# Build a Data Mesh with Dataplex: Challenge Lab

## Task 1. Create a Dataplex lake with two zones and two assets

**Note:** For all tasks in this challenge lab, create the resources in the `us-central1` region, unless otherwise directed.

The Cloud Storage bucket and BigQuery dataset for step 2 have been pre-created in this lab.

1. Create a Dataplex lake named **Sales Lake** with two regional zones:

- Raw zone named **Raw Customer Zone**
- Curated zone named **Curated Customer Zone**

2. Attach one pre-created asset to each zone:

- To the raw zone, attach the Cloud Storage bucket named `qwiklabs-gcp-01-62a5e8382403`**-customer-online-sessions** as a new asset named **Customer Engagements**.
- To the curated zone, attach the BigQuery dataset named `qwiklabs-gcp-01-62a5e8382403`**.customer_orders** as a new asset named **Customer Orders**.

**Helpful hint for creating a Dataplex lake!**

Review the lab titled Dataplex: Qwik Start - Console.

### Enable the Cloud Dataplex API

1. In the Google Cloud Console, enter **Cloud Dataplex API** in the top search bar.

2. Click on the result for **Cloud Dataplex API** under Marketplace.

3. Click **ENABLE**.

## Create a lake

In Dataplex, a lake is the highest organizational domain that represents a specific data area or business unit. For example, you can create a lake for each department or data domain in your organization, so that you can organize and provide data for specific user groups.

In this task, you create a lake to start building a data mesh.

1. In the Google Cloud Console, in the **Navigation menu** (☰), navigate to **Analytics** > **Dataplex**.

If prompted `Welcome to the new Dataplex experience`, click **Close**.

2. Under **Manage lakes**, click **Manage**.

3. Click **+Create lake**.

4. Enter the required information to create a new lake:

| **Property** | **Value** |
|---|---|
| **Display Name** | **Sales Lake** |
| **ID** | Leave the default value. |
| **Region** | `us-central1` |

Leave the other default values.

5. Click **Create**.

It can take up to 3 minutes for the lake to be created.

## Add a zone to your lake

After you create a lake, you can add zones to the lake. Zones are subdomains within a lake that you can use to categorize data further. For example, you can categorize data by stage, usage, or restrictions.

There are two types of zones:

- Raw zones contain data in raw formats (such as files in Cloud Storage buckets) and are not subject to strict type-checking.
- Curated zones contain data that is cleaned, formatted, and ready for analytics such as BigQuery tables.

In this task, you create a raw zone for working with files in a Cloud Storage bucket.

1. On the **Manage** tab, click on the name of your lake.

2. Click **+Add zone**.

3. Enter the required information to create a new zone:

| **Property** | **Value** |
|---|---|
| **Display Name** | `temperature raw data` |
| **ID** | Leave the default value. |
| **Type** | **Raw zone** |
| **Data locations** | **Regional** |

Leave the other default values.

For example, the option for **Enable metadata discovery** under **Discovery settings** is enabled by default and allows authorized users to discover the data in the zone.

4. Click **Create**.

It can take up to 2 minutes for the zone to be created.

You can perform the next task once the status of the zone is **Active**.

## Attach an asset to a zone

Data stored in Cloud Storage buckets or BigQuery datasets can be attached as assets to zones within a Dataplex lake.

In this task, you attach a Cloud Storage bucket that you create in the Google Cloud console.

1. On the **Zones** tab, click on the name of your zone.

2. On the **Assets** tab, click **+ADD ASSEST**.

3. Click **+Add AN ASSET**.

4. Enter the required information to attach a new asset:

| **Property** | **Value** |
|---|---|
| **Type** | **Storage bucket** |
| **Display Name** | `measurements` |
| **ID** | Leave the default value. |

Leave the other default values.

5. For **Bucket**, click **Browse**.

You can attach an existing Cloud Storage bucket or create a new one without leaving Dataplex. In the next steps, you create a new Cloud Storage bucket and attach it to the zone.

6. Click **+Create new bucket** ( ).

7. Provide your project ID as the bucket name (`Project ID`), and then click **Continue**.

8. For **Location type**, select **Region**, and then select `Region`.

Leave other default values.

