

# Manage transfers

This document shows how to manage existing data transfer configurations.

You can also [manually trigger an existing transfer](#) (`#manually_trigger_a_transfer`), also known as starting a *backfill run*.

## View your transfers

View your existing transfer configurations by viewing information about each transfer, listing all existing transfers, and viewing transfer run history or log messages.

## Required roles

To get the permissions that you need to view transfer details, ask your administrator to grant you the [BigQuery User](https://cloud.google.com/iam/docs/understanding-roles#bigquery.user) (`roles/bigquery.user`) IAM role on the project. For more information about granting roles, see [Manage access](#) (`/iam/docs/granting-changing-revoking-access`).

You might also be able to get the required permissions through [custom roles](#) (`/iam/docs/creating-custom-roles`) or other [predefined roles](#) (`/iam/docs/understanding-roles`).

Additionally, to view log messages through Google Cloud console, you must have permissions to view Cloud Logging data. The Logs Viewer role (`roles/logging.viewer`) gives you read-only access to all features of Logging. For more information about the Identity and Access Management (IAM) permissions and roles that apply to cloud logs data, see the Cloud Logging [access control guide](#) (`/logging/docs/access-control`).

For more information about IAM roles in BigQuery Data Transfer Service, see [Access control](#) (`/bigquery/docs/access-control`).

## Get transfer details

After you create a transfer, you can get information about the transfer's configuration. The configuration includes the values you supplied when you created the transfer, as well as other important information such as resource names.

To get information about a transfer configuration:

[Console](#) (#console) [bq](#) (#bq) [API](#) (#api) [Java](#) (#java) [Python](#) (#python)

1. Go to the **Data transfers** page.

[Go to Data transfers](https://console.cloud.google.com/bigquery/transfers) (https://console.cloud.google.com/bigquery/transfers)

2. Select the transfer for which you want to get the details.

3. To see the transfer configuration and the data source details, click **Configuration** on the **Transfer details** page. The following example shows the configuration properties for a Google Ads transfer:

✓ My Transfer	
Transfer config details	
Resource name	projects/[redacted]/locations/us/transferConfigs/[redacted]
Source	Google Ads (formerly AdWords)
Destination dataset	<a href="#">my_dataset</a>
Schedule (UTC)	None
Start date (UTC)	None
End date (UTC)	None
Notification Cloud Pub/Sub topic	None
Email notifications	None
Data source details	
Customer ID	[redacted]
Exclude removed/disabled items	None

## List transfer configurations

To list all existing transfer configurations in a project:

[Console](#) (#console) [bq](#) (#bq) [API](#) (#api) [Java](#) (#java) [Python](#) (#python)

Before trying this sample, follow the Python setup instructions in the [BigQuery quickstart using client libraries](#) (/bigquery/docs/quickstarts/quickstart-client-libraries). For more information, see the [BigQuery Python API reference documentation](#) (/python/docs/reference/bigquery/latest).

To authenticate to BigQuery, set up Application Default Credentials. For more information, see [Set up authentication for client libraries](https://cloud.google.com/bigquery/docs/authentication#client_libs) (/bigquery/docs/authentication#client-lib).

```
from google.cloud import bigquery_datatransfer

transfer_client = bigquery_datatransfer.DataTransferServiceClient()

project_id = "my-project"
parent = transfer_client.common_project_path(project_id)

configs = transfer_client.list_transfer_configs(parent=parent)
print("Got the following configs:")
for config in configs:
    print(f"\tID: {config.name}, Schedule: {config.schedule}")
```

## View transfer run history

As your scheduled transfers are run, a run history is kept for each transfer configuration that includes successful transfer runs and transfer runs that fail. Transfer runs more than 90 days old are automatically deleted from the run history.

To view the run history for a transfer configuration:

Console (#console) API (#api) Java (#java)

1. In the Google Cloud console, go to the Data transfers page.  
  
[Go to Data transfers](https://console.cloud.google.com/bigquery/transfers) (<https://console.cloud.google.com/bigquery/transfers>)
2. Click on the transfer in the data transfers list.
3. You will be on the **RUN HISTORY** page for the selected transfer.

