

Monitor database health in a dashboard

Published: 2020-02-22

When someone reports that a database query failed or is too slow, several questions come to mind. Finding the answers can be a time-consuming endeavour. You might spend hours combing the logs of database instances just to narrow down where the issue began.

With the ExtraHop platform, metrics about every database transaction on your network are located in one place. A dashboard in the ExtraHop system offers a central location for exploring this information and building helpful charts.

Dashboards are flexible and customizable, letting you build several types of charts that reveal different types of database metrics, which can shed light on underlying causes of database issues.

This walkthrough shows you how to build charts with the Metric Explorer and develop a dashboard to analyze database metrics. By modifying charts and selecting different metrics, you can explore data that brings you closer to the answers for the following questions:

- How many database errors do I have?
- When did the errors occur?
- What is the percentage of database errors on my network?
- Which servers are sending database errors?
- What are the slowest database instances?


Prerequisites

- You must have access to an ExtraHop Discover appliance with a user account that has limited or full write privileges.
- Your ExtraHop appliance must also have network data with database traffic. The ExtraHop system supports modules for major databases, including IBM DB2, IBM Informix, Microsoft SQL Server, MongoDB, MySQL, Oracle, PostgreSQL, Sybase ASE, and Sybase IQ.
- Familiarize yourself with the concepts in this walkthrough by reading the [Dashboards concepts](#) topic.

If you do not have access to database server data or the required privileges for the Discover appliance, you can perform this walkthrough in the [ExtraHop demo](#).

Create a dashboard

When you first log into the ExtraHop Web UI, you see a dashboard called the Activity dashboard. To create your own dashboard to display database metrics, complete the following steps.

1. On the bottom left of the Dashboard page, click **New Dashboard**.
2. In the Title field of Dashboard Properties, type a name for your dashboard. For this walkthrough, type **Database Health**.
3. Click **Create**. When you create a new dashboard, a workspace opens in an editable layout mode. This workspace contains a single region and two empty widgets: a chart and a text box.
4. Text box widgets can include custom explanatory text about a dashboard or chart. For this walkthrough, however, we won't be adding text. Delete the text box by completing the following steps:
 - a) Click the command menu  in the upper right corner of the text box widget and select **Delete**.
 - b) Click **Delete Widget**.

Next steps

Let's add database error metrics to the empty chart.

How many errors do I have?

These steps show you how to create a chart to display the number of database errors for a specified time interval.

Database errors are a general indicator of the stability and health of an application. In the ExtraHop system, database errors are the number of request operations that failed on all database instances. Any database error should be investigated.

To build the dashboard charts in this walkthrough, you'll select the All Activity application as the source. All Activity is a metric source that is available by default to all users and contains metrics about all of the devices discovered on your network.

1. Click the empty chart widget in your newly created dashboard to open the Metric Explorer.
2. Click **Add Source**.
3. In the Sources field, type **All Activity** to filter the results, and then select **All Activity**.
If you are building your dashboard in the Command appliance, select an All Activity application for a connected Discover appliance.
4. In the Metrics field, type **database errors** to filter results from all of the available metrics, and then select **Database Errors**.
5. From the bottom of the window, click the **Value** chart.

All database errors should be investigated.
A high number of errors can indicate negative effects on database performance.

The screenshot shows the 'Metric Explorer: Edit Chart' window. The left pane is titled 'Application Metrics' and contains three sections: 'Sources' with 'All Activity' selected, 'Metrics' with 'Database - Errors' selected, and 'Details' with 'Drill down by None'. A '+ Add Source' button is at the bottom of the left pane. The main area shows 'Last 30 minutes' and 'All Activity Database Errors' with a large '41,479 Errors' value. Below this is a chart selection grid with options: Area, Bar, Box Plot, Candlestick, Column, Heatmap, Histogram, Line, Line & Column, List, Pie, Status, Table, and Value (selected). The 'Value' chart shows a summary of 3.85M and 640K. At the bottom, there is an 'Open Metric Catalog' link, 'Cancel', and 'Save' buttons.

