# Have not received notification

Last updated by | Charlene Wang | Oct 14, 2020 at 8:37 PM PDT

Contents



## Have not received notification

The observation in the incident about not triggering until it resolves is correct. It will not send notification till it resolves. It stays triggered until the condition resets.

Alert jobs run regularly about every 5 min. So we don't send alert emails every 5 min to the customers. We send one when it activates and one when it resolves.

This explains why the alert was triggered 562 times during the weekend from the backend but you have not received many emails. Unfortunately we don't expose the logs for the alert jobs so there really is no way the customer will be able to tell.

The alert is like a sliding window that moves over time, every time the job runs the window moves forward to query metrics. If the metrics returned for that window cause the alert to trigger it starts and they get an email, it will stay triggered until the condition of the alert is not met in a future execution.

For the scale up metric that is not triggering because of the data and how the alert is defined. When we are querying SQL for metics for the last 10 minutes we get 2 values back from SQL, for example (95,0) these are the 2 metrics available during the period the alert runs. The 0 is because the data from the service has not yet made it thru the system by the time we requested it. This seems a known limitation of the SQL metrics that there is a latency on delivery of the data, problem is that it is variable and changes with load on their service. So when the alert runs it takes an average over these 2 values and ends up with 47.5.

In this case the real issue is that the alert time window is only 10 min while the metrics are only emitted every 5 min. So it is only seeing 2 values at best. And with one being 0 it is only 1/2 what is expected. Ideally we would not be getting 0's but that fairly common with SQL since it can take longer to get the data. One way to address this is to increase the window size so that it gets more data points, if they up the window size to 20 or 30 min the average will be closer to correct.

SQL are working on a MDM migration which will ideally address this issue, in that case we have less latency and a consistent way to address what latency there is.

Reference case: 117041415605598

**How good have you found this content?**