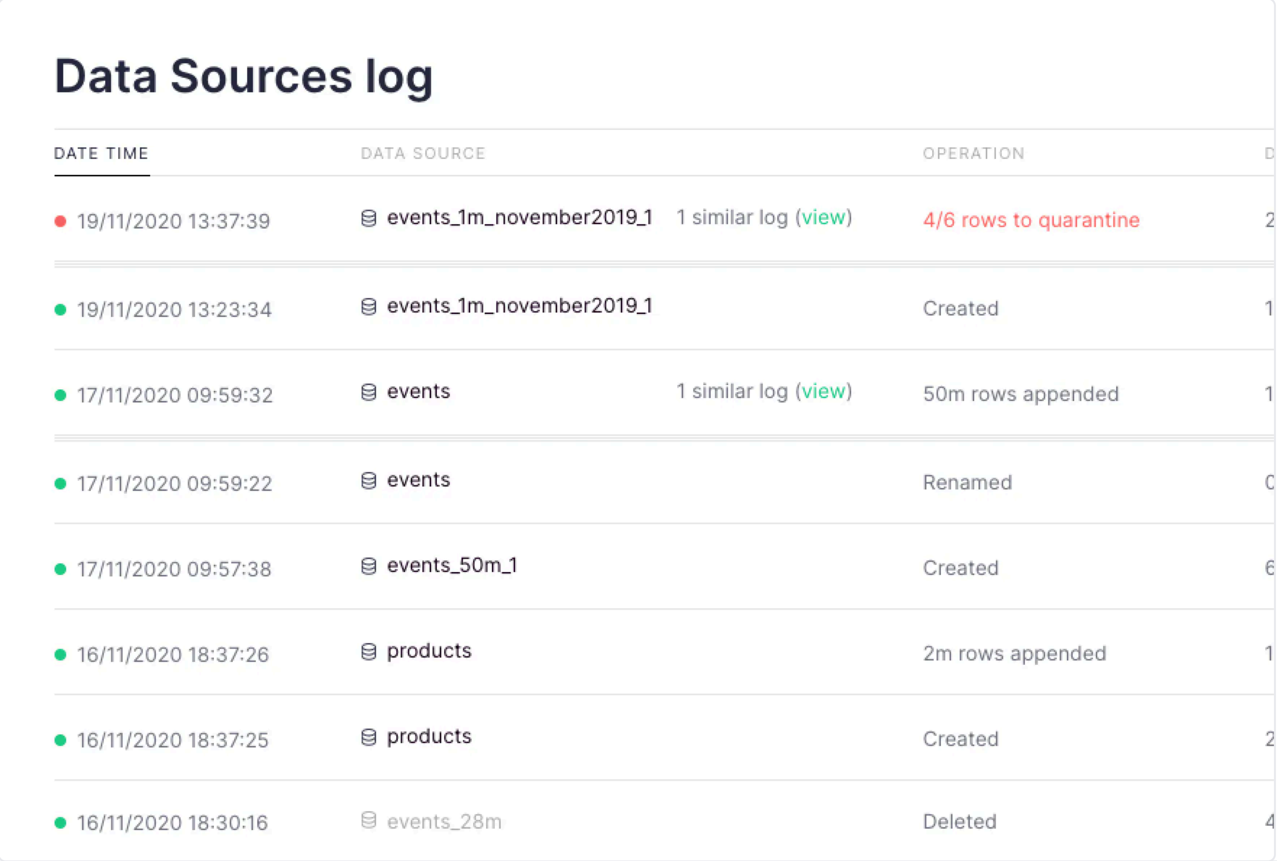
continuously. Use the built-in Tinybird tools to monitor your data ingestion and API endpoint processes.

Data Source health

Once you have fixed all the possible errors in your source files, matched the Data Source schema to your needs and done the on-the-fly transformations (if needed), at some point, you'll start ingesting data periodically. Knowing the status of your ingestion processes will be key.

Data Sources Log

From the 'Data Sources log' in your Dashboard, you can check whether there are new rows in quarantine, if jobs are failing or if there is any other problem.



Data Sources log

DATE TIME	DATA SOURCE		OPERATION	D
● 19/11/2020 13:37:39	events_1m_november2019_1	1 similar log (view)	4/6 rows to quarantine	2
● 19/11/2020 13:23:34	events_1m_november2019_1		Created	1
● 17/11/2020 09:59:32	events	1 similar log (view)	50m rows appended	1
● 17/11/2020 09:59:22	events		Renamed	0
● 17/11/2020 09:57:38	events_50m_1		Created	6
● 16/11/2020 18:37:26	products		2m rows appended	1
● 16/11/2020 18:37:25	products		Created	2
● 16/11/2020 18:30:16	events_28m		Deleted	4

The Data Sources log shows you the operations performed on your data

In addition to the tools we provide in our User Interface (UI), there are powerful tools that you can use for advanced monitoring.