6. Click **Save**.

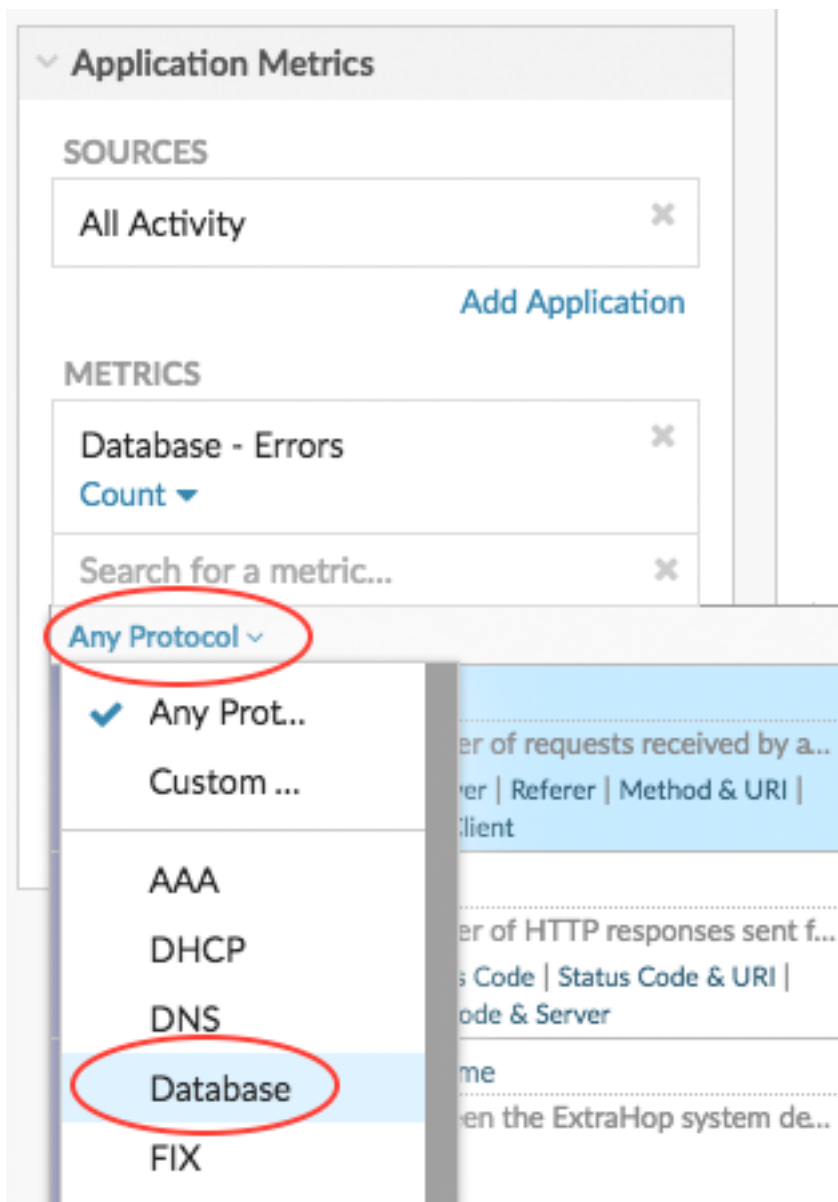


Tip: You can add a sparkline to your Value chart, which shows you how the number of errors changed over time. Click the **Options** tab at the top of the left pane and then select **Include sparklines**.

When did the errors occur?

