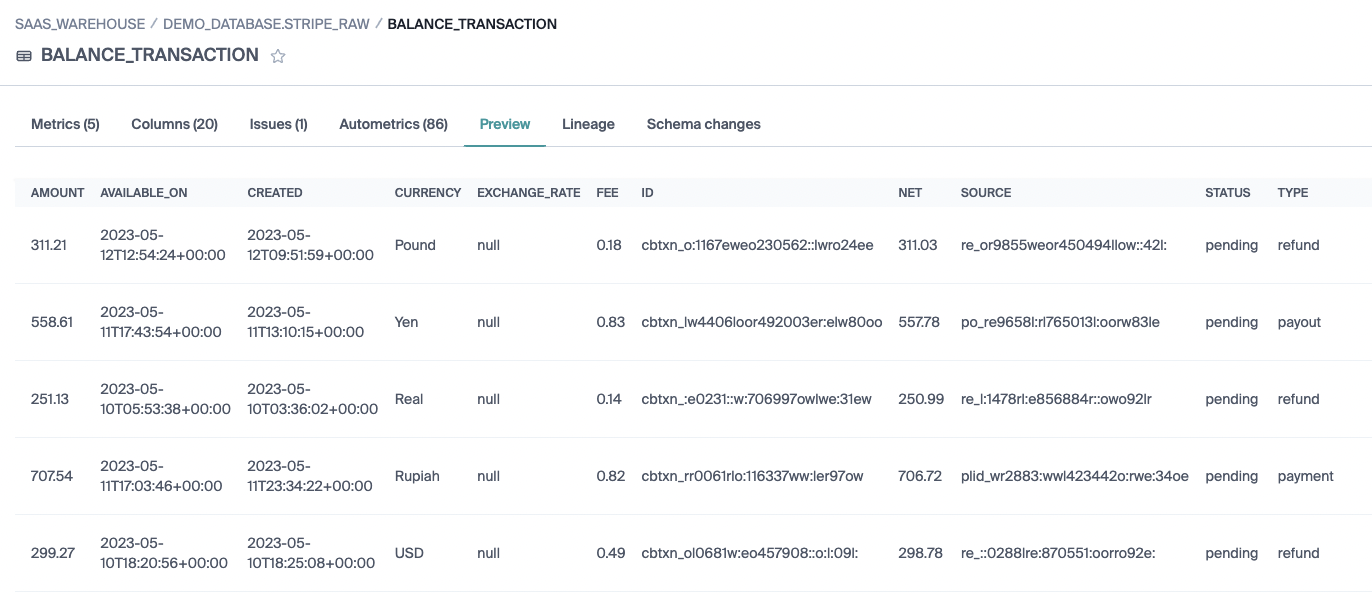
At many companies, Stripe data is relied upon to make key business decisions. One of the best ways to ensure the quality of this data is by using a data observability tool.

This blog post shows you how quickly you can spin up deep data monitoring on your critical Stripe datasets with Bigeye.

### Understanding Stripe Datasets

Depending on which Stripe product you are using, and how you’re using it, different types of raw Stripe tables will be produced. For example, a SaaS software company might use the Stripe subscription product, while an e-commerce company might use Stripe to accept payment for orders. A lending company might use Stripe for credit notes. Below, we show how raw Stripe data might look as it arrives in your data warehouse through an ETL tool like Fivetran.

This is a common setup to ingest your raw Stripe data in your data warehouse or lake, have your data engineering team perform transformations and give access to your BI teams for analysis. Having improper financial data land in your BI teams can result in disastrous business decisions being made.

