

```
!pip install scikit-tda
```

The `!` character at the beginning of the command indicates to Jupyter Notebook that the command should be run as a shell command, not a Python statement.

In some cases, exploratory data analysis is an end in itself, but in many cases, it is the first step to defining a workload that will be repeated. In those cases, you can use Cloud Composer to orchestrate those workloads.

Orchestrating Workflows with Cloud Composer

Cloud Composer is a fully managed workflow orchestration service based on Apache Airflow. *Workflows* are defined as directed acyclic graphs (DAGs), which are specified in Python. Workflows can make use of many GCP services, including the following:

- BigQuery
- Cloud Dataflow
- Cloud Dataproc
- Cloud Datastore
- Cloud Storage
- Cloud Pub/Sub
- AI Platform

Elements of workflows can run on premises and in other clouds as well as in GCP.

AIRFLOW ENVIRONMENTS

Apache Airflow is a distributed platform, and it requires several GCP services. When it is deployed, the GCP resources deployed are known as a *Cloud Composer environment*. Environments are stand-alone deployments based on Kubernetes Engine. There can be multiple Cloud Composer environments in a single project.

Environments can be created in the GCP console or by using the command line. The SDK command to create an environment is `gcloud beta composer`.



As of this writing, this service is in beta release. By the time you read this, the word "beta" may have been dropped from the command. Also, some parameters may have changed.

When you create an instance, you can specify node configuration and network configuration, as well as environment variables.

CREATING DAGS

Airflow DAGs are defined in Python as a set of operators and operator relationships. An operator specifies a single task in a workflow. The most commonly used operators are as follows:

- **BashOperator:** Executes a command in the Bash shell
- **PythonOperator:** Executes a Python function
- **EmailOperator:** Sends an email message
- **SimpleHTTPOperator:** Sends HTTP requests
- **Database operators:** Includes `PostgresOperator`, `MySQLOperator`, `SQLiteOperator`, and `JdbcOperator`
- **Sensor:** Waits for a certain event, such as a specific time or the creation of a file or other resource

The order of operators is specified using the `>>` symbol. For example, assuming that you have created a `write_files_python PythonOperator` and a `delete_temp_files_bash BashOperator`, you can have `write_files_python` executed first followed by `delete_temp_files_bash` as follows:

```
write_files_python >> delete_temp_files_bash
```

AIRFLOW LOGS

The Airflow environment creates two types of logs: Airflow logs and streaming logs. *Airflow logs* are associated with a single DAG task. These

files are stored in the Cloud Storage logs folder of the Cloud Composer environment. Logs are retained after an environment is shut down. You will need to delete logs manually. *Streaming logs* are a superset of Airflow logs. These logs are stored in Stackdriver and can be viewed using the Logs viewer. You can also use log-based metrics for monitoring and alerting. Airflow generates several logs, including the following:

- **Airflow-database-init-job:** For database initialization
- **Airflow-scheduler:** For logs generated by the scheduler
- **Airflow-webservice:** For logs generated by the web interface
- **Airflow-worker:** For logs generated as DAGs are executed
- **Airflow-monitoring:** For logs generated by Airflow monitoring
- **Airflow:** For otherwise uncategorized logs

To summarize, the key points about Cloud Composer are that it is a workflow orchestration service that runs within Kubernetes Engine and executes tasks specified in a Python script composed of operators that execute tasks. Tasks can be executed on a schedule, manually, or in response to an external event.

Exam Essentials

Know that Data Catalog is a metadata service for data management. Data Catalog is fully managed, so there are no servers to provision or configure. Its primary function is to provide a single, consolidated view of enterprise data. Metadata is collected automatically during ingest operations to BigQuery and Cloud Pub/Sub, as well through APIs and third-party tools.

Understand that Data Catalog will collect metadata automatically from several GCP sources. These sources include Cloud Storage, Cloud Bigtable, Google Sheets, BigQuery, and Cloud Pub/Sub. In addition to native metadata, Data Catalog can collect custom metadata through the use of tags.

Know that Cloud Dataprep is an interactive tool for preparing data for analysis and machine learning. Cloud Dataprep is used to cleanse, enrich, import, export, discover, structure, and validate data. The main cleansing operations in Cloud Dataprep center around altering column names, reformatting strings, and working with numeric values. Cloud Dataprep supports this process by providing for filtering data, locating outliers, deriving aggregates, calculating values across columns, and comparing strings.