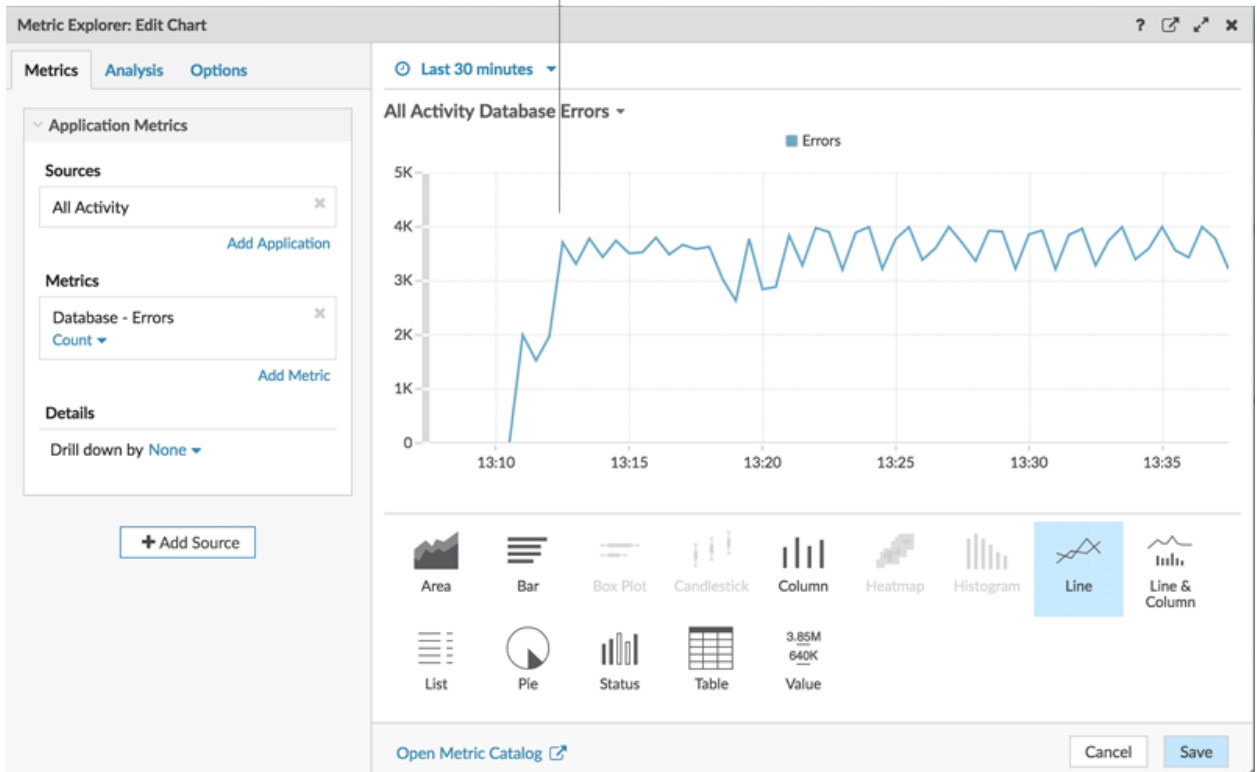
Now that you have determined the scope of database errors, let's take a look at when the errors occurred and how they changed over time.

1. Click and drag a new chart widget from the bottom of the page into an empty space on the region. If there is overlap, dashboard components are outlined in red, and you must click and drag the sides of the widgets and regions to make room.
2. Click the chart.
3. Click **Add Source** and select **All Activity**.
4. In the Metrics field, click **Any Protocol** and select **Database**. This shortcut helps narrow down your search for metrics by protocol.



5. Type `errors` to filter results and then select **Database Errors**.
6. From the bottom of the page, click the **Line** chart.

Look for when spikes occurred.



