**What is Anomaly Detection?**

Anomaly detection refers to using statistical models to identify unexpected or unusual events in data. Anomaly detection is a well-studied academic field within statistics and machine learning, and in recent years has found many applications in industry – from systematic trading to fraud detection to observability and now, data observability.

**Why is Anomaly Detection Critical for Data Observability?**

Data observability consists of monitoring, alerting, metadata, and lineage.

Anomaly detection plays a critical role in the monitoring and alerting component of data observability. Since data observability is about knowing the state of your data systems at all times, and reacting appropriately when there is a problem, anomaly detection helps by identifying anomalies in data that might correspond to a data outage.

**Why manually set rules aren’t always sufficient**

The main alternatives to anomaly detection for data observability are **simple, manual thresholds and rules**. For example, for a metric that tracks the percentage of successful Airflow jobs, we might set a manual threshold of 90%. If more than 10% of Airflow jobs failed over a certain period of time, then we’ll get an alert.

Simple, manual thresholds work for many scenarios, in particular in systems observability, where metrics are expected to remain constant over time. However, they also have a number of drawbacks:

1. They require manual setting and tuning, which is fine when you have one data metric but not when you have 1000.
2. They are inaccurate for metrics that are not constant.

**Why simple Anomaly Detection methods aren’t sufficient**

Some of the simplest anomaly detection methods are easy to grasp and implement, for example, **formulaic thresholds that are calculated using simple statistics**: if your current data point exceeds the mean + standard deviation of a certain lookback window, then let’s call the data point an anomaly.

This kind of proto-anomaly detection, while automatable, is limited in its efficacy. It doesn’t work on data that goes through phases or data where the current value is linked to the previous value. Moreover, most businesses are dynamic and changes are made incrementally. When these changes are rolled out, they can create huge jumps in data that render past means and standard deviations moot.

**Benefits of Bigeye’s anomaly detection for data observability**

Bigeye’s advanced anomaly detection goes beyond formulaic thresholds, using a mixture of sophisticated statistical models to pick up on anomalies in the data that would otherwise be missed. There are three main advantages: accurate detection, intelligent adaptation, and continuous improvement.

## Accurate Detection

Accurate detection is especially important in the data observability context because both false negatives and false positives are negative: if your anomaly detection algorithm fails to detect an anomaly that turns out to be a data outage, you’ve potentially lost customers and revenue. Conversely, if your anomaly detection algorithm detects an anomaly that turns out not to be a data outage, but alerts on-call engineers anyways, over time, you erode their trust in observability systems.

Bigeye’s advanced anomaly detection uses both forecast and non-forecast models to detect hard-to-detect anomalies. This includes tricky cases like slow degradation, where a metric trends upwards or downwards slowly enough to go undetected. This can be problematic – imagine if the slowly downtrending metric is revenue, or number of requests!

Additionally, Bigeye’s anomaly detection understands trends and seasonality. If there’s a huge jump in the data, it can progress the history and the pattern by looking at user feedback.

## Intelligent Adaptation

Unlike in systems observability, where things are more or less expected to remain the same, data is inherently dynamic: “expected” values for data volume, for example, grow as the business grows.

Anomaly detection that can identify and adapt to these sorts of pattern changes removes the need for the data team to continually anticipate pattern shifts in the business and manually tune the data observability system.

This, in turn, enables data observability at scale. Bigeye’s auto-thresholds are calculated automatically, and are constantly updated, meaning that as a data scientist or engineer, you don’t have to manually set thresholds for potentially thousands of metrics. And once the metrics are being tracked, you don’t have to spend time re-tuning.

## Continuous Improvement

Out of the box, no data observability system is going to completely understand your business, or what your team cares about. Anomaly detection should learn from your team’s feedback to become more accurate over time. Bigeye’s anomaly detection has two features that assists with this: reinforcement learning and anomaly exclusion.

**Reinforcement learning** collects inputs from the data team to help to fine-tune detection and alerting. Perhaps the data team is more interested in knowing about extreme changes to data batches rather than small fluctuations. Or, perhaps the data team wants to understand if there’s any fluctuation in the machine learning feature store so that downstream automations are run with the most consistent inputs.

**Anomaly exclusion**, on the other hand, has to do with the way the anomaly detection system treats bad values. For example, let’s say there is an anomaly where the metric values drop significantly. The values of the metric during this anomalous period are not representative of the healthy state of the data infrastructure, and need to be excluded; otherwise, the anomaly detection system might become tuned to the anomalous state.