[View transfer run details and log messages](#)

When a transfer run appears in the run history, you can view the run details including log messages, warnings and errors, the run name, and the start and end time.

To view transfer run details:

[Console](#) [bq. \(#bq\)](#) [API \(#api\)](#) [Java \(#java\)](#)  
(#console)

1. In the Google Cloud console, go to the Data transfers page.  
  
[Go to Data transfers](https://console.cloud.google.com/bigquery/transfers) (https://console.cloud.google.com/bigquery/transfers)
2. Click on the transfer in the data transfers list.
3. You will be on the **RUN HISTORY** page for the selected transfer.
4. Click on an individual run of the transfer, and the **Run details** panel will open for that run of the transfer.
5. In the **Run details**, note any error messages. This information is needed if you contact Cloud Customer Care. The run details also include log messages and warnings.

×

Run details

Resource name	projects/[REDACTED]/locations/us /transferConfigs/[REDACTED] [REDACTED]
Started	January 29, 2020 at 5:13:34 PM UTC-8
Duration	13 min 53 sec

## Modify your transfers

You can modify existing transfers by editing information on the transfer configuration, updating a user's credentials attached to a transfer configuration, and disabling or deleting a transfer.

## Required roles

To get the permissions that you need to modify transfers, ask your administrator to grant you the [BigQuery Admin](https://cloud.google.com/iam/docs/understanding-roles#bigquery.admin) (`roles/bigquery.admin`) IAM role on the project. For more information about granting roles, see [Manage access](/iam/docs/granting-changing-revoking-access) (`/iam/docs/granting-changing-revoking-access`).

You might also be able to get the required permissions through [custom roles](/iam/docs/creating-custom-roles) (`/iam/docs/creating-custom-roles`) or other [predefined roles](/iam/docs/understanding-roles) (`/iam/docs/understanding-roles`).

## Update a transfer

After you create a transfer configuration, you can edit the following fields:

- Destination dataset
- Display name
- Any of the parameters specified for the specific transfer type
- Run notification settings
- Service account

You cannot edit the source of a transfer once a transfer is created.

To update a transfer:

[Console](#) (#console) [bq](#) (#bq) [API](#) (#api) [Java](#) (#java) [Python](#) (#python)

Before trying this sample, follow the Python setup instructions in the [BigQuery quickstart using client libraries](/bigquery/docs/quickstarts/quickstart-client-libraries) (`/bigquery/docs/quickstarts/quickstart-client-libraries`). For more information, see the [BigQuery Python API reference documentation](/python/docs/reference/bigquery/latest) (`/python/docs/reference/bigquery/latest`).

To authenticate to BigQuery, set up Application Default Credentials. For more information, see [Set up authentication for client libraries](/bigquery/docs/authentication#client-libs) (`/bigquery/docs/authentication#client-libs`).

```

from google.cloud import bigquery_datatransfer
from google.protobuf import field_mask_pb2

transfer_client = bigquery_datatransfer.DataTransferServiceClient()

transfer_config_name = "projects/1234/locations/us/transferConfigs/abcd"
new_display_name = "My Transfer Config"

transfer_config = bigquery_datatransfer.TransferConfig(name=transfer_config_name, display_name=new_display_name)

transfer_config.display_name = new_display_name

transfer_config = transfer_client.update_transfer_config(
    {
        "transfer_config": transfer_config,
        "update_mask": field_mask_pb2.FieldMask(paths=["display_name"]),
    }
)

print(f"Updated config: '{transfer_config.name}'")
print(f"New display name: '{transfer_config.display_name}'")

```

## Update credentials

A transfer uses the credentials of the user that created it. If you need to change the user attached to a transfer configuration, you can update the transfer's credentials. This is useful if the user who created the transfer is no longer with your organization.

To update the credentials for a transfer:

[Console](#) (#console) [bq](#) (#bq) [API](#) (#api) [Java](#) (#java) [Python](#) (#python)

Before trying this sample, follow the Python setup instructions in the [BigQuery quickstart using client libraries](#) (/bigquery/docs/quickstarts/quickstart-client-libraries). For more information, see the [BigQuery Python API reference documentation](#) (/python/docs/reference/bigquery/latest).