7. Click **Save**.

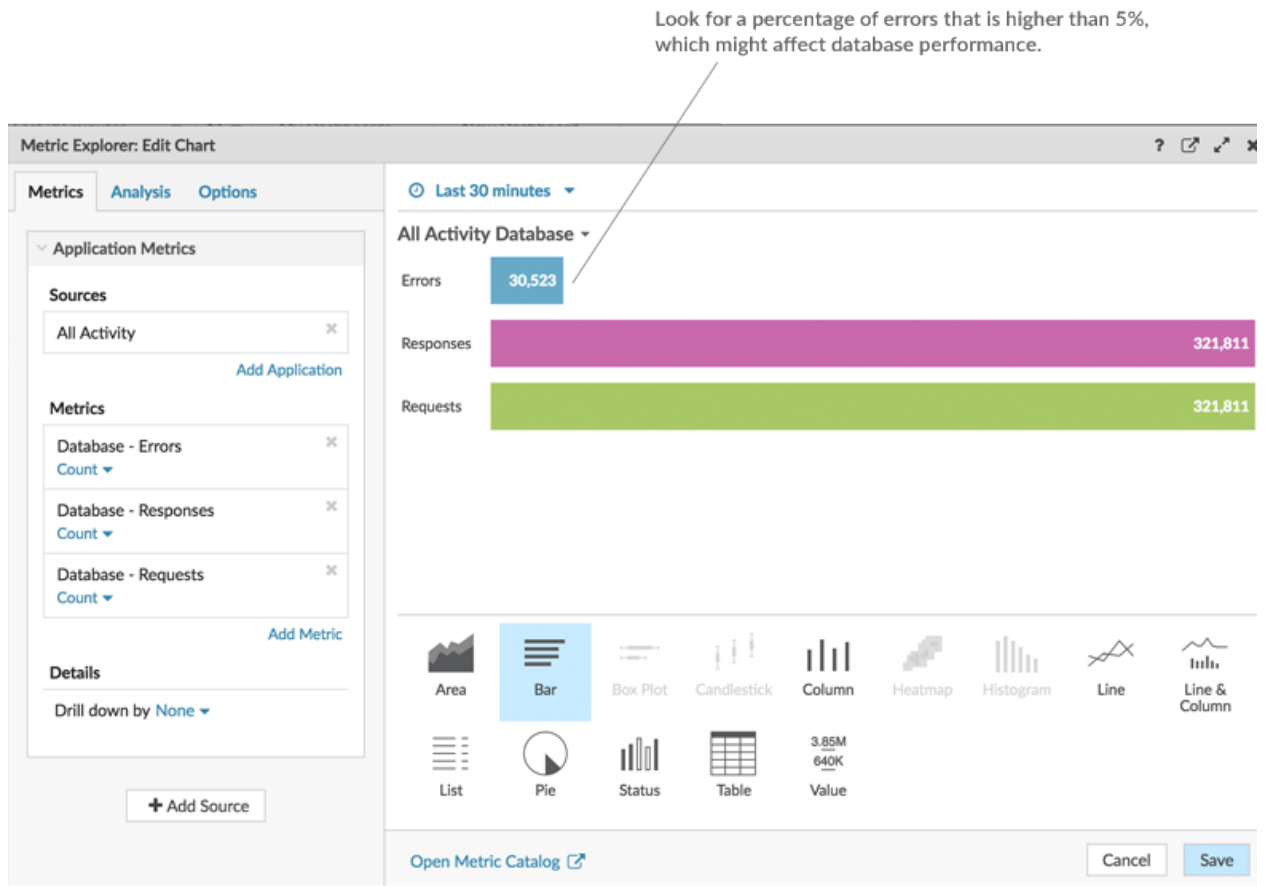
Next steps

Let's continue to add more database error charts to reveal a bigger picture about database errors on your network.

What is the percentage of errors happening on my network?

Comparing the number of database errors to the number of database transactions (requests and responses) can help you gauge the scope of issues on your network.

1. From the bottom of the page, click and drag a chart widget into the empty space.
2. Click the empty chart.
3. Click **Add Source**, select **All Activity**, and then select **Database Errors**.
4. Click **Add Metrics**.
5. Type `responses` and select **Database Responses**.
6. Click **Add Metric**.
7. Type `requests` and select **Database Requests**.
8. From the bottom of the page, click the **Bar** chart. You can now calculate the ratio of errors to transactions.



9. Click **Save**.

Next steps

You now have three charts that help you visualize the health of databases in your network. Next, let's add charts that help you drill into the cause of database errors.

Which operations methods are causing the errors?

Methods describe database operations. Uncovering the methods associated with database errors can help you determine the type of instance related to the database errors. If an operation is called against a table, the table name is displayed after an @ symbol.

1. From the bottom of the page, click and drag a chart widget into the empty space.
2. Click the empty chart.
3. Click **Add Source**, select **All Activity**, and then select **Database Errors**.
4. From the bottom of the page, click **Table**.
5. In the same chart you are editing, click **Drill down by <None>** from the Details section and select **Methods**. In a dashboard, you can drill down on a top-level metric, such as database errors, and view up to 20 of the methods that contributed to the overall number of database errors on your network.

Drill down on the database errors metric by method.

Look for the methods that generated the most errors. If a table name is available, it is displayed after the @ symbol.