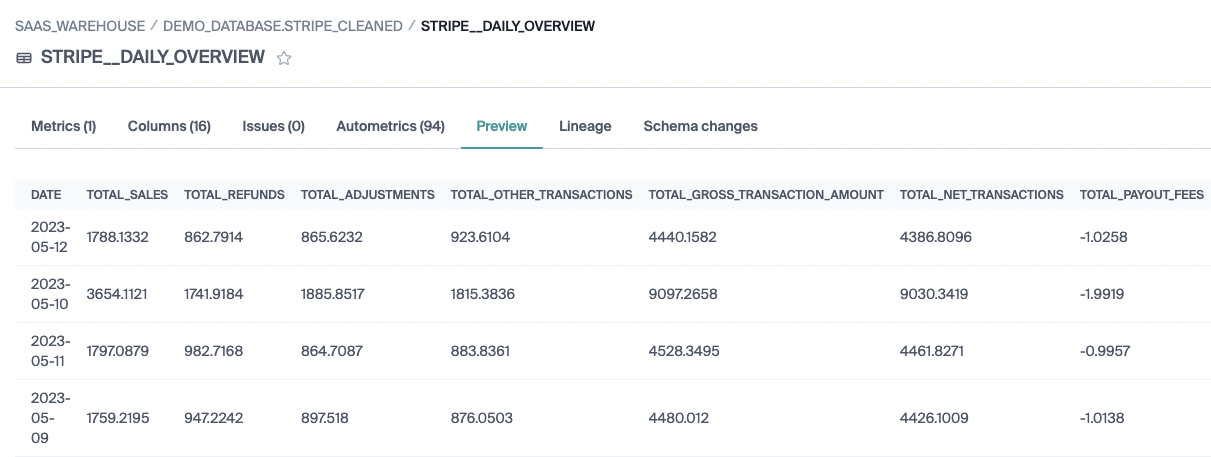


There are two critical tables for every organization that uses Stripe:  
  
**Balance transaction:** a running log of every transaction hitting your Stripe account

**Customer:** a running log of all customer information

Other tables containing data like payment method cards, intent, and payouts, may also be available as metadata on the account itself.

Generally speaking, you will do some preliminary cleanup of the raw Stripe data. Then, the data will be aggregated into higher-level tables that make business metrics like sales, revenue, refunds, etc, easily accessible.





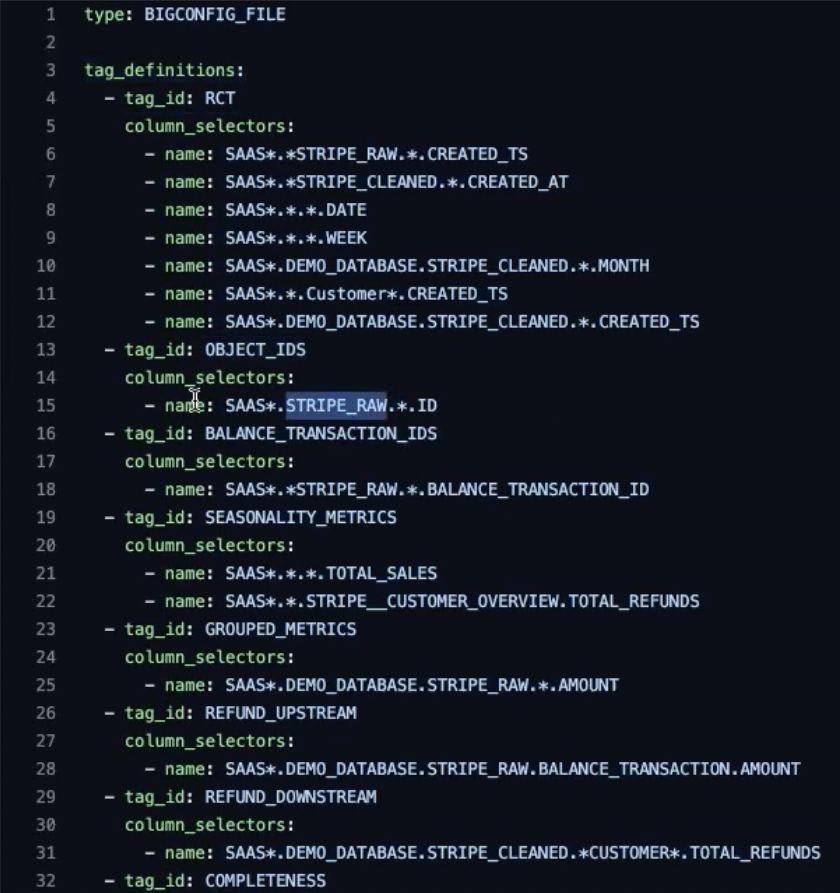
### How to monitor Stripe data with Bigeye

Bigeye suggests taking a programmatic approach to deploying data monitoring on your critical datasets. Why?

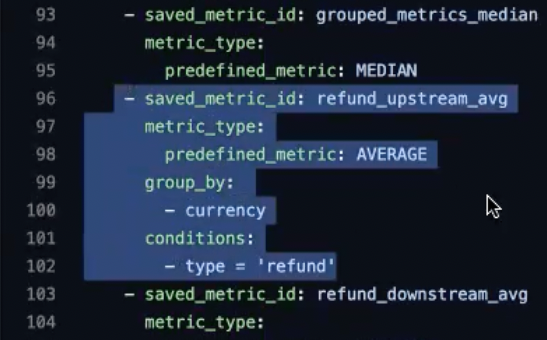
Stripe tables can be repetitive, in that they include many of the same columns. As a result, when deploying metrics against Stripe data, it’s useful to use [Bigconfig](https://docs.bigeye.com/docs/bigconfig), which allows you to specify, in config form, that all columns with certain column names, should be monitored in a certain way.