**Something is broken in the data - now what?**

Anomaly detection of data at rest is currently the prevailing philosophy of most data observability tools, but as a recent article pointed out, detecting anomalies alone is not sufficient. Once an anomaly has been identified, engineers want to be able to take action on them – to understand the root cause and assess the impact. What broke, why and where it broke – knowing the answers to these questions is critical for actually fixing the problem.

**Proactive anomaly detection**

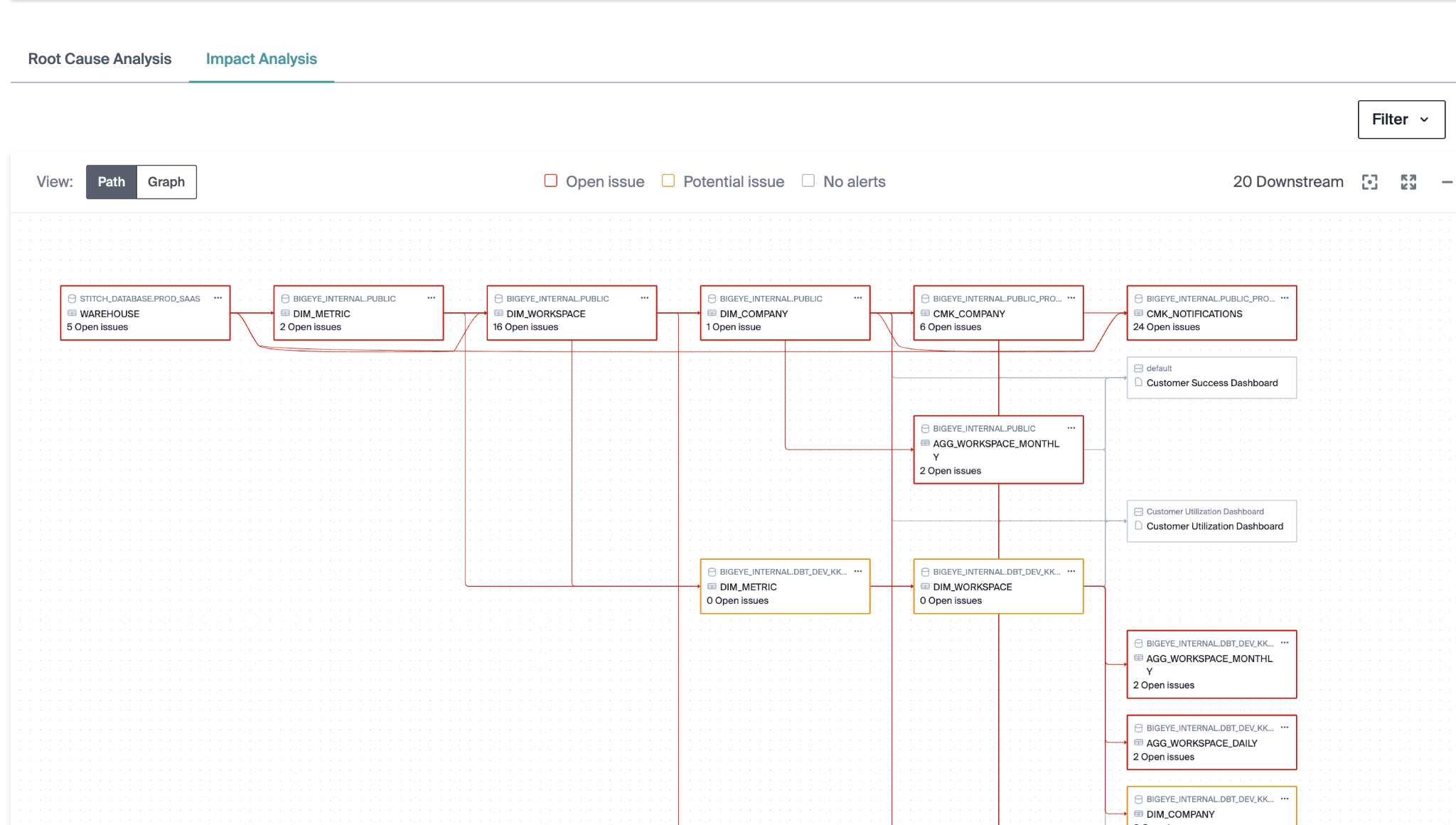
In answer to the “now what” question, Bigeye is making anomaly detection proactive by providing a UI that aids your investigations, including:

1. Root cause analysis reports

The root cause analysis tab on a Bigeye issue shows the upstream lineage of the alerting table and whether there are open issues on any of those tables.