Metric Explorer: Edit Chart

Metrics Analysis Options

Last 30 minutes

All Activity Database Errors by Method

Method	↓ Errors
SELECT @FAQ	29,089

Application Metrics

Sources

All Activity

Add Application

Metrics

Database - Errors

Count Any Error

Add Metric

Details

Drill down by Method

= Any

Top 5 results

Area Bar Box Plot Candlestick Column Heatmap Histogram Line Line & Column

List Pie Status Table 3.85M 640K Value

Open Metric Catalog

Cancel Save

6. Click **Save**.

Next steps

You now have a chart that helps you track database errors to specific operations. Next, let's track a metric about database performance.

Evaluate server processing times by server

Let's explore the performance of database instances by looking at the server processing time metric. In the ExtraHop system, server processing time is calculated as the time it takes the server to send the first packet of a response after receiving the last packet of a request. High server processing time can indicate resource contention.

Before adding a processing time chart to your dashboard, let's first add another region to the dashboard to organize the charts into logical groups.

1. On one of the charts, click the command menu in the upper right corner.
2. Hover over **Copy to...** and select the name of your dashboard from the menu. The most recently created dashboards are listed at the bottom of the flyout menu. This step creates a copy of the chart in a new region.

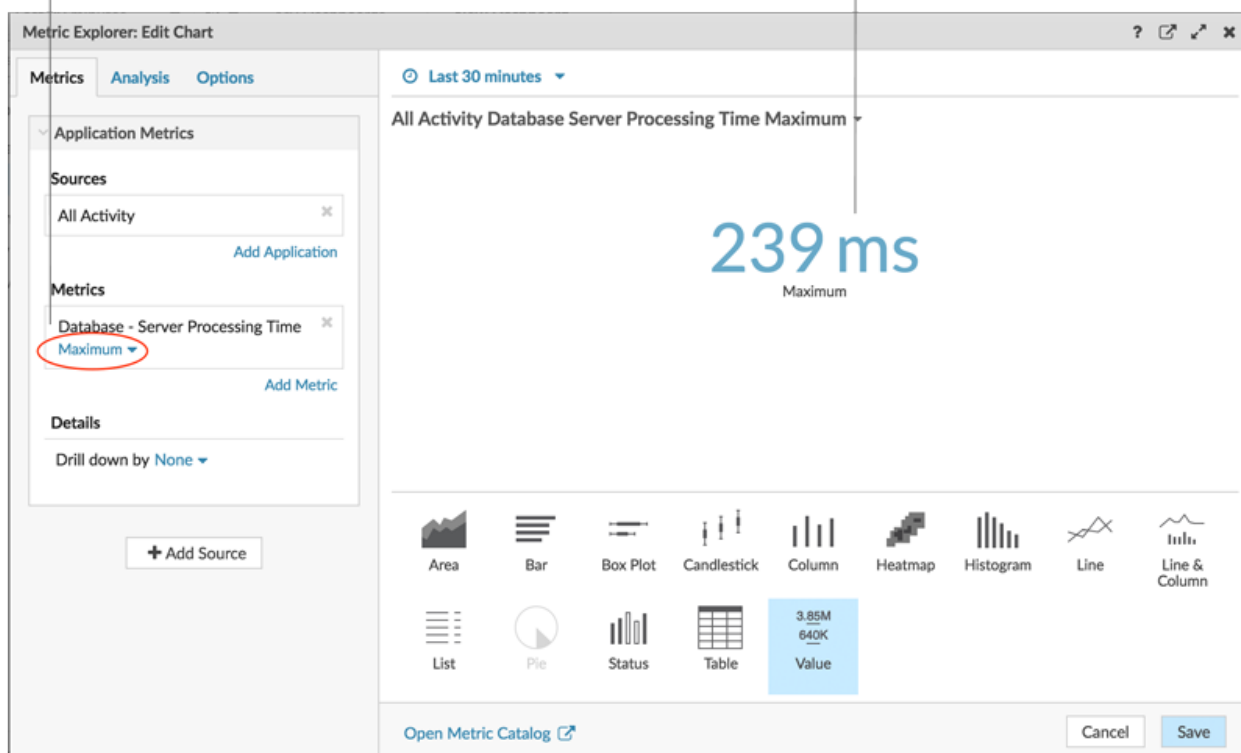
Method	↓ Errors
SELECT @FAQ	37,948
INSERT @ORDERS	1

- In the new region, click **Rename**. Type Database Performance and then click **Save**.