### Bigconfig

BigConfig consists of two components: **tag definitions** and **metric definitions**. Tag definitions allow you to use selectors with wildcards to identify repetitive columns like transaction IDs, amounts, and currencies that occur across multiple tables. In the example below, the tag definition **OBJECT\_IDS** states that every time a column name is of the form **SAAS\*.STRIPE\_RAW.\*.id**, it belongs in the **OBJECT\_IDS** tag.

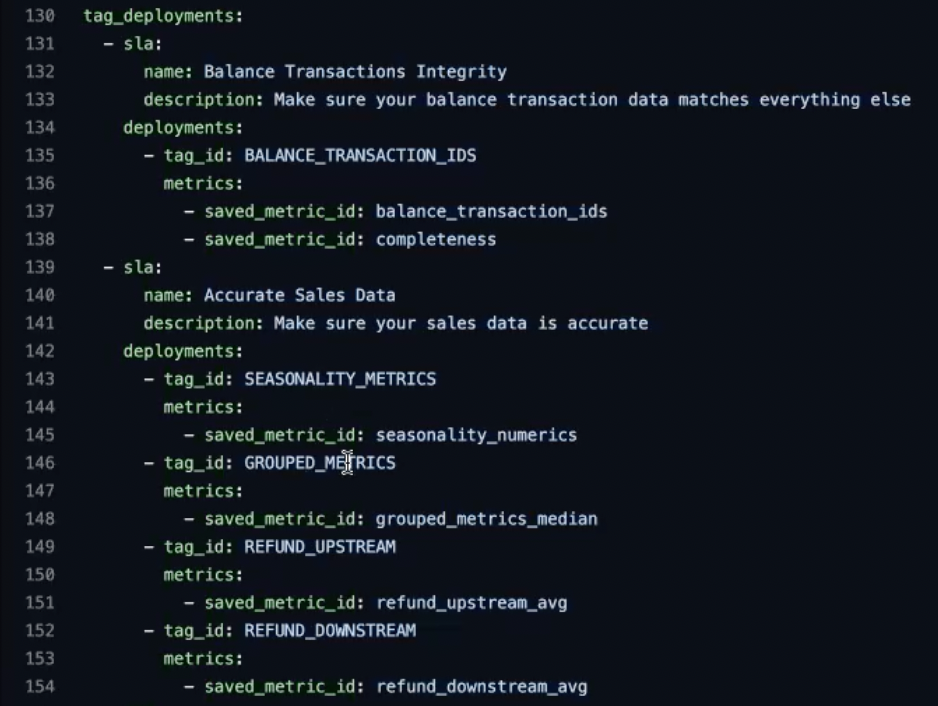


**Metric definitions** then allow you to apply certain kinds of metrics to each tag, without having to go through and enable the appropriate metrics for each column. This can greatly simplify your monitoring deployment.

