### Introduction

Metadata metrics are a great first step in your data observability journey, but they are not the end of it. Column-level metrics provide deeper information about your data and can be used to make better assessments.

### What are Metadata Metrics?

Metadata metrics are broad freshness and volume metrics meant to tell you if a data pipeline has succeeded or not. As you can see from the table below, they mostly have to do with whether a table has been updated and/or read from.

| Metadata Metric Name | API Name | Description |
| --- | --- | --- |
| Hours since last load | HOURS\_SINCE\_LAST\_LOAD | The number of hours since an INSERT, COPY, or MERGE was performed on a table. It is suggested as an autometric once per table. |
| Rows inserted | ROWS\_INSERTED | The number of rows added to the table via INSERT, COPY, or MERGE statements in the past 24 hours. It is suggested as an autometric once per table. |
| Read queries | COUNT\_READ\_QUERIES | The number of SELECT queries issued on a table in the past 24 hours. It is suggested as an autometric once per table. |

With Bigeye, metadata metrics are available for deployment from the moment you connect your data warehouse: Bigeye scans your existing query logs to automatically track these three metrics across every table. This makes metadata metrics a cornerstone of Bigeye’s T-shaped monitoring philosophy, which recommends that you track fundamentals across all your data while applying deeper monitoring on the most critical datasets, such as those used for financial planning, machine learning models, and executive-level dashboards.

### What do Metadata Metrics not provide?

Metadata metrics do not provide column-level information on things that go wrong, for example:

* If you loaded blank values into a column that never had blank values
* If you loaded dates into a column where there have never been dates before
* If a transform went wrong, and you ended up with the wrong values in different columns than you expected

### I’ve implemented metadata metrics, now what?

Once you’ve set up broad coverage of all your tables with metadata metrics, the next step is to drill down further with column-level metrics.

### How do I turn on column-level metrics?

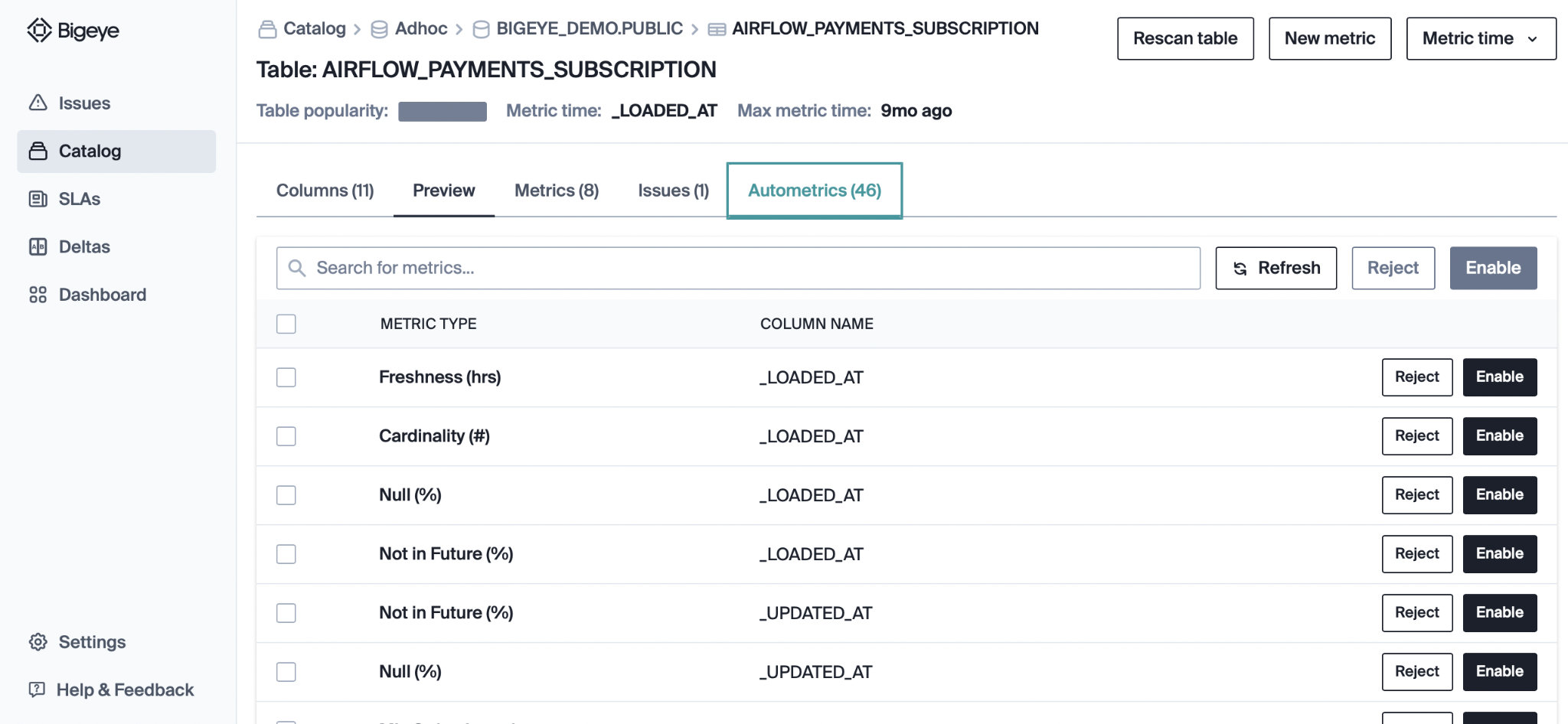
With Bigeye, it’s simple to implement column-level monitoring.