- Click the chart.
- Click the **x** icon next to the current metric and drill down selections.
- Click **Add metrics** and then select **Database Server Processing Time**.
- Click **Mean** and then select **Maximum**.
- At the bottom of the window, click the **Value** chart. This value represents the slowest time a database instance took to process a query and prepare the response.

Click Mean or Maximum to change the data calculation.

A maximum server processing time over 100 milliseconds can indicate slow database performance.



9. Add a new chart to the Database Performance region. From the bottom of the page, drag and drop a chart widget into an empty area.
10. Click **Add Source**, select **All Activity**, and then select **Database Server Processing Time**.
11. At the bottom of the window, click the **Bar** chart.
12. Click **Maximum** and then select **Mean**.
13. Click the chart title and select **Rename**. Type **Processing Time by Server** in the custom name field and then click **Save**.
14. In the Details section, click **Drill down by <None>** and then select **Server**.

Select Mean to view the average processing time of each database server in your network.

Drill down on the database server processing time metric by server.

Click the chart title to rename the chart.

Metric Explorer: Edit Chart

Metrics Analysis Options Last 30 minutes

All Activity Database Server Processing Time by Server Mean

mysql1-nyc 4 ms

mysql1-lon <1 ms

mysql1-sea <1 ms

Sources: All Activity

Metrics: Database - Server Processing Time (Mean)

Details: Drill down by Server (Any IP Address)

Top 5 results

Area Bar Box Plot Candlestick Column Heatmap Histogram Line Line & Column

List Pie Status Table 3.85M 640K Value

Open Metric Catalog Cancel Save

Note: To display more servers, type a larger number into the Top results field. You can view up to 20 drill-down items in a dashboard chart.

15. Click **Save**.
16. At the upper right corner of the dashboard page, click **Exit Layout Mode**.

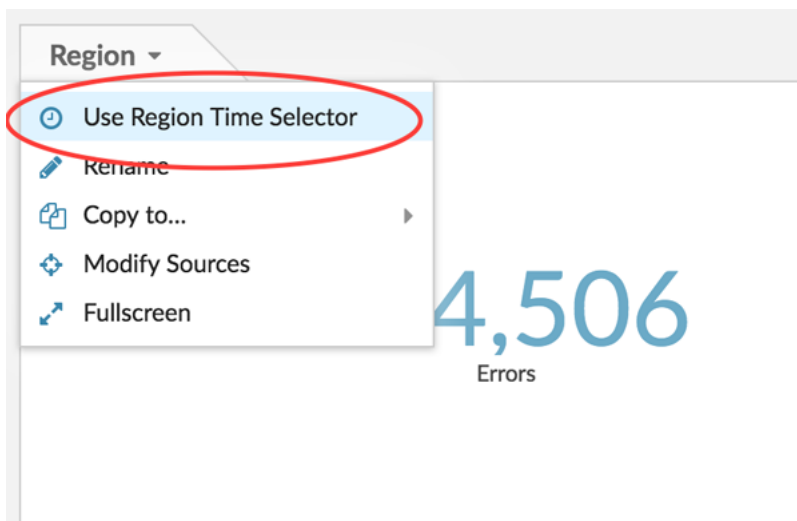
Next steps

Your dashboard is complete! You can now monitor the general health of database transactions on your network. The following sections offer additional tips for analyzing database metrics from your dashboard.