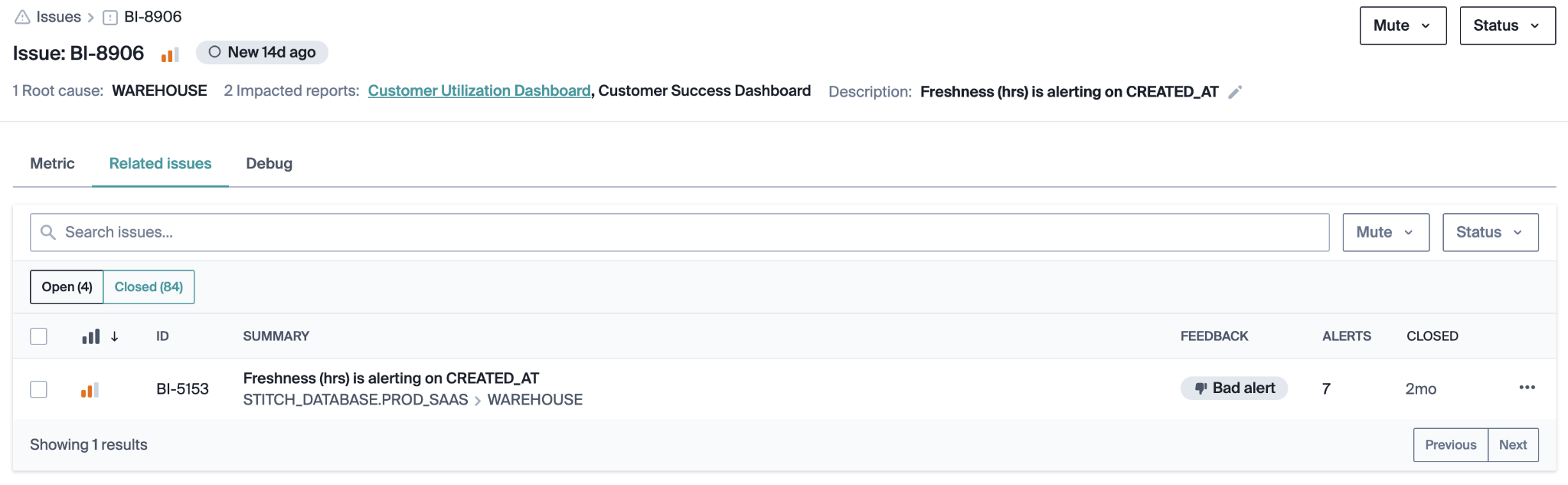
1. Impact analysis reports

The impact analysis tab, meanwhile, shows the downstream lineage of the alerting table – whether tables that depend on this one might also be affected.