Be familiar with Data Studio as a reporting and visualization tool. The Data Studio tool is organized around reports, and it reads data from data sources and formats the data into tables and charts. Data Studio uses the concept of a connector for working with datasets. Datasets can come in a variety of forms, including a relational database table, a Google Sheet, or a BigQuery table. Connectors provide access to all or to a subset of columns in a data source. Data Studio provides components that can be deployed in a drag-and-drop manner to create reports. Reports are collections of tables and visualization.

Understand that Cloud Datalab is an interactive tool for exploring and transforming data. Cloud Datalab runs as an instance of a container. Users of Cloud Datalab create a Compute Engine instance, run the container, and then connect from a browser to a Cloud Datalab notebook, which is a Jupyter Notebook. Many of the commonly used packages are available in Cloud Datalab, but when users need to add others, they can do so by using either the `conda install` command or the `pip install` command.

Know that Cloud Composer is a fully managed workflow orchestration service based on Apache Airflow. Workflows are defined as directed acyclic graphs, which are specified in Python. Elements of workflows can run on premises and in other clouds as well as in GCP. Airflow DAGs are defined in Python as a set of operators and operator relationships. An operator specifies a single task in a workflow. Common operators include `BashOperator` and `PythonOperator`.

Review Questions

You can find the answers in the appendix.

1. Analysts and data scientists at your company ask for your help with data preparation. They currently spend significant amounts of time searching for data and trying to understand the exact definition of the data. What GCP service would you recommend that they use?
 1. Cloud Composer
 2. Data Catalog
 3. Cloud Dataprep
 4. Data Studio
2. Machine learning engineers have been working for several weeks on building a recommendation system for your company's e-commerce platform. The model has passed testing and validation, and it is ready to be deployed. The model will need to be updated every day with the latest data. The engineers want to automate the model building process that in-

cludes running several Bash scripts, querying databases, and running some custom Python code. What GCP service would you recommend that they use?

1. Cloud Composer
2. Data Catalog
3. Cloud Dataprep
4. Data Studio

3. A business intelligence analyst has just acquired several new datasets. They are unfamiliar with the data and are especially interested in understanding the distribution of data in each column as well as the extent of missing or misconfigured data. What GCP service would you recommend they use?

1. Cloud Composer
2. Cloud Catalog
3. Cloud Dataprep
4. Data Studio

4. Line-of-business managers have asked your team for additional reports from data in a data warehouse. They want to have a single report that can act as a dashboard that shows key metrics using tabular data as well as charts. What GCP service would you recommend?

1. Cloud Composer
2. Data Catalog
3. Cloud Dataprep
4. Data Studio

5. You are using Cloud Dataprep to prepare datasets for machine learning. Another team will be using the data that you prepare, and they have asked you to export your data from Cloud Dataprep. The other team is concerned about file size and asks you to compress the files using GZIP. What formats can you use in the export file?

1. CSV only
2. CSV and JSON only
3. CSV and AVRO only
4. JSON and AVRO only

6. The finance department in your company is using Data Studio for data warehouse reporting. Their existing reports have all the information they need, but the time required to update charts and tables is longer than expected. What kind of data source would you try to improve the query performance?

1. Live data source
2. Extracted data source
3. Compound data source
4. Blended data source

7. A DevOps team in your company uses Data Studio to display application performance data. Their top priority is timely data. What kind of connection would you recommend they use to have data updated in reports automatically?

1. Live data source
2. Extracted data source
3. Compound or blended data source
4. Extracted or live data source

8. A machine learning engineer is using Data Studio to build models in Python. The engineer has decided to use a statistics library that is not installed by default. How would you suggest that they install the missing library?

1. Using `conda install` or `pip install` from a Cloud shell
2. Using `conda install` or `pip install` from within a Jupyter Notebook
3. Use the Linux package manager from within a Cloud shell
4. Download the source from GitHub and compile locally

9. A DevOps engineer is working with you to build a workflow to load data from an on-premises database to Cloud Storage and then run several data preprocessing and analysis programs. After those are run, the output is loaded into a BigQuery table, an email is sent to managers indicating that new data is available in BigQuery, and temporary files are deleted. What GCP service would you use to implement this workflow?

1. Cloud Dataprep

- 2. Cloud Dataproc
- 3. Cloud Composer
- 4. Data Studio

10. You have just received a large dataset. You have comprehensive documentation on the dataset and are ready to start analyzing. You will do some visualization and data filtering, but you also want to be able to run custom Python functions. You want to work interactively with the data. What GCP service would you use?

- 1. Cloud Dataproc
- 2. Cloud Datalab
- 3. Cloud Composer
- 4. Data Studio

[Support](#) / [Sign Out](#)

 [PREV](#)
[Chapter 7 Designing Databases for Reliability, Scalability, an...](#)

[Chapter 9 Deploying Machine Learning Pipelines](#) 