Operations Log

By clicking on an individual Data Source in the left-hand panel you can see the size of the Data Source, the number of rows, the number of rows in the quarantine Data Source (if any) and when it was last updated. The Operations log contains details of the events for the Data Source, which are displayed as the results of the query.

Service Data Sources for continuous monitoring

Service Data Sources can help you with ingestion health checks. They can be used like any other Data Source in your Workspace, which means you can create API endpoints to monitor your ingestion processes.

Querying the 'tinybird.datasources_ops_log' directly, you can, for example, list your ingest processes during the last week:

LISTING INGESTIONS IN THE LAST 7 DAYS

```
SELECT *
FROM tinybird.datasources_ops_log
WHERE toDate(timestamp) > now() - INTERVAL 7 DAY
ORDER BY timestamp DESC
```

This query calculates the percentage of quarantined rows for a given period of time:

CALCULATE % OF ROWS THAT WENT TO QUARANTINE

```
SELECT
  countIf(result != 'ok') / countIf(result == 'ok') * 100 percentage_failed,
  sum(rows_quarantine) / sum(rows) * 100 quarantined_rows
FROM tinybird.datasources_ops_log
```

This query monitors the average duration of your periodic ingestion processes for a given Data Source:

CALCULATING AVERAGE INGEST DURATION

```
SELECT avg(elapsed_time) avg_duration
FROM tinybird.datasources_ops_log
WHERE datasource_id = 't_8417d5126ed84802aa0addce7d1664f2'
```

If you want to configure or build an external service that monitors these metrics, you just need to create an API endpoint and raise an alert when passing a threshold. When you receive an alert, you can check the Quarantine Data Source or the Operations log to see what's going on and fix your source files or ingestion processes.

Monitoring API Endpoints

You can use the Service Data Sources 'pipe_stats' and 'pipe_stats_rt' to monitor the performance of your endpoints.

Every request to a pipe is logged to 'tinybird.pipe_stats_rt' and kept in this Data Source for the last week. This example API endpoint aggregates the statistics for each hour for the selected pipe.

PIPE_STATS_RT_BY_HR

```
SELECT
  toStartOfHour(start_datetime) as hour,
  count() as view_count,
  round(avg(duration), 2) as avg_time,
  arrayElement(quantiles(0.50)(duration),1) as quantile_50,
  arrayElement(quantiles(0.95)(duration),1) as quantile_95,
  arrayElement(quantiles(0.99)(duration),1) as quantile_99
FROM tinybird.pipe_stats_rt
WHERE pipe_id = ‘PIPE_ID’
GROUP BY hour
ORDER BY hour
```

'pipe_stats' contains statistics about your Pipe endpoints API calls aggregated per day using intermediate states.

PIPE_STATS_BY_DATE

```
SELECT
  date,
  sum(view_count) view_count,
  sum(error_count) error_count,
  avgMerge(avg_duration_state) avg_time,
  quantilesTimingMerge(0.9, 0.95, 0.99)(quantile_timing_state) quantiles_timing_
FROM tinybird.pipe_stats
WHERE pipe_id = 'PIPE_ID'
GROUP BY date
ORDER BY date
```

API endpoints such as these can be used to raise alerts for further investigation whenever statistics pass certain thresholds.

To see how Pipes and Data Sources health can be monitored in a dashboard have a look at the blog Operational Analytics in Real Time with Tinybird and Retool.

Company

Product

Pricing

ROI Calculator

About Us

Shop

Careers

Resources

Docs

Blog

Community

Live Coding

Customer Stories

RSS Feed

Request a demo

Integrations

Amazon S3

Kafka Data Streams

Google Cloud Storage

Google BigQuery

Snowflake

Confluent

Use cases

In-Product Analytics

Operational Intelligence

Realtime Personalization

Anomaly Detection & Alerts

Usage Based Pricing

Sports Betting/Gaming

Smart Inventory Management

Serverless ClickHouse

Terms & conditions Cookies Security

Spain
Calle del Dr. Fourquet, 27
28012 Madrid

USA
41 East 11th Street 11th floor
New York, NY 10003