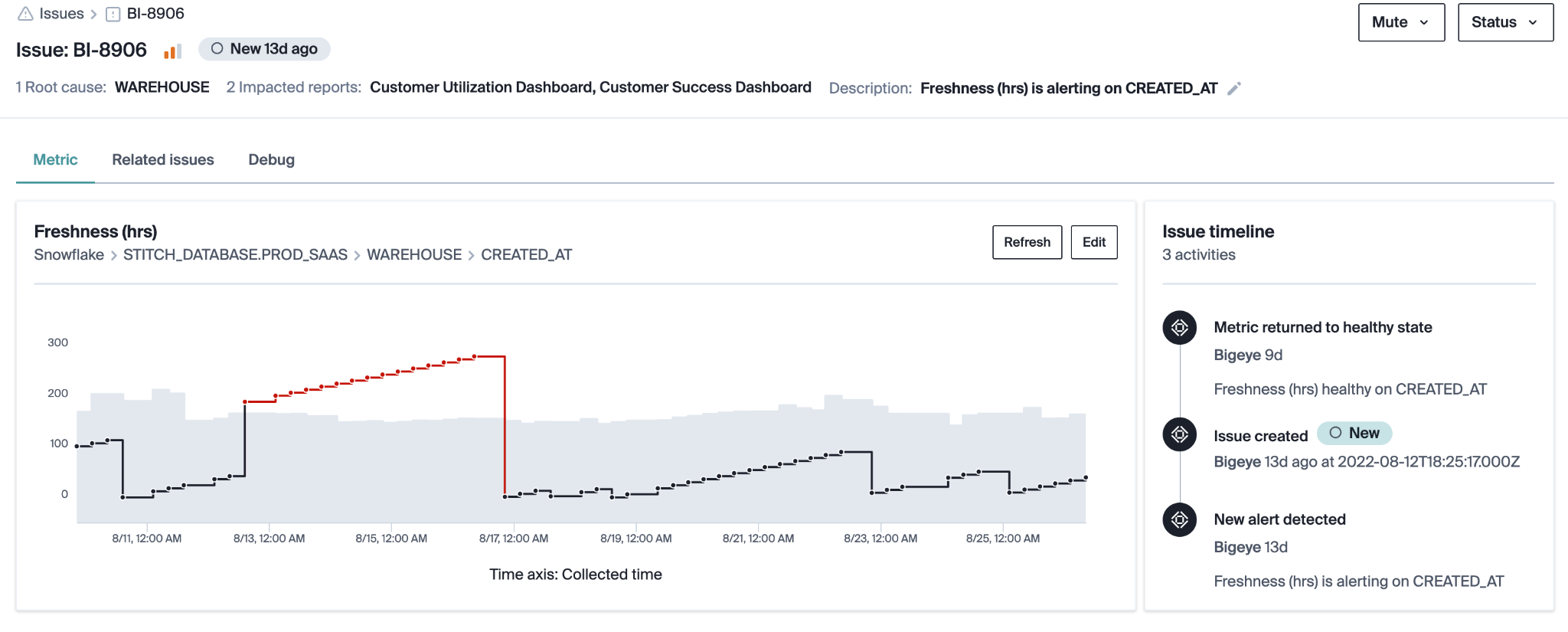


1. Related issues

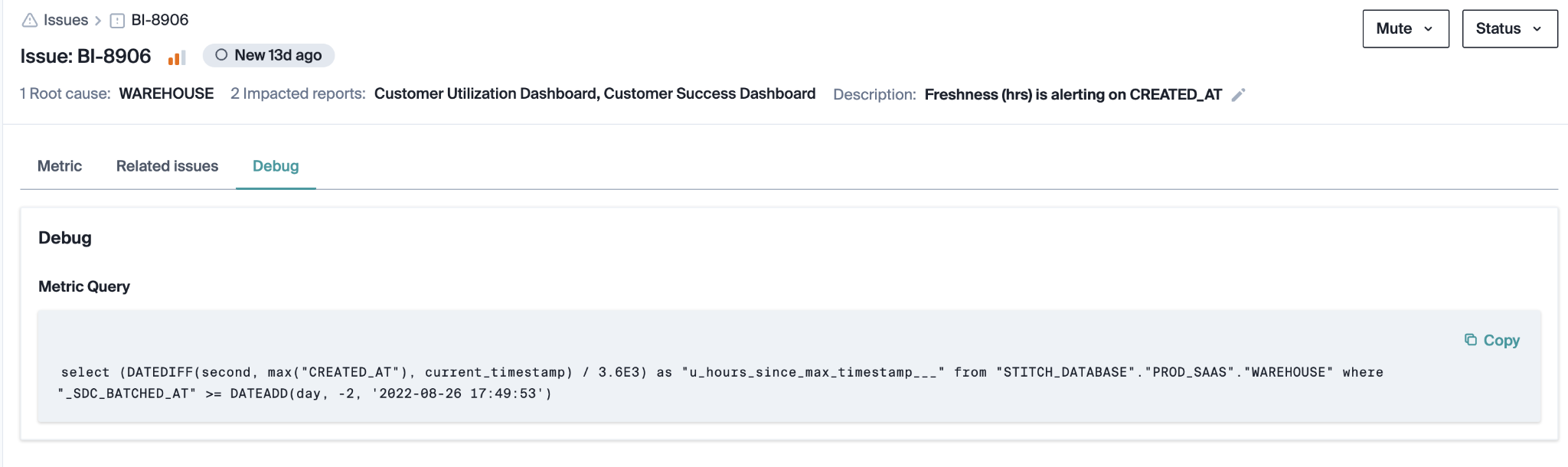
The related issues tab shows other issues that might be related (for instance if the same column alerted at some point in the recent past).



1. Graph and timelines of the alerting metric that give you a sense of relevant events at a glance



1. Copy-pastable queries that allow you to debug easily



**Conclusion**

When evaluating a data observability platform — whether to improve trust for self-service analytics, protect pipelines for machine learning, or validate third-party data — it's important to consider the strength of the anomaly detection under the hood. Bigeye’s anomaly detection combines sophisticated statistical models with intuitive UI to not only alert you when there’s an issue with your data, but help you resolve the issue.