

## Confluent Connector

The Confluent Connector allows you to ingest data from your existing Confluent Cloud cluster into Tinybird.

The Confluent Connector is fully managed and requires no additional tooling. Connect Tinybird to your Confluent Cloud cluster, choose a topic, and Tinybird will automatically begin consuming messages from Confluent Cloud.

Note that you need to grant READ permissions to both the Topic and the Consumer Group to ingest data from Confluent into Tinybird.

## Using the UI

To connect Tinybird to your Confluent Cloud cluster, select the + icon next to the data project section on the left navigation menu, select **Data Source**, and select **Confluent** from the list of available Data Sources.

Enter the following details:

- **Connection name:** A name for the Confluent Cloud connection in Tinybird.
- **Bootstrap Server:** The comma-separated list of bootstrap servers (including port numbers).
- **Key:** The **Key** component of the Confluent Cloud API Key.
- **Secret:** The **Secret** component of the Confluent Cloud API Key.
- **Decode Avro messages with schema registry:** (Optional) Enable Schema Registry support to decode Avro messages. To allow this functionality, enter the Schema Registry URL, username, and password.

Once you have entered the details, select **Connect**. This creates the connection between Tinybird and Confluent Cloud. You then see a list of your existing topics and can select the topic to consume from. Tinybird creates a **Group ID** that specifies the name of the consumer group that this Kafka consumer belongs to. You can customize the Group ID, but ensure that your Group ID has **Read** permissions to the topic.

Once you have chosen a topic, you can select the starting offset to consume from. You can choose to consume from the **earliest** offset or the **latest** offset:

- If you choose to consume from the **earliest** offset, Tinybird will consume all messages from the beginning of the topic.



- If you choose to consume from the **latest** offset, Tinybird will only consume messages that are produced after the connection is created.

Choose the offset, and select **Next**.

Tinybird then consumes a sample of messages from the topic and displays the schema. You can adjust the schema and Data Source settings as needed, then select **Create Data Source**.

Tinybird then begins consuming messages from the topic and loading them into the Data Source. Success!

## Using .datasource files

If you are managing your Tinybird resources in files, there are several settings available to configure the Confluent Connector in `.datasource` files.

- `KAFKA_CONNECTION_NAME`: The name of the configured Confluent Cloud connection in Tinybird.
- `KAFKA_BOOTSTRAP_SERVERS`: The comma-separated list of bootstrap servers (including Port numbers).
- `KAFKA_KEY`: The **Key** component of the Confluent Cloud API Key.
- `KAFKA_SECRET`: The **Secret** component of the Confluent Cloud API Key.
- `KAFKA_TOPIC`: The name of the Kafka topic to consume from.
- `KAFKA_GROUP_ID`: The Kafka Consumer Group ID to use when consuming from Confluent Cloud.
- `KAFKA_AUTO_OFFSET_RESET`: The offset to use when no previous offset can be found, e.g. when creating a new consumer. Supported values: `latest`, `earliest`. Default: `latest`.
- `KAFKA_STORE_RAW_VALUE`: Optionally, you can store the raw message in its entirety as an additional column. Supported values: `'True'`, `'False'`. Default: `'False'`.

For example, to define Data Source with a new Confluent Cloud connection in a `.datasource` file:

DATA SOURCE WITH A NEW CONFLUENT CLOUD CONNECTION

SCHEMA >

```
`value` String,  
`topic` LowCardinality(String),  
`partition` Int16,  
`offset` Int64,  
`timestamp` DateTime,  
`key` String
```

```
ENGINE "MergeTree"
ENGINE_PARTITION_KEY "toYYYYMM(timestamp)"
ENGINE_SORTING_KEY "timestamp"

KAFKA_CONNECTION_NAME my_connection_name
KAFKA_BOOTSTRAP_SERVERS my_server:9092
KAFKA_KEY my_username
KAFKA_SECRET my_password
```

Or, to define Data Source that uses an existing Confluent Cloud connection:

DATA SOURCE WITH AN EXISTING CONFLUENT CLOUD CONNECTION



```
SCHEMA >
  `value` String,
  `topic` LowCardinality(String),
  `partition` Int16,
  `offset` Int64,
  `timestamp` DateTime,
  `key` String

ENGINE "MergeTree"
ENGINE_PARTITION_KEY "toYYYYMM(timestamp)"
ENGINE_SORTING_KEY "timestamp"

KAFKA_CONNECTION_NAME my_connection_name
KAFKA_TOPIC my_topic
KAFKA_GROUP_ID my_group_id
```

## Using INCLUDE to store connection settings

To avoid configuring the same connection settings across many files, or to prevent leaking sensitive information, you can store connection details in an external file and use `INCLUDE` to import them into one or more `.datasource` files.