To authenticate to BigQuery, set up Application Default Credentials. For more information, see [Set up authentication for client libraries](#).

(/bigquery/docs/authentication#client-lib).

```
from google.cloud import bigquery_datatransfer
from google.protobuf import field_mask_pb2

transfer_client = bigquery_datatransfer.DataTransferServiceClient()

service_account_name = "abcdef-test-sa@abcdef-test.iam.gserviceaccount.com"
transfer_config_name = "projects/1234/locations/us/transferConfigs/abcd"

transfer_config = bigquery_datatransfer.TransferConfig(name=transfer_config_name)

transfer_client.update_transfer_config(
    {
        "transfer_config": transfer_config,
        "update_mask": field_mask_pb2.FieldMask(paths=["service_account_name",
        "service_account_name": service_account_name,
    })

print("Updated config: '{}'".format(transfer_config.name))
```

## Disable a transfer

When you disable a transfer, ***disabled*** is added to the transfer name. When the transfer is disabled, no new transfer runs are scheduled, and no new backfills are allowed. Any transfer runs in progress are completed.

Disabling a transfer does **not** remove any data already transferred to BigQuery. Data previously transferred incurs standard BigQuery [storage costs](/bigquery/pricing#storage) until you [delete the dataset](/bigquery/docs/managing-datasets#deleting_a_dataset) or [delete the tables](/bigquery/docs/managing-tables#deleting_tables).

To disable a transfer:

[Console](#) [bq](#) ([#bq](#)) [API](#) ([#api](#)) [Java](#) ([#java](#))  
([#console](#))

1. In the Google Cloud console, go to the BigQuery page.

[Go to BigQuery](https://console.cloud.google.com/bigquery) (https://console.cloud.google.com/bigquery)

2. Click **Transfers**.
3. On the **Transfers** page, click on the transfer in the list that you want to disable.
4. Click on **DISABLE**. To re-enable the transfer, click on **ENABLE**.

## Delete a transfer

When a transfer is deleted, no new transfer runs are scheduled. Any transfer runs in progress are stopped.

Deleting a transfer does **not** remove any data already transferred to BigQuery. Data previously transferred incurs standard BigQuery [storage costs](/bigquery/pricing#storage) (/bigquery/pricing#storage) until you [delete the dataset](/bigquery/docs/managing-datasets#deleting_a_dataset) (/bigquery/docs/managing-datasets#deleting\_a\_dataset) or [delete the tables](/bigquery/docs/managing-tables#deleting_tables) (/bigquery/docs/managing-tables#deleting\_tables).

To delete a transfer:

[Console](#) (#console)[bq](#) (#bq)[API](#) (#api)[Java](#) (#java)[Python](#) (#python)

Before trying this sample, follow the Python setup instructions in the [BigQuery quickstart using client libraries](/bigquery/docs/quickstarts/quickstart-client-libraries) (/bigquery/docs/quickstarts/quickstart-client-libraries). For more information, see the [BigQuery Python API reference documentation](/python/docs/reference/bigquery/latest) (/python/docs/reference/bigquery/latest).

To authenticate to BigQuery, set up Application Default Credentials. For more information, see [Set up authentication for client libraries](/bigquery/docs/authentication#client-lib) (/bigquery/docs/authentication#client-lib).

```
import google.api_core.exceptions
from google.cloud import bigquery_datatransfer

transfer_client = bigquery_datatransfer.DataTransferServiceClient()

transfer_config_name = "projects/1234/locations/us/transferConfigs/abcd"
try:
    transfer_client.delete_transfer_config(name=transfer_config_name)
```



```
except google.api_core.exceptions.NotFound:
    print("Transfer config not found.")
else:
    print(f"Deleted transfer config: {transfer_config_name}")
```

## Manually trigger a transfer

You can manually trigger a transfer, also called a *backfill run*, to load additional data files outside of your automatically scheduled transfers. With data sources that support runtime parameters, you can also manually trigger a transfer by specifying a date or a time range to load past data from.