9. Click **Create**.

If prompted `Public access will be prevented`, click **Confirm**.

10. Click **Select** to select the bucket you just created, and then click **Continue**.

11. For **Discovery settings**, select **Inherit** to inherit the Discovery settings from the zone level, and then click **Continue**.

12. Click **Submit**.

## Task 2. Create and apply a tag template to a zone

1. Create a public tag template named **Protected Customer Data Template** with two enumerated fields:

- First field named **Raw Data Flag** with two values: `Yes` and `No`.
- Second field named **Protected Contact Information Flag** with two values: `Yes` and `No`

2. Use this template to tag the **Raw Customer Zone** using a value of `Yes` for both flags.

In this task, you create a public tag template to label BigQuery table columns with a protected status. With a public tag template, users who have access to the underlying BigQuery table columns will be able to see the tags applied to the columns.

1. On the left menu, under **Manage Metadata**, click **Tag templates**.

2. Click **Create tag template**.

3. Enter the required information to define the tag template:

| **Property** | **Value** |
|---|---|
| **Template Display Name** | **Protected Customer Data Template** |
| **Template ID** | Leave the default value. |
| **Location** | `us-central1` |
| **Visibility** | **Public** |

4. Click **Add field**, and enter the required information to add a new field to the template:

| **Property** | **Value** |
|---|---|
| **Field Display Name** | **Raw Data Flag** |
| **Field ID** | Leave the default value. |
| **Type** | **Enumerated** |

5. For **Enumerated values** > **Values 1**, enter `YES`.

6. Click **Add value**, and for **Values 2**, enter `NO`.

7. Click **Done**.

5. Click **Add field**, and enter the required information to add a new field to the template:

| **Property** | **Value** |
|---|---|
| **Field Display Name** | **Protected Contact Information Flag** |
| **Field ID** | Leave the default value. |
| **Type** | **Enumerated** |

8. For **Enumerated values** > **Values 1**, enter `YES`.

9. Click **Add value**, and for **Values 2**, enter `NO`.

10. Click **Done**.

11. Click **Create**.

## Task 3. Assign a Dataplex IAM role to another user

- Using the principle of least privilege, assign the appropriate Dataplex IAM role to User 2 (`student-00-1c7d4de31e60@qwiklabs.net`) that allows them to upload new Cloud Storage files to the Dataplex asset named **Customer Engagements**.

1. In the Google Cloud Console, in the **Navigation menu** (☰), under **Analytics**, navigate to **Dataplex** > **Secure**.

2. In the **Dataplex resources** menu, expand the arrow next to the project ID (s`tudent-00-1c7d4de31e60@qwiklabs.net`).

3. Expand the arrow next to the name of your lake.

4. Expand the arrow next to the name of your zone.

5. Click on the asset name (Customer Online Sessions).

6. Click **Grant access**.

7. For **New principals**, enter the email for User 2: `User 2 ID`

8. For **Select a role**, select **Dataplex Data Writer** under **Cloud Dataplex**.

9. Click **Save**.

## Task 4. Create and upload a data quality specification file to Cloud Storage

The Cloud Storage bucket for step 2 has been pre-created in this lab.

1. Create a data quality specification file named **dq-customer-orders.yaml** with the following specifications:

- **NOT NULL** rule applied to the **user_id** column of the **customer_orders.ordered_items** table
- **NOT NULL** rule applied to the **order_id** column of the **customer_orders.ordered_items** table

2. Upload the file to the Cloud Storage bucket named `Project ID`**-dq-config**.

### Create the data quality specification file

1. In Cloud Shell, run the following command to create a new empty file for the data quality specification:

```
nano dq-customer-raw-data.yaml
```

Copied!