You can find more information about `INCLUDE` in the [Advanced Templates](#) documentation.

As an example, you may have two Confluent Cloud `.datasource` files, which re-use the same Confluent Cloud connection. You can create an include file which stores the Confluent Cloud connection details.

The Tinybird project may use the following structure:

TINYBIRD DATA PROJECT FILE STRUCTURE



```
ecommerce_data_project/  
  datasources/  
    connections/  
      my_connector_name.incl  
      my_confluent_datasource.datasource  
      another_datasource.datasource  
  endpoints/  
  pipes/
```

Where the file `my_connector_name.incl` has the following content:

```
INCLUDE FILE CONTAINING CONFLUENT CLOUD CONNECTION DETAILS
```

```
KAFKA_CONNECTION_NAME my_connection_name  
KAFKA_BOOTSTRAP_SERVERS my_server:9092  
KAFKA_KEY my_username  
KAFKA_SECRET my_password
```

And the Confluent Cloud `.datasource` files look like the following:

```
DATA SOURCE USING INCLUDES FOR CONFLUENT CLOUD CONNECTION DETAILS
```

```
SCHEMA >  
  `value` String,  
  `topic` LowCardinality(String),  
  `partition` Int16,  
  `offset` Int64,  
  `timestamp` DateTime,  
  `key` String  
  
ENGINE "MergeTree"  
ENGINE_PARTITION_KEY "toYYYYMM(timestamp)"  
ENGINE_SORTING_KEY "timestamp"  
  
INCLUDE "connections/my_connection_name.incl"  
  
KAFKA_TOPIC my_topic  
KAFKA_GROUP_ID my_group_id
```

When using `tb pull` to pull a Confluent Cloud Data Source using the CLI, the `KAFKA_KEY` and `KAFKA_SECRET` settings will **not** be included in the file to avoid exposing credentials.

## FAQs

### Is the Confluent Cloud Schema Registry supported?

Yes, for decoding Avro messages. You can choose to enable Schema Registry support when connecting Tinybird to Confluent Cloud. You will be prompted to add your Schema Registry connection details, e.g. `https://<SCHEMA_REGISTRY_API_KEY>:<SCHEMA_REGISTRY_API_SECRET>@<SCHEMA_REGISTRY_API_URL>`. However, the Confluent Cloud Data Source schema will not be defined using the Schema Registry, the Schema Registry is simply used to decode the messages.

### Can Tinybird ingest compressed messages?

Yes, Tinybird can consume from Kafka topics where Kafka compression has been enabled, as decompressing the message is a standard function of the Kafka Consumer.

However, if you compressed the message before passing it through the Kafka Producer, then Tinybird cannot do post-Consumer processing to decompress the message.

For example, if you compressed a JSON message through gzip and produced it to a Kafka topic as a `bytes` message, it would be ingested by Tinybird as `bytes`. If you produced a JSON message to a Kafka topic with the Kafka Producer setting `compression.type=gzip`, while it would be stored in Kafka as compressed bytes, it would be decoded on ingestion and arrive to Tinybird as JSON.

#### Company

[Product](#)[Pricing](#)[ROI Calculator](#)[About Us](#)

#### Resources

[Docs](#)[Blog](#)[Community](#)[Live Coding](#)

[Shop](#)

[Customer Stories](#)

[Careers](#)

[RSS Feed](#)

[Request a demo](#)

**Integrations**

**Use cases**

[Amazon S3](#)

[In-Product Analytics](#)

[Kafka Data Streams](#)

[Operational Intelligence](#)

[Google Cloud Storage](#)

[Realtime Personalization](#)

[Google BigQuery](#)

[Anomaly Detection & Alerts](#)

[Snowflake](#)

[Usage Based Pricing](#)

[Confluent](#)

[Sports Betting/Gaming](#)

[Smart Inventory Management](#)

[Serverless ClickHouse](#)

Copyright © 2024 Tinybird. All rights reserved

[Terms & conditions](#) [Cookies](#) [Security](#)

---

Spain  
Calle del Dr. Fourquet, 27  
28012 Madrid

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