


RedPanda Connector

The RedPanda Connector allows you to ingest data from your existing RedPanda cluster and load it into Tinybird.

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

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

Using the UI

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

Enter the following details:

- **Connection name:** A name for the RedPanda connection in Tinybird.
- **Bootstrap Server:** The comma-separated list of bootstrap servers (including Port numbers).
- **Key:** The **Key** component of the RedPanda API Key.
- **Secret:** The **Secret** component of the RedPanda API Key.
- **Decode Avro messages with schema registry:** Optionally, you can enable Schema Registry support to decode Avro messages. You will be prompted to enter the Schema Registry URL, username and password.

Once you have entered the details, select **Connect**. This creates the connection between Tinybird and RedPanda. You will then see a list of your existing topics and can select the topic to consume from. Tinybird will create a **Group ID** that specifies the name of the consumer group this 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 **latest** offset or the **earliest** 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. Select the offset, and click **Next**.



Tinybird will then consume a sample of messages from the topic and display the schema. You can adjust the schema and Data Source settings as needed, then click **Create Data Source** to create the Data Source.

Tinybird will now begin consuming messages from the topic and loading them into the Data Source.

Using .datasource files

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

- `KAFKA_CONNECTION_NAME`: The name of the configured RedPanda connection in Tinybird.
- `KAFKA_BOOTSTRAP_SERVERS`: A comma-separated list of one or more RedPanda brokers (including Port numbers).
- `KAFKA_KEY`: The key used to authenticate with RedPanda.
- `KAFKA_SECRET`: The secret used to authenticate with RedPanda.
- `KAFKA_TOPIC`: The name of the RedPanda topic to consume from.
- `KAFKA_GROUP_ID`: The Consumer Group ID to use when consuming from RedPanda.
- `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 RedPanda connection in a `.datasource` file:

```
DATA SOURCE WITH A NEW REDPANDA 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 RedPanda connection:

```
DATA SOURCE WITH AN EXISTING REDPANDA 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 RedPanda `.datasource` files, which re-use the same RedPanda connection. You can create an `INCLUDE` file that stores the RedPanda connection details.

The Tinybird project may use the following structure:

```
ecommerce_data_project/
├── datasources/
│   ├── connections/
│   │   └── my_connector_name.incl
│   ├── my_kafka_datasource.datasource
│   └── another_datasource.datasource
```

```
|— endpoints/  
|— pipes/
```

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

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

And the RedPanda `.datasource` files look like the following:

```
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 RedPanda Data Source using the CLI, the `KAFKA_KEY` and `KAFKA_SECRET` settings will **not** be included in the file to avoid exposing credentials.

Company

- Product
- Pricing
- ROI Calculator
- About Us
- Shop
- Careers
- Request a demo

Resources

- Docs
- Blog
- Community
- Live Coding
- Customer Stories
- RSS Feed

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

Spain
Calle del Dr. Fourquet, 27
28012 Madrid

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