content_copy

2. Paste the following code:

```
metadata_registry_defaults:
  dataplex:
    projects: Project ID
    locations: Region
    lakes: ecommerce-lake
    zones: customer-contact-raw-zone
row_filters:
  NONE:
    filter_sql_expr: |-
      True
  INTERNATIONAL_ITEMS:
    filter_sql_expr: |-
      REGEXP_CONTAINS(item_id, 'INTNL')
rule_dimensions:
  - consistency
  - correctness
  - duplication
  - completeness
  - conformance
  - integrity
  - timeliness
  - accuracy
rules:
  NOT_NULL:
    rule_type: NOT_NULL
    dimension: completeness
  VALID_EMAIL:
    rule_type: REGEX
    dimension: conformance
    params:
      pattern: |-
        ^[^@]+[@]{1}[^@]+$
rule_bindings:
  VALID_CUSTOMER:
    entity_uri: bigquery://projects/Project
ID/datasets/customers/tables/contact_info
    column_id: id
    row_filter_id: NONE
    rule_ids:
      - NOT_NULL
  VALID_EMAIL_ID:
    entity_uri: bigquery://projects/Project
ID/datasets/customers/tables/contact_info
    column_id: email
    row_filter_id: NONE
    rule_ids:
      - VALID_EMAIL
```

Copied!

content_copy

3. Review the code to identify the two primary data quality rules that are defined in this file.

The `dq-customer-raw-data.yaml` file begins with key parameters to identify the Dataplex resources including the project ID, region, and names of the Dataplex lake and zone.

Next, it specifies the allowed rule dimensions and two primary rules:

- The rule for **NOT_NULL** values refers to the completeness dimension such as null values.
- The rule for **VALID_EMAIL** values refers to the conformance dimension such as invalid values.

Last, the rules are bound to entities (tables) and columns using rule bindings for data quality validation:

- The first rule binding named **VALID_CUSTOMER** binds the **NOT_NULL** rule to the **id** column of the **contact_info** table, which will validate if the ID column has any NULL values.
- The second rule binding named **VALID_EMAIL_ID** binds the **VALID_EMAIL** rule to the **email** column of the **contact_info** table, which will check for valid emails.

4. Enter `Ctrl+X`, then `Y`, to save and close the file.

## Upload the file to Cloud Storage

- In Cloud Shell, run the following command to upload the file to a Cloud Storage bucket that has been created for this lab:

```
gsutil cp dq-customer-raw-data.yaml gs://Project ID-bucket
```

## Task 5. Define and run a data quality job in Dataplex

The BigQuery dataset for step 1 has been pre-created in this lab.

1. Define a data quality job using the **dq-customer-orders.yaml** file with the following specifications:

| Property | Value |
|---|---|

| | |
|---|---|
| **Data Quality Job Name** | **Customer Orders Data Quality Job** |
| **BigQuery destination table for the results** | `student-00-1c7d4de31e60@qwiklabs.net`)..**orders_dq_dataset.results** |
| **User service account** | **Compute Engine default service account** |

2. Run the data quality job immediately.

In this task, you define and run a data quality job using the data quality specification YAML file uploaded to Cloud Storage in the previous task. When you define the job, you also specify a pre-created BigQuery dataset named **customer_dq_dataset** to store the data quality results.

1. In the Google Cloud Console, in the **Navigation menu** (☰), navigate to **Analytics** > **Dataplex**.

2. Under **Manage lakes**, click **Process**.

3. Click **+CREATE TASK**.

4. Under Check Data Quality, click **Create task**.

5. Enter the required information to create a new data quality job:

| **Property** | **Value** |
|---|---|
| **Dataplex lake** | **ecommerce-lake** |
| **Display name** | `Customer Data Quality Job` |
| **ID** | Leave the default value. |

| | |
|---|---|
| **Select GCS file** | `student-00-1c7d4de31e60@qwiklabs.net` -bucket/dq-customer-raw-data.yaml |
| **Select BigQuery dataset** | `student-00-1c7d4de31e60@qwiklabs.net`.customers_dq_dataset |
| **BigQuery table** | `dq_results` |
| **User service account** | **Compute Engine default service account** |

Leave the other default values.

Note that the Compute Engine default service account has been preconfigured for this lab to have the appropriate IAM roles and permissions. For more information, review the Dataplex documentation titled Create a service account.

6. Click **Continue**.

7. For **Start**, select **Immediately**.

8. Click **Create**.

**Note:** It can take several minutes for the job to run. You may need to refresh the page to see that the job has run successfully.