You can manually initiate data backfills at any time. In addition to source limits, the BigQuery Data Transfer Service supports a maximum of 180 days per backfill request. Simultaneous backfill requests are not supported.

**Caution:** When backfilling large date ranges, break your backfill requests into **180 day** chunks, and wait for the previous backfill request to finish before creating another one.

For information on how much data is available for backfill, see the transfer guide for your data source.

## Required roles

To get the permissions that you need to modify transfers, ask your administrator to grant you the BigQuery Admin (<https://cloud.google.com/iam/docs/understanding-roles#bigquery.admin>) (`roles/bigquery.admin`) IAM role on the project. For more information about granting roles, see Manage access (</iam/docs/granting-changing-revoking-access>).

You might also be able to get the required permissions through custom roles (</iam/docs/creating-custom-roles>) or other predefined roles (</iam/docs/understanding-roles>).

## Manually trigger a transfer or backfill

You can manually trigger a transfer or backfill run with the following methods:

- Select your transfer run using the Google Cloud console, then clicking **Run transfer now** or **Schedule backfill**.
- Use the `bq mk -transfer run` command using the bq command-line tool
- Call the `projects.locations.transferConfigs.startManualRuns` method API method

For detailed instructions about each method, select the corresponding tab:

Console (#console) bq (#bq) API (#api) Java (#java) Python (#python)

Before trying this sample, follow the Python setup instructions in the [BigQuery quickstart using client libraries](#) (/bigquery/docs/quickstarts/quickstart-client-libraries). For more information, see the [BigQuery Python API reference documentation](#) (/python/docs/reference/bigquery/latest).

To authenticate to BigQuery, set up Application Default Credentials. For more information, see [Set up authentication for client libraries](#) (/bigquery/docs/authentication#client-libs).

```
import datetime

from google.cloud.bigquery_datatransfer_v1 import (
    DataTransferServiceClient,
    StartManualTransferRunsRequest,
)

# Create a client object
client = DataTransferServiceClient()

# Replace with your transfer configuration name
transfer_config_name = "projects/1234/locations/us/transferConfigs/abcd"
now = datetime.datetime.now(datetime.timezone.utc)
start_time = now - datetime.timedelta(days=5)
end_time = now - datetime.timedelta(days=2)

# Some data sources, such as scheduled_query only support daily run.
# Truncate start_time and end_time to midnight time (00:00AM UTC).
start_time = datetime.datetime(
    start_time.year, start_time.month, start_time.day, tzinfo=datetime.tim
```

```

)
end_time = datetime.datetime(
    end_time.year, end_time.month, end_time.day, tzinfo=datetime.timezone.l
)

requested_time_range = StartManualTransferRunsRequest.TimeRange(
    start_time=start_time,
    end_time=end_time,
)

# Initialize request argument(s)
request = StartManualTransferRunsRequest(
    parent=transfer_config_name,
    requested_time_range=requested_time_range,
)

# Make the request
response = client.start_manual_transfer_runs(request=request)

# Handle the response
print("Started manual transfer runs:")
for run in response.runs:
    print(f"backfill: {run.run_time} run: {run.name}")

```

**Caution:** Depending on the data source, and the amount of data you request, backfills may take several hours or days to complete.

## Logging and monitoring

The BigQuery Data Transfer Service exports logs and metrics to Cloud Monitoring and Cloud Logging that provide observability into your transfers. You can [use Monitoring](/bigquery/docs/dts-monitor) (/bigquery/docs/dts-monitor) to set up dashboards to monitor transfers, evaluate transfer run performance, and view error messages to troubleshoot transfer failures. You can [use Logging](/bigquery/docs/dts-monitor#logs) (/bigquery/docs/dts-monitor#logs) to view logs related to a transfer run or a transfer configuration.

You can also [view audit logs](/bigquery/docs/audit-logging) (/bigquery/docs/audit-logging) that are available to the BigQuery Data Transfer Service for transfer activity and data access logs.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2024-08-23 UTC.