Overview of Amazon S3 transfers

The BigQuery Data Transfer Service for Amazon S3 allows you to automatically schedule and manage recurring load jobs from Amazon S3 into BigQuery.

Supported file formats

The BigQuery Data Transfer Service currently supports loading data from Amazon S3 in one of the following formats:

- Comma-separated values (CSV)
- JSON (newline-delimited)
- Avro
- Parquet
- ORC

Supported compression types

The BigQuery Data Transfer Service for Amazon S3 supports loading compressed data. The compression types supported by BigQuery Data Transfer Service are the same as the compression types supported by BigQuery load jobs. For more information, see Loading compressed and uncompressed data (/bigquery/docs/loading-data#loading_compressed_and_uncompressed_data).

Amazon S3 prerequisites

To load data from an Amazon S3 data source, you must:

- Provide the Amazon S3 URI for your source data
- Have your access key ID

- Have your secret access key
- Set, at a minimum, the AWS managed policy `AmazonS3ReadOnlyAccess` (https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_manage.html) on your Amazon S3 source data

Amazon S3 URIs

When you supply the Amazon S3 URI, the path must be in the following format `s3://bucket/folder1/folder2/...` Only the top-level bucket name is required. Folder names are optional. If you specify a URI that includes only the bucket name, all files in the bucket are transferred and loaded into BigQuery.

Amazon S3 transfer runtime parameterization

The Amazon S3 URI and the destination table can both be parameterized (/bigquery-transfer/docs/s3-transfer-parameters), allowing you to load data from Amazon S3 buckets organized by date. Note that the bucket portion of the URI cannot be parameterized. The parameters used by Amazon S3 transfers are the same as those used by Cloud Storage transfers.

For details, see using runtime parameters in transfers (/bigquery-transfer/docs/s3-transfer-parameters).

Wildcard support for Amazon S3 URIs

If your source data is separated into multiple files that share a common base name, you can use a wildcard in the URI when you load the data. A wildcard consists of an asterisk (*), and can be used anywhere in the Amazon S3 URI except for the bucket name.

While more than one wildcard can be used in the Amazon S3 URI, some optimization is possible when the Amazon S3 URI specifies only a single wildcard:

- There is a higher limit (#quotas_and_limits) on the maximum number of files per transfer run.

- The wildcard will span directory boundaries. For example, the Amazon S3 URI `s3://my-bucket/*.csv` will match the file `s3://my-bucket/my-folder/my-subfolder/my-file.csv`.

Amazon S3 URI examples

Example 1

To load a single file from Amazon S3 into BigQuery, specify the Amazon S3 URI of the file.

```
my-bucket/my-folder/my-file.csv
```

Example 2

To load all files from an Amazon S3 bucket into BigQuery, specify only the bucket name, with or without a wildcard.

```
my-bucket/
```

or

```
my-bucket/*
```

Note that `s3://my-bucket*` is not a permitted Amazon S3 URI, as a wildcard can't be used in the bucket name.

Example 3

To load all files from Amazon S3 that share a common prefix, specify the common prefix followed by a wildcard.

```
my-bucket/my-folder/*
```

Note that in contrast to loading all files from a top level Amazon S3 bucket, the wildcard must be specified at the end of the Amazon S3 URI for any files to be loaded.

Example 4

To load all files from Amazon S3 with a similar path, specify the common prefix followed by a wildcard.

```
my-bucket/my-folder/*.csv
```

Example 5

Note the wildcards span directories, so any `csv` files in `my-folder`, as well as in subfolders of `my-folder` will be loaded into BigQuery.

If you have these source files under a `logs` folder:

```
my-bucket/logs/logs.csv
my-bucket/logs/system/logs.csv
my-bucket/logs/some-application/system_logs.log
my-bucket/logs/logs_2019_12_12.csv
```

then the following identifies them:

```
my-bucket/logs/*
```

Example 6

If you have these source files, but want to transfer only those that have `logs.csv` as the filename:

```
my-bucket/logs.csv
my-bucket/metadata.csv
```

```
my-bucket/system/logs.csv
my-bucket/system/users.csv
my-bucket/some-application/logs.csv
my-bucket/some-application/output.csv
```

then the following identifies the files with `logs.csv` in the name:

```
my-bucket/*logs.csv
```

Example 7

By using multiple wildcards, more control can be achieved over which files are transferred, at the cost of lower limits (#quotas_and_limits). Using multiple wildcards means that each wildcard will only match up to the end of a path within a subdirectory. For example, for the following source files in Amazon S3:

```
my-bucket/my-folder1/my-file1.csv
my-bucket/my-other-folder2/my-file2.csv
my-bucket/my-folder1/my-subfolder/my-file3.csv
my-bucket/my-other-folder2/my-subfolder/my-file4.csv
```

If the intention is to only transfer `my-file1.csv` and `my-file2.csv`, use the following as the value for the Amazon S3 URI:

```
my-bucket/*/*.csv
```

As neither wildcard spans directories, this URI would limit the transfer to only the csv files that are in `my-folder1` and `my-other-folder2`. Subfolders would not be included in the transfer.

AWS access keys

The access key ID and secret access key are used to access the Amazon S3 data on your behalf. As a best practice, create a unique access key ID and secret access key specifically for

Amazon S3 transfers to give minimal access to the BigQuery Data Transfer Service. For information on managing your access keys, see the AWS general reference documentation (https://docs.aws.amazon.com/general/latest/gr/managing-aws-access-keys.html).

Consistency considerations

When you transfer data from Amazon S3, it is possible that some of your data will not be transferred to BigQuery, particularly if the files were added to the bucket very recently. It should take approximately 10 minutes for a file to become available to the BigQuery Data Transfer Service after it is added to the bucket.

In some cases, however, it may take longer than 10 minutes. To reduce the possibility of missing data, schedule your Amazon S3 transfers to occur at least 10 minutes after your files are added to the bucket. For more information on the Amazon S3 consistency model, see Amazon S3 data consistency model (https://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html#ConsistencyModel) in the Amazon S3 documentation.

Egress cost best practice

Transfers from Amazon S3 could fail if the destination table has not been configured properly. Reasons that could result in an improper configuration include:

- the destination table does not exist
- the table schema is not defined
- the table schema is not compatible with the data being transferred

To avoid Amazon S3 egress costs, you should first test a transfer with a small but representative subset of the files. Small means the test should have a small data size, and a small file count.

Pricing

For information on BigQuery Data Transfer Service pricing, see the Pricing (/bigquery-transfer/pricing) page.

Note that costs can be incurred outside of Google by using this service. Please review the Amazon S3 pricing page (https://aws.amazon.com/s3/pricing/) for details.

Quotas and limits

The BigQuery Data Transfer Service uses load jobs to load Amazon S3 data into BigQuery. All BigQuery Quotas and limits (/bigquery-transfer/quotas#load_jobs) on load jobs apply to recurring Amazon S3 transfers, with the following additional considerations.

Value	Limit
Maximum size per load job transfer run	15 TB
Maximum number of files per transfer run when the Amazon S3 URI includes 0 or 1 wildcards	10,000,000 files
Maximum number of files per transfer run when the Amazon S3 URI includes more than 1 wildcard	10,000 files

What's next

- Learn about setting up an Amazon S3 transfer (/bigquery-transfer/docs/s3-transfer)
- Learn about Using runtime parameters in S3 transfers (/bigquery-transfer/docs/s3-transfer-parameters)
- Learn more about the BigQuery Data Transfer Service (/bigquery-transfer/docs/transfer-service-overview)

Last updated 2020-06-25 UTC.