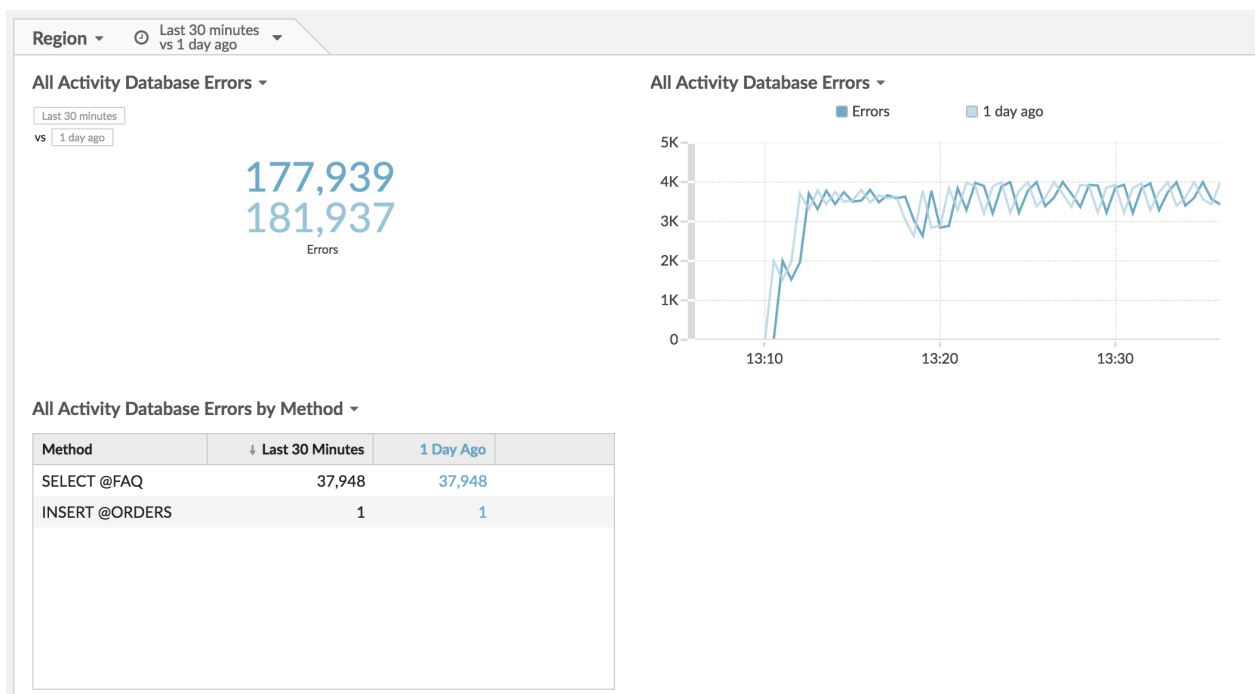
Compare different time intervals

By applying a delta comparison of time intervals to your charts, you can see changes in data from two time intervals side-by-side.

1. Click the region header and select **Use Region Time Selector**.



2. In the region header, click **Last 30 Minutes**.
3. Near the bottom of the time interval window, click **Compare**. You can now select two intervals to perform a delta comparison of metrics from each time period. For this example, let's compare metrics from yesterday to the last 30 minutes.
4. Click **Save**. You will now see the delta comparison of metrics in all charts within the region.



Note: You can perform a delta comparison for the entire dashboard by changing the global time interval. The global time interval is located at the top left corner of the dashboard page.

5. To remove the delta comparison, click **Last 30 minutes vs 1 day ago** in the region header, click **Remove Delta**, and then click **Save**.

Additional database metrics to monitor

Database errors and server processing time are a couple of sources of information about the health of database traffic. Here are suggestions for other metrics that you can add to your dashboard to answer the following questions.

Question	Database metric	Database metrics to add to your dashboard
Who is connecting to a database instance?	Database Requests, drill down by User	The ExtraHop system tracks who creates a session (connection) with a database server instance. A database user is determined during the negotiation phase of a connection to a database instance. In the ExtraHop system, database users are listed by login name, or as anonymous, unknown, and pre-login.
Are database servers experiencing a significant load?	Database Requests	Requests cover a range of operations: connection negotiations, session configuration, data definition language (DDL), data modification language (DML), or data reads (select). The number of requests provide insight into the load on a database instance. Fluctuations in requests can indicate when a service might become resource-constrained.
Is the network affecting database transactions?	Database Round Trip Time	Round trip time (RTT) is calculated by observing the time it takes for packets to travel across the network between the database server and client. A high RTT can indicate network latency.