#### Bigeye recommends column-level metrics with autometrics

When you first connect your data warehouse (and whenever a new table is added to the warehouse), Bigeye begins profiling your data to understand what it looks like. Bigeye can then *generate* Autometrics for the table based on the content of each of the columns. For example:

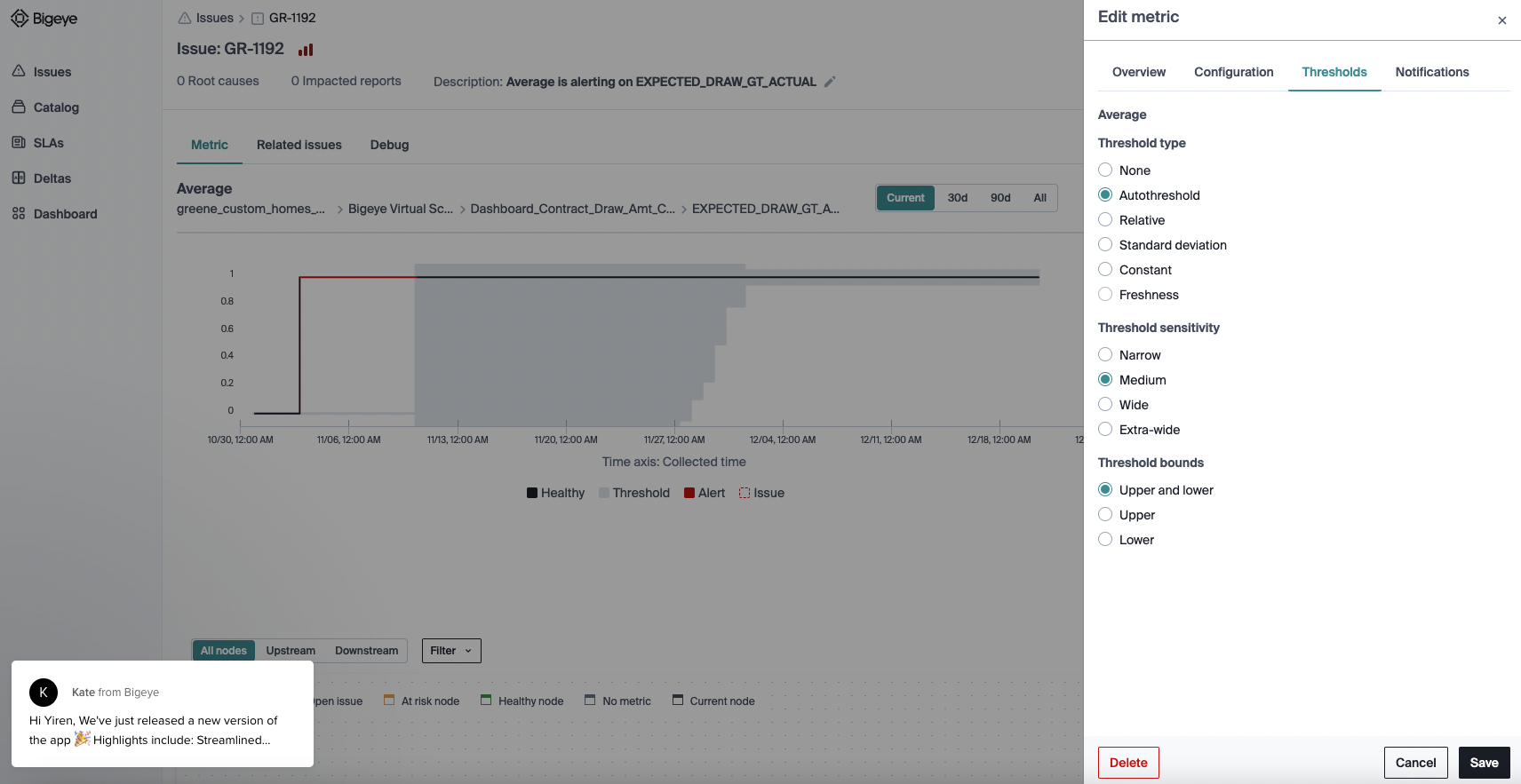
* + If you’ve got three values in the column, it’s probably an enum
  + If you have no duplicate values, maybe you never want the column to have any duplicates
  + Maybe it looks like an ID column, which means you’ll want to check for duplicates
  + If the column is full of strings, maybe it’s a column or timedate column.

Depending on these heuristics, Bigeye suggests a set of metrics for each table. You can find these suggestions on the \*\*Autometrics\*\* tab of the table's catalog page.



