

Connections can further be configured once they have been validated. One important option to set is the frequency with which Fivetran will interrogate the source system for new data. In Figure 7, we can see how easy Fivetran has made it to set the sync frequency with intervals ranging from 5 minutes to 24 hours.

The screenshot shows the 'Setup' tab of a Fivetran connector configuration. At the top, it displays the connector ID 'towson_fiscally' and the status 'Connected'. Below this, the 'Connection Details' section provides a detailed breakdown of the connection parameters, including the connector ID, connected on date (2023-06-28), external ID (net_illuminate), bucket (e2-demo-gtm-insurance), role ARN (arn:*****), schema (s3), table (insurance_demo_s3), folder path (accidents), and pattern (read all files as csv). It also lists file reading behavior (decompress all files with gzip), compression behavior (gzip), error handling (fails a file sync if improperly formatted data detected), modified file merge (overwrite rows), escape character (none), delimiter (none), null sequence (false), headerless CSV (lists and retrieves every newly modified file each sync), and connection method (directly). The 'Sync Frequency' section at the bottom allows setting the replication interval from 5m to 24h, with a note: 'Set how often Fivetran attempts to replicate data from your Amazon S3 source to your destination.'

Figure 7 — Overview of configuration for a Fivetran connector.

Fivetran will immediately interrogate and ingest data from source systems once a connection is validated. Data is stored as Delta tables and can be viewed from within Databricks through the [Catalog Explorer](#). By default, Fivetran will store all data under the Hive metastore. A new schema is created for each new connection, and each schema will contain at least two tables: one containing the data and another with logs from each attempted ingestion cycle (see Figure 8).

The screenshot shows the 'Tables' section of the Catalog Explorer for the 'hive_metastore.insurance_demo_mongodb_master' schema. It lists two tables: 'claims' and 'fivetran_audit'. The 'claims' table was created on June 28, 2023, at 10:42:23 UTC. The 'fivetran_audit' table was created on June 28, 2023, at 10:42:23 UTC. Both tables are owned by the 'fivetran' user.

Figure 8 — Summary of tables created by Fivetran in the Databricks Warehouse for an example connection.

Having the data stored in Delta tables is a significant advantage. Delta Lake natively supports granular data versioning, meaning we can time travel through each ingestion cycle (see Figure 9). We can use DB SQL to interrogate specific versions of the data to analyze how the source records evolved.

The screenshot shows the 'History' tab of the 'fivetran_audit' table in the 'hive_metastore.insurance_demo_mongodb_master' schema. It displays three log entries:

- Version 2, timestamp 2022-06-28T23:06:50, operation MERGE, operation parameters: {"matchedPredicates": "[{"actionType": "update"}]", "notMatchedPredicates": "[{"actionType": "insert"}]", "predicate": "(main.id = _fivetran_staging_id)"}, job ID 1, WriteSerializable: false, 12 items.
- Version 1, timestamp 2022-06-21T14:34:52, operation COPY INTO, operation parameters: {}, 0 items.
- Version 0, timestamp 2022-06-21T14:34:42, operation CREATE TABLE, operation parameters: {}, 4 items.

Figure 9 — View of the history showing changes made to the Fivetran audit table.