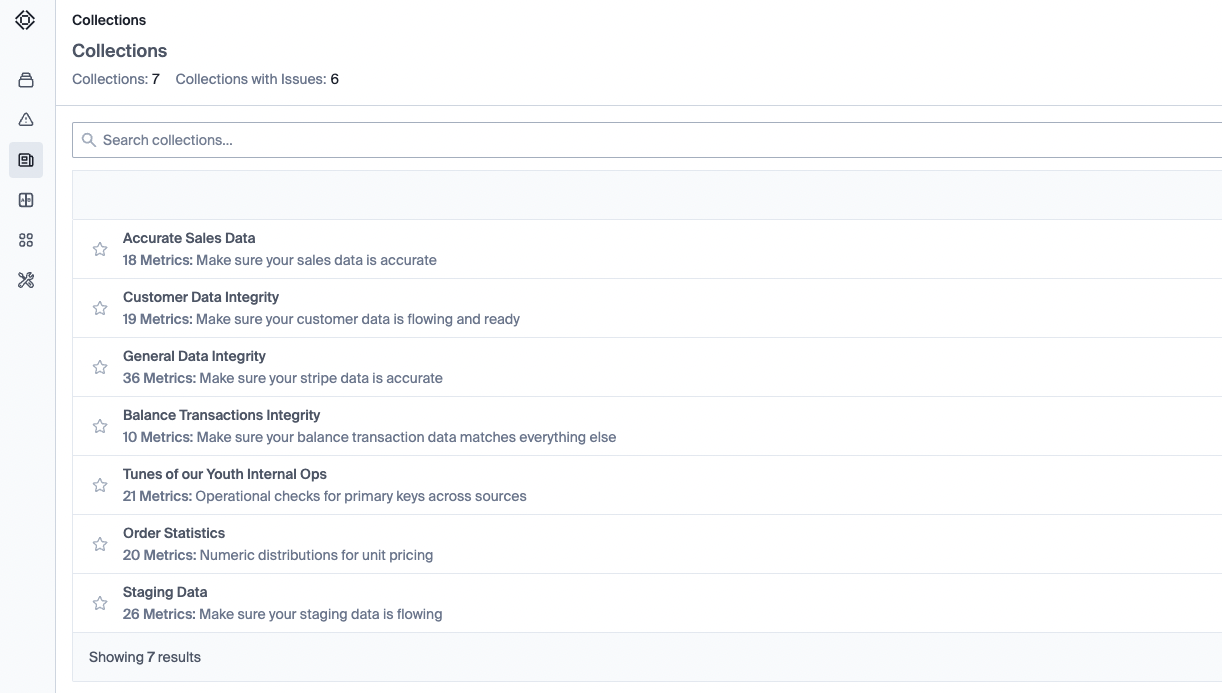


**Metric definitions** also allow you to create custom metrics. For instance, you might want to track your refund data, grouping it by currency and calculating the average refund per currency.

Once you have defined your tags and your metrics, you can apply certain metrics to certain tags in a **deployment**.



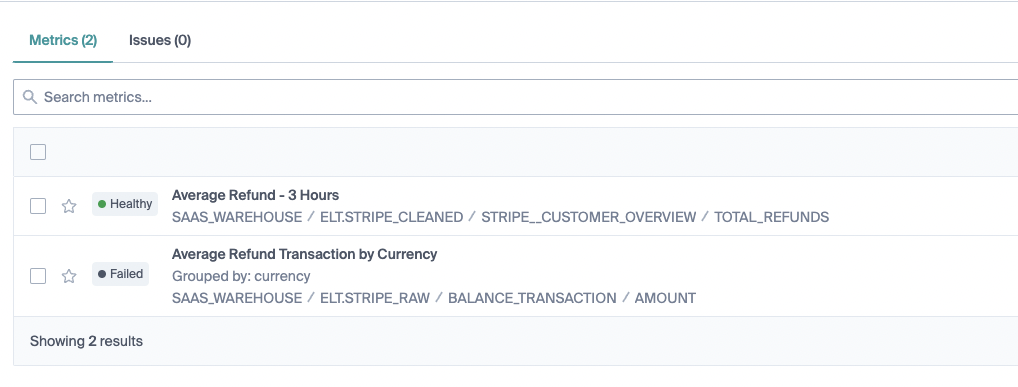
In Bigconfig, you can also create **Collections** to organize the tag and metric definitions. For example, we may create one collection that displays all sales data, one for customer data integrity, and another for general data integrity. Each SLA in the BigConfig file maps to a **collection** in the Bigeye UI. Each collection can be configured to send notifications to specific individuals.



### What types of metrics to monitor on your Stripe data

You’ll notice that in the UI above, we’ve created several collections. These collections roughly correspond to three types metrics that we suggest you monitor on your Stripe data:  
  
**General Data Integrity/Balance Transaction Integrity metrics:** These collections include checks that all transaction IDs are unique, non-null, and matching across multiple tables.

**Accurate Sales Data metrics:** These collections include checks that numerical values like number of sales made, number of refunds issued, chargebacks, invoices, fees, etc., all make sense. We recommend grouping these metrics by currency to ensure each currency is operating correctly.



**Customizable metrics:** Finally, Bigeye’s metric templates allow you to define custom checks. For example, maybe one of your Stripe datasets contains some JSON data. Checking that JSON data directly for whether it’s empty, etc., might not give granular enough guarantees that key/value pairs are present. Instead, with Bigeye, you can define queries that expand the JSON data into specific columns to check into specific values.

### Step-by-step instructions for monitoring your Stripe data

1. [Add the data warehouse](https://docs.bigeye.com/docs/data-source-1) that contains your Stripe data, as a data source in Bigeye
2. Copy [the Stripe BigConfig template](https://docs.bigeye.com/recipes/stripe-prebuilt-bigconfig) into a local yaml file
3. Follow [the instructions in the recipe](https://docs.bigeye.com/recipes/stripe-prebuilt-data-obvservability-as-code-config) to add your own custom tag definitions and metric definitions.
4. Follow the instructions [here](https://docs.bigeye.com/docs/bigconfig#prerequisites) to install the BigConfig CLI tool
5. Apply the BigConfig to your dataset using a single CLI command

### Conclusion

Monitoring your Stripe data with Bigeye can ensure the integrity of your financial data. The process is made easy by the Bigconfig tool, which allows for efficient data monitoring and notification in case of any issues. As we continue to expand our turn-key coverage of other popular SaaS data sources, like Hubspot, we look forward to offering even more low-friction service to our users.

If you would like to try this out, [contact us now](https://www.bigeye.com/request-demo). You just need a read-only account with access to your Stripe data, or you can use our dummy Stripe datasets to give it a spin!