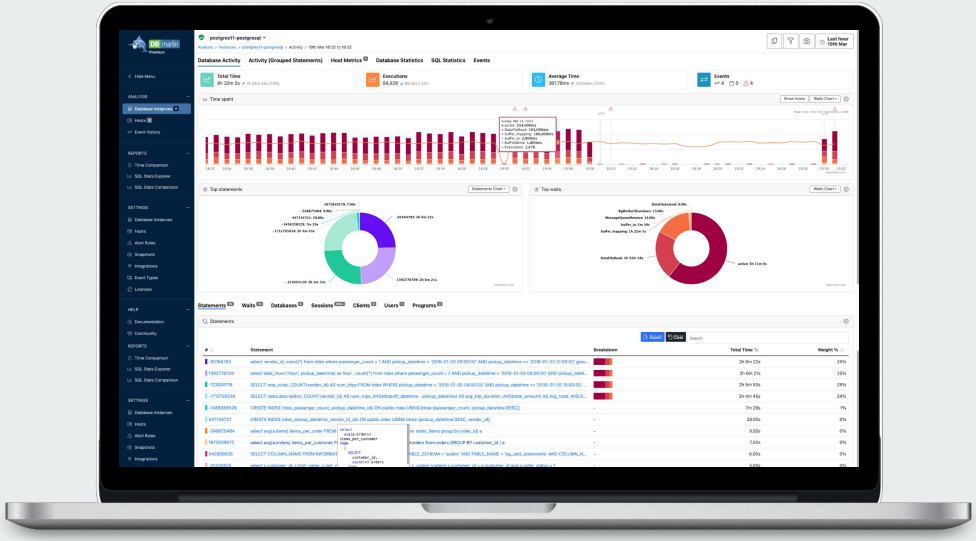
DBmarlin Cookbook



Worked examples to help you understand more about DBmarlin

Copyright © 2026, Application Performance - Confidential

Cookbook

1. Find Performance Degradation
2. Analyse SQL Statement performance
3. Analyse wait states
4. Troubleshoot database locking
5. Configure DBmarlin co-pilot
6. Use DBmarlin co-pilot to tune SQL Query
7. Use DBmarlin co-pilot to understand wait states
8. Use KB to understand wait states
9. Select a time period for analysis
10. Take a snapshot of database performance
11. Compare database activity between snapshots or time periods
12. Explore SQL Statistics
13. Configure SQL Statistics
14. Compare SQL Statistics over time or between instances
15. Review Recommendations
16. Explore Database versions
17. Configure https access to DBmarlin
18. Update DBmarlin
19. Collect logs
20. Use tags to classify database instances
21. Configure Authentication
22. Configure Role-based Access Control
23. Configure Single sign-on (SSO)
24. Ask an AI assistant about database performance
25. Monitor SQL Server Always On availability groups
26. Monitor SQL Server backups
27. Find repeated query patterns with Grouped SQL
28. Monitor Oracle multitenant (CDB and PDBs)

Copyright © 2026, Application Performance - Confidential

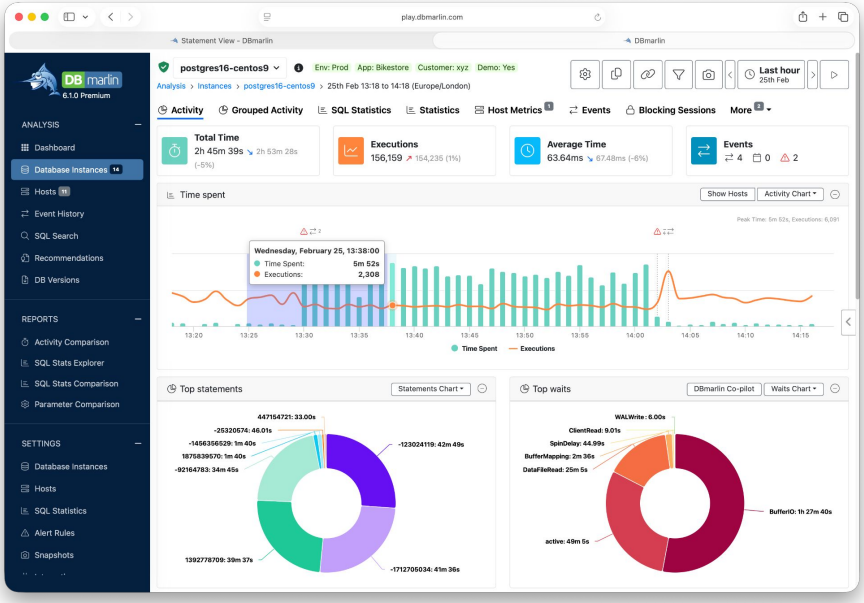
Using the cookbook

All screenshots and steps in the cookbook have been completed on the DBmarlin play site: https://www.dbmarlin.com/play

Using this, you should see identical results.

For some of the screenshots, I selected a time period where performance for the postgres16-centos9 instance gets worse.

1. Click on the postgres16-centos9 database in the main dashboard
2. On the main timeline, drag to select time where green bars can hardly be seen to where green bars are seen



Copyright © 2026, Application Performance - Confidential

1. Find Performance Degradation

Problem

You want to quickly find the performance degradation amongst many different databases and many different database platforms so that the problem can be fixed rapidly and accurately, minimising time to recovery.

Solution

Open up Database Instances for all databases. Look for “Average Time” - It represents the time the database takes to run queries. You can see rising trends. Also look for “Executions” - A sudden drop or sudden rise could indicate a problem in the application or website layer.

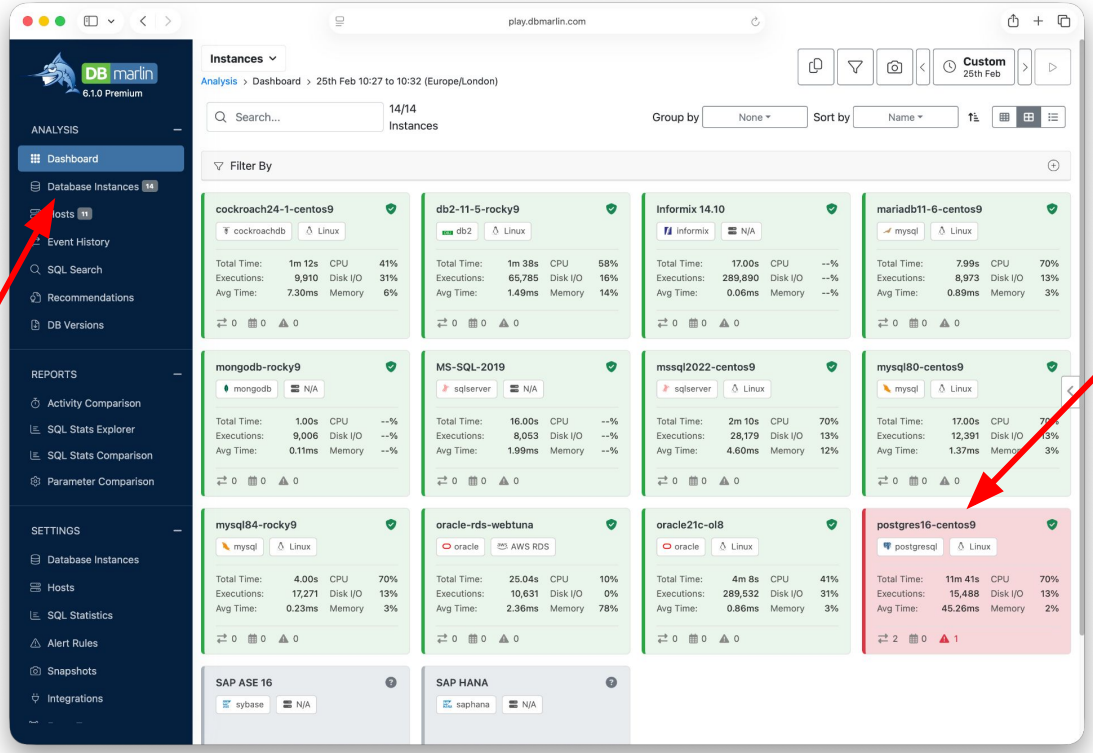
Also look for Alerts, Changes or Events, these can indicate likely correlations.

Copyright © 2026, Application Performance - Confidential

Find Performance Degradation

Dashboard

Click on Database Instances to see the databases in a list



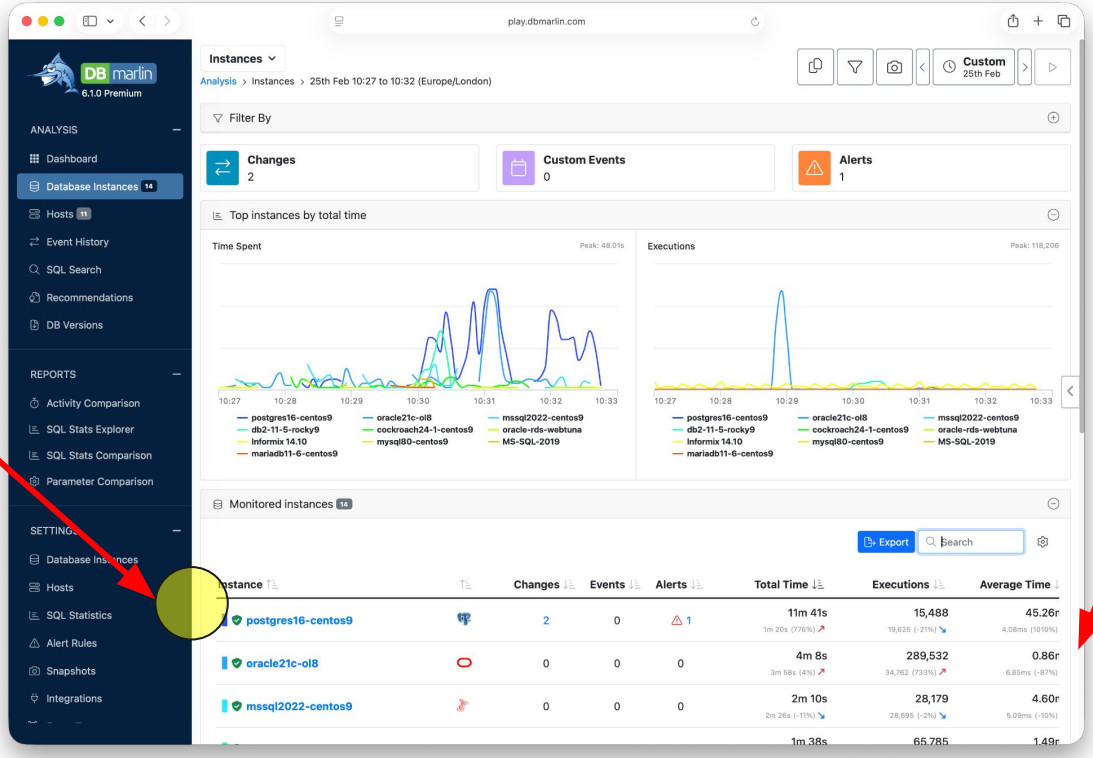
Database of interest is coloured Red.

Copyright © 2026, Application Performance - Confidential

Find Performance Degradation

Database Instances

Hide left menu - hover around the highlighted area, an arrow should appear to collapse the menu.

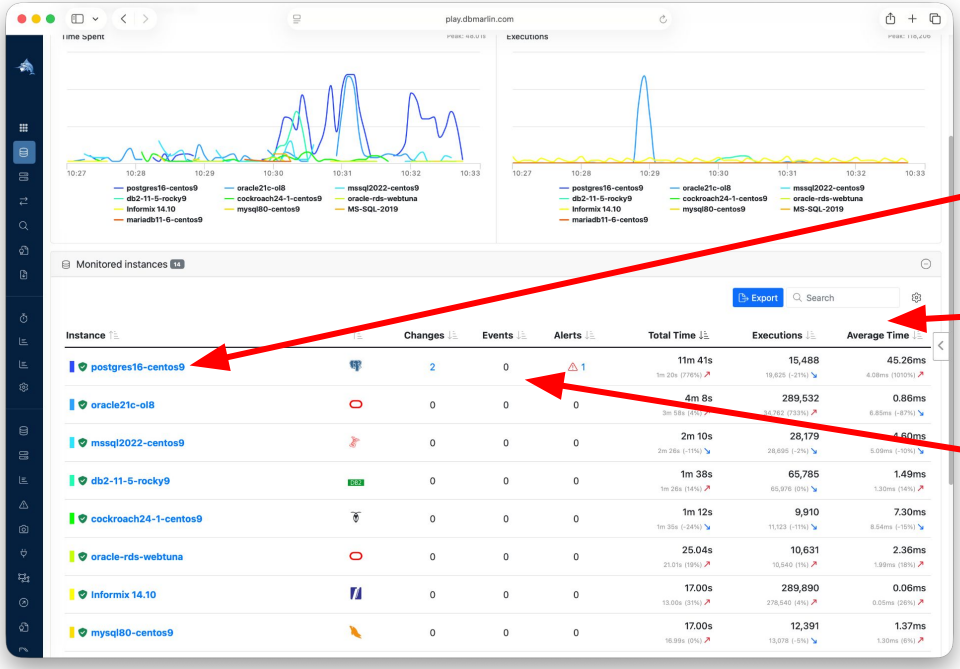


Scroll down for a deeper look at the instances

Copyright © 2026, Application Performance - Confidential

Find Performance Degradation

Take a Look at top item



postgres16-centos9 instance

Average time 45.26ms up 1010%

2 changes detected
1 alert

Copyright © 2026, Application Performance - Confidential

2. Analyse SQL Statement Performance

Problem

You want to understand why a SQL Statement is running slowly. If it has recently got worse, you want to understand if a recent change has caused a problem.

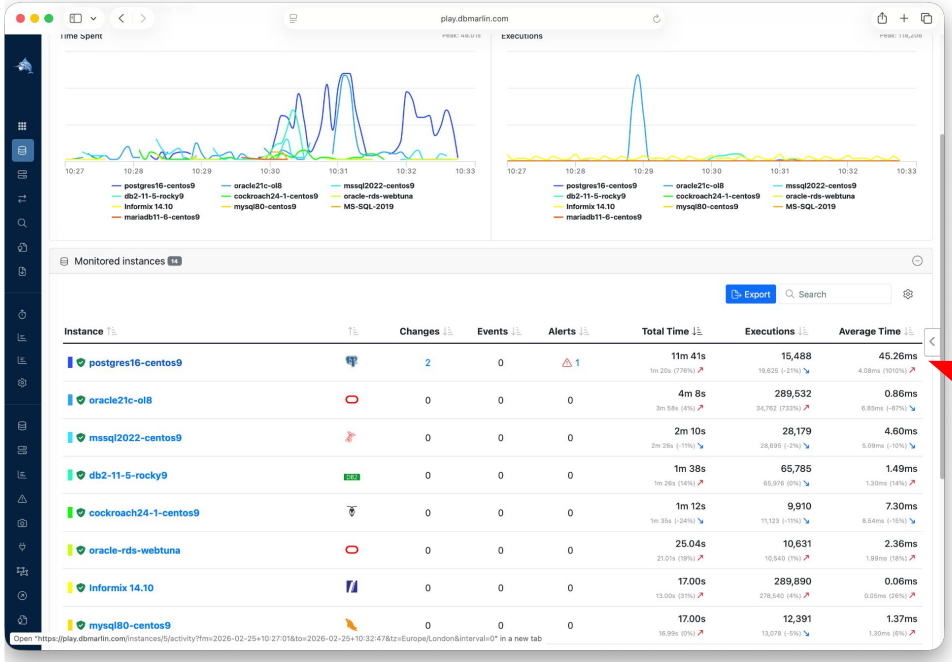
Solution

- Select the statement of interest (DBmarlin presents the statements with the highest wait time first)
  DBmarlin will present the statement and the query execution plan. There may be more than one plan used during the selected time period. If the plan is less efficient, then this can help to diagnose the problem
- Optionally, use DBmarlin Co-Pilot to use AI to offer suggestions as to the cause and possible solutions.

Copyright © 2026, Application Performance - Confidential

Analyse SQL Statement Performance

Select the Database of Interest

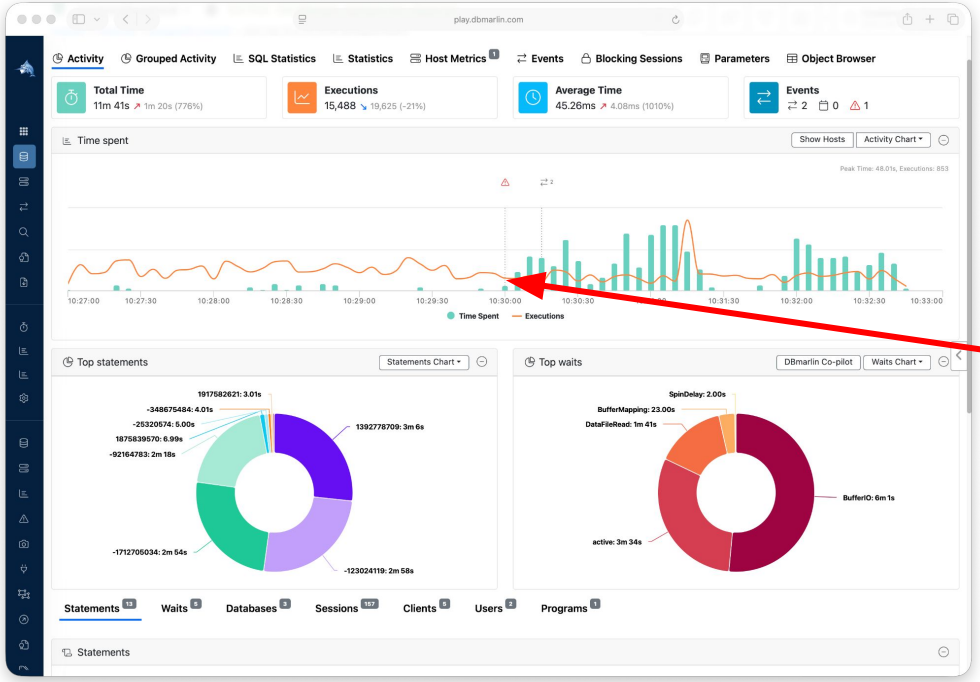


In this case the top database (postgres16-centos9) is interesting
Average time highest and rising

Copyright © 2026, Application Performance - Confidential

Analyse SQL Statement Performance

Select the Statement of Interest

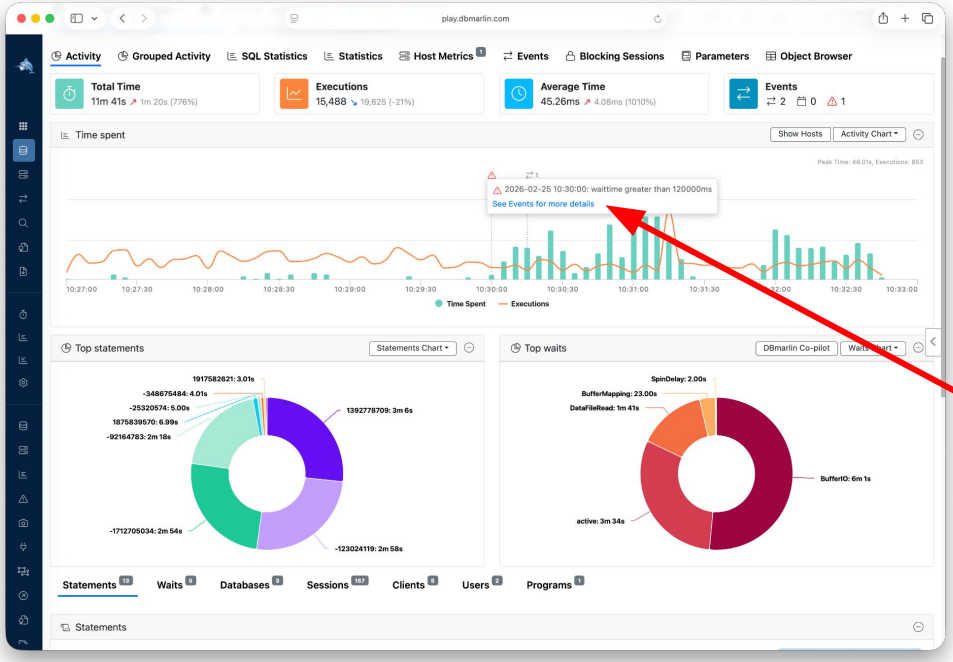


Timeline shows increase in time from 10:30 onwards

Copyright © 2026, Application Performance - Confidential

Analyse SQL Statement Performance

Examine alerts on timeline

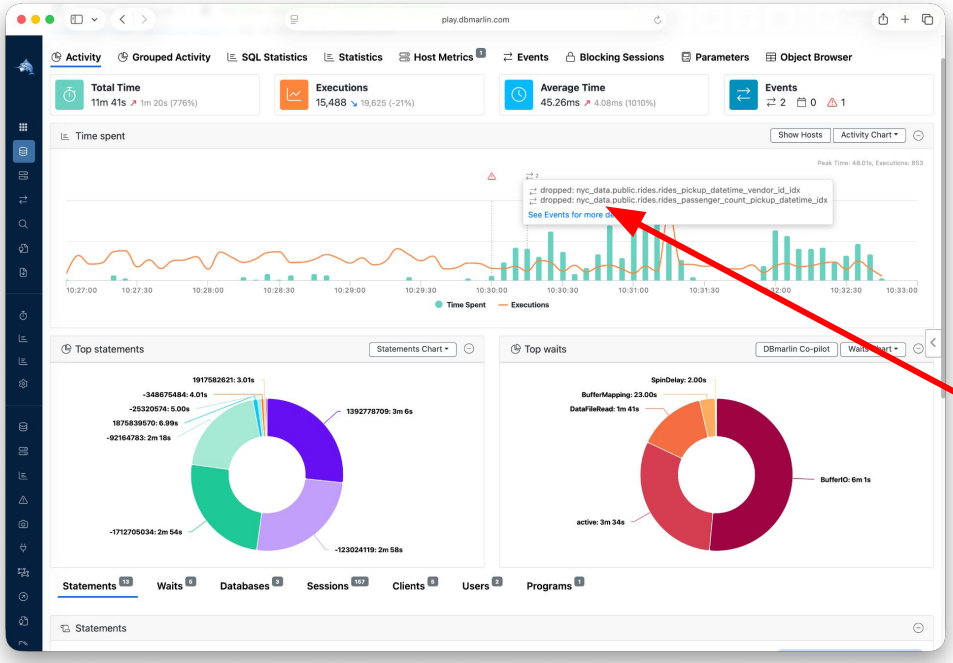


Hover over warning triangle in timeline to see alert detail. *Alert for waittime greater than 120000ms. Link is available to see events as a list.*

Copyright © 2026, Application Performance - Confidential

Analyse SQL Statement Performance

Examine changes in timeline

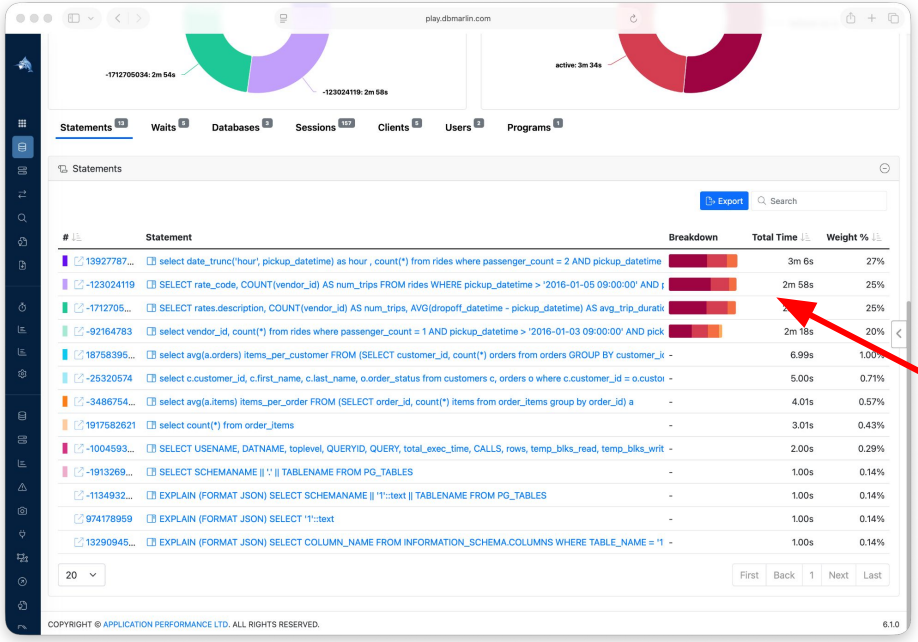


Hover over bidirectional arrows on timeline.
2 Indexes have been dropped
Changes are frequently cause of performance issues

Copyright © 2026, Application Performance - Confidential

Analyse SQL Statement Performance

Select the statement of interest

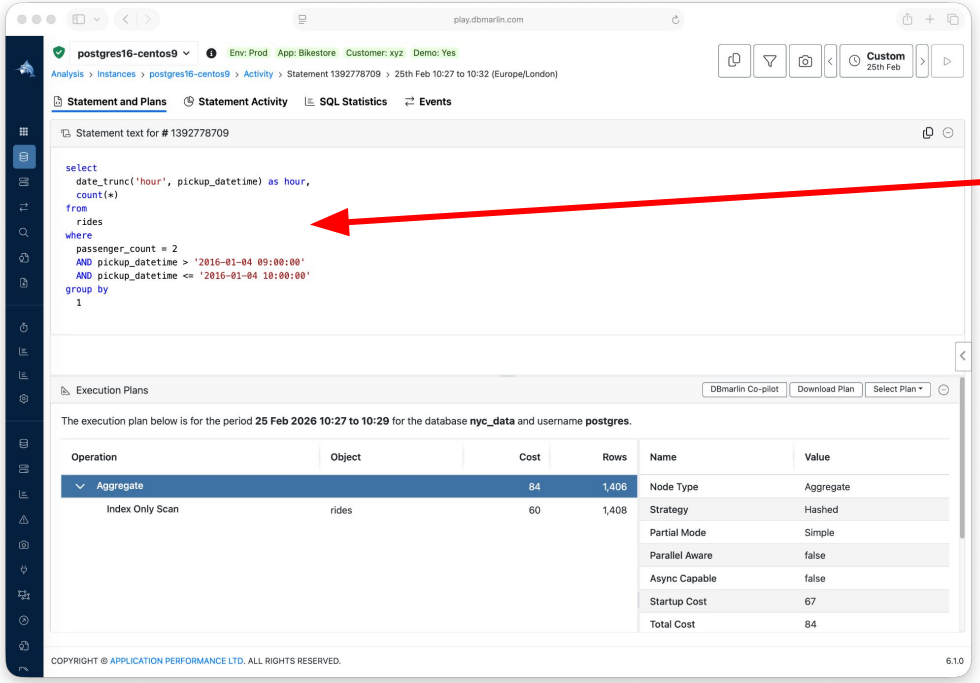


Select statement of interest. This list is sorted by Total Time, so most interesting statements are at the top.

Copyright © 2026, Application Performance - Confidential

Analyse SQL Statement Performance

Review the statement and plan(s)



The statement is presented
Along with the execution plan

Note: DBmarlin automatically collects statement execution plans for you.

Copyright © 2026, Application Performance - Confidential

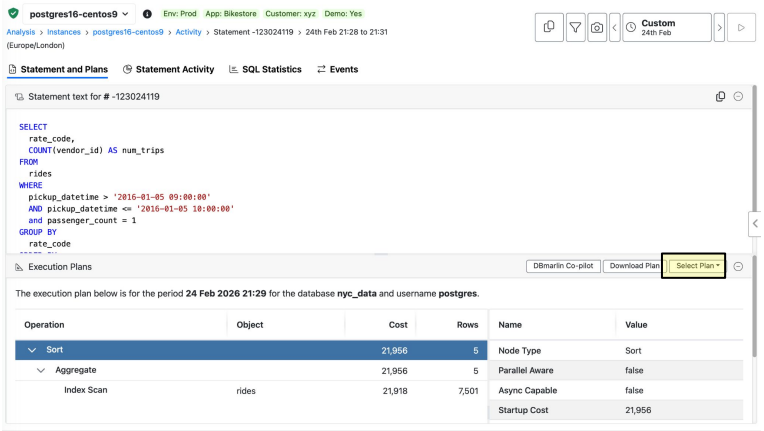
Analyse SQL Statement Performance

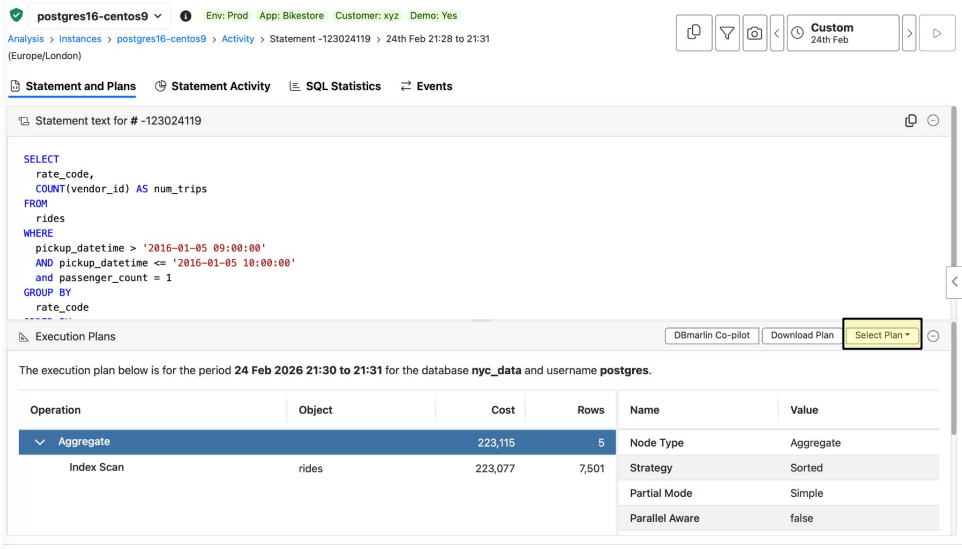
Examining a Query Execution Plan

In this case, there are two execution plans. Click the “Select Plan” button (highlighted in illustration) to select the other query plan. The cost shows which plan is slower than the other.

The one on the right has a cost of 223,115 compared to the one on the left with a cost of 21,956

It seems that the change (dropped indexes) means that the database executes the query much more slowly than before.

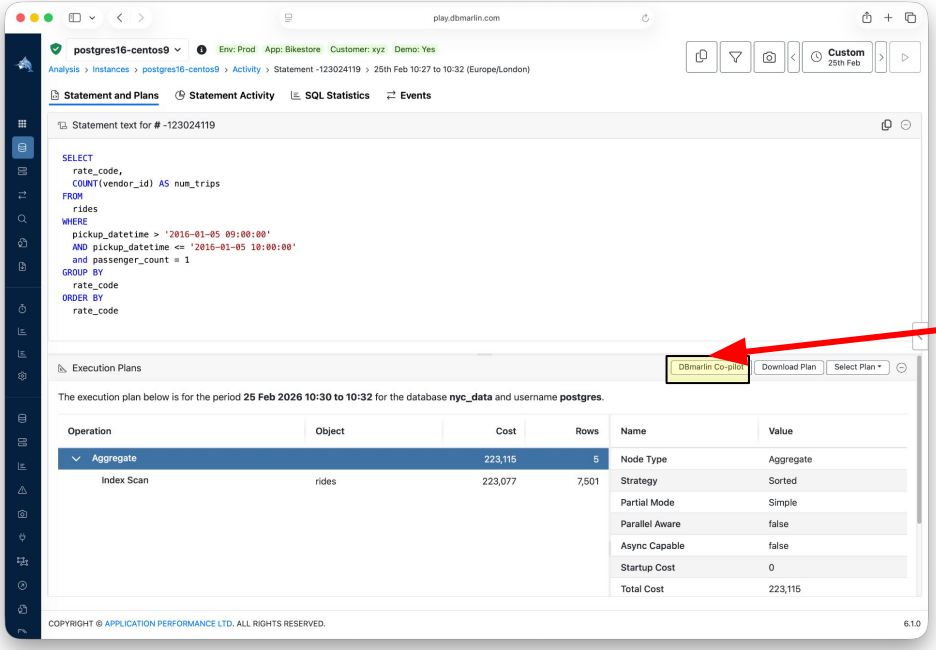




Copyright © 2026, Application Performance - Confidential

Analyse SQL Statement Performance

Use DBmarlin co-pilot to suggest cause and remediation

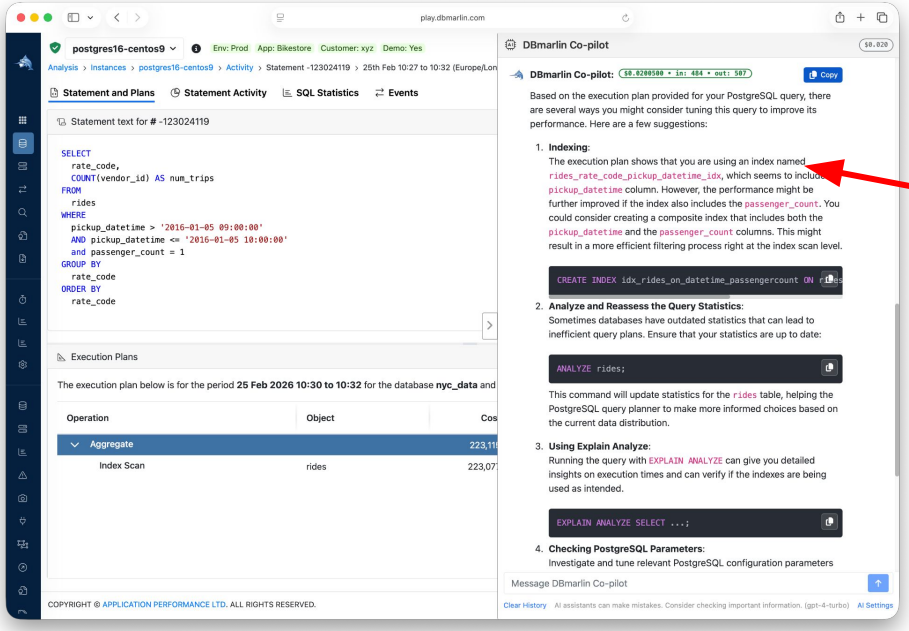


Click on DBmarlin co-pilot. DBmarlin co-pilot is given information about the query and the plan, and comes up with an AI view of the efficiency, and what could be improved.
Even for expert DBAs, this capability is useful, especially with new database platforms. It helps a DBA to transfer their skills to a new platform.

Copyright © 2026, Application Performance - Confidential

Analyse SQL Statement Performance

View the DBmarlin Co-pilot output



DBmarlin Co-pilot has correctly identified the index that would need to be added to improve performance.
The `psql` command to make the recommended changes can be copied / pasted.
You can ask DBmarlin Co-pilot any follow on questions that you would like.

Copyright © 2026, Application Performance - Confidential

3. Analyse wait states

Problem

Databases spend most of their time either executing work or waiting for resources (I/O, locks, CPU or memory). You want to understand wait states for the whole database or for an individual statement, so the database can run these statements more quickly.

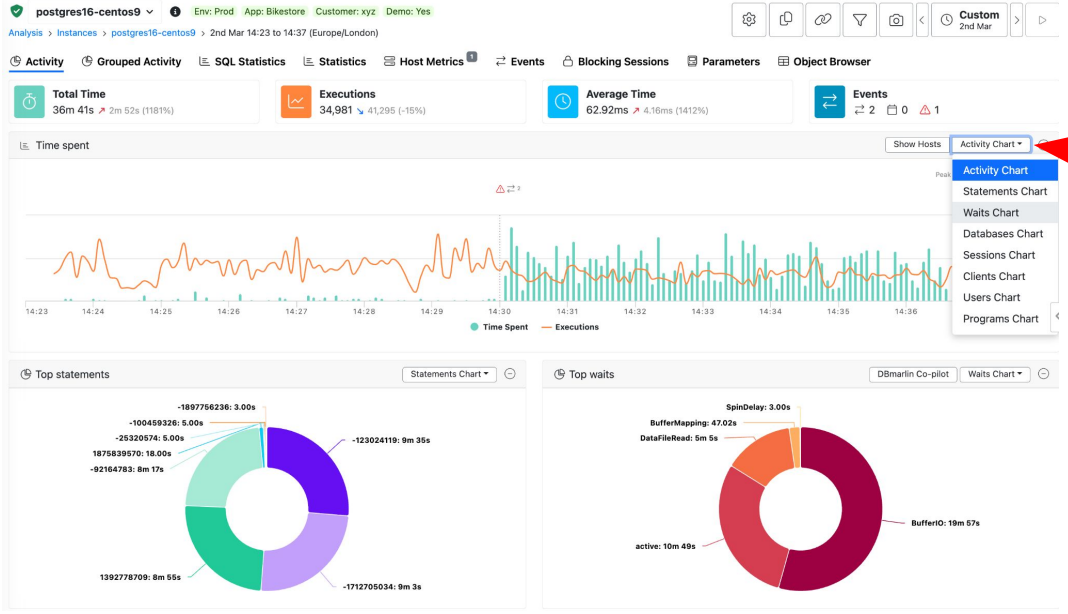
Solution

Either at the database level or the statement level, examine the wait states. If you are looking for inspiration as to what to look for, with the wait state selected, ask for assistance from either the knowledge base or the AI option with co-pilot

Copyright © 2026, Application Performance - Confidential

Analyse wait states

Put wait states into the timeline



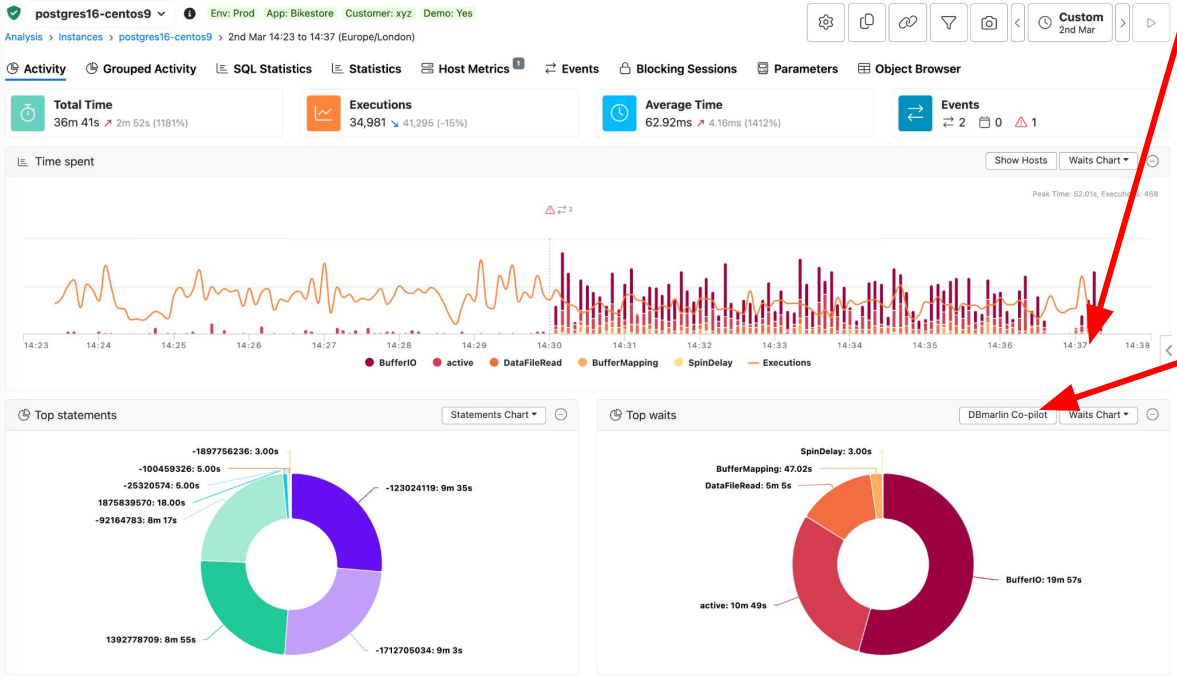
In the timeline, click the dropdown marked Activity Chart and choose Waits Chart

Copyright © 2026, Application Performance - Confidential

DB marlin

Analyse wait states

View wait states at database level.



Hover over bars to see distribution of waits.

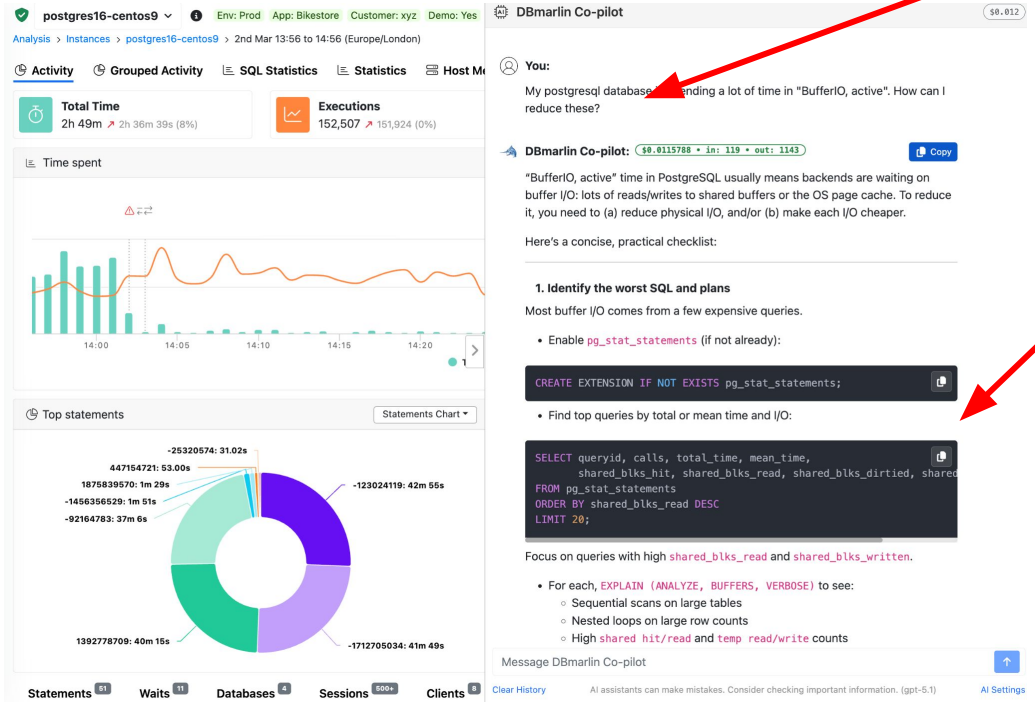
BufferIO is the most common wait state

In the Top Waits pie chart, click on “DBmarlin Co-pilot” to ask for help from AI

Copyright © 2026, Application Performance - Confidential

Analyse wait states

View DBmarlin Co-pilot results



This is the question sent DBmarlin Co-pilot