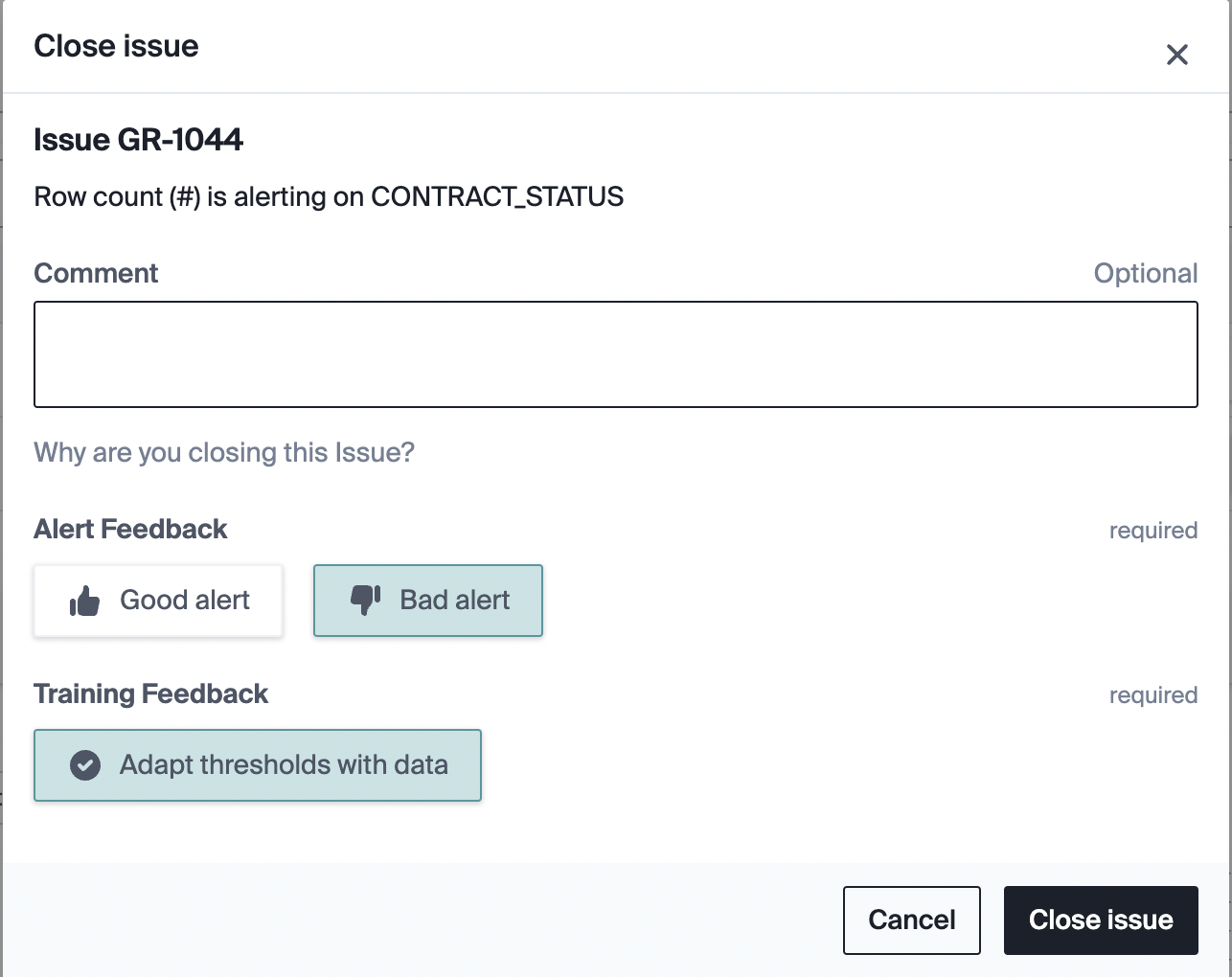
For more details on which metrics are available as Autometrics, and the criteria data must match in order for them to be suggested on a given column, review Bigeye's [Available Metrics](https://docs.bigeye.com/docs/available-metrics).

In addition to the metrics themselves, Bigeye also generates auto-thresholds, which are automatic thresholds calculated from historical patterns.



Auto-thresholds free you from having to manually set, tune, and update potentially thousands of sets of thresholds. Thanks to Bigeye’s anomaly detection engine, these thresholds are also dynamic – they adapt to business changes, seasonality, and your feedback.

For example, when a data issue notification is fired, but the user thinks that the data batch in question is actually good in practice, the user can tell Bigeye that the underlying data state is tolerable or that a false positive alert is present. Bigeye will take this information into account so that similar behavior in the future will not trigger an alert.



#### Turn on the column-level metrics that you actually care about!

Unlike other data observability vendors, Bigeye allows you to pick and choose which column-level metrics on which tables you want to enable, rather than forcing you to enable all of them on all tables.

This allows the data team to avoid alert fatigue by focusing on the columns that are important to them and ignoring the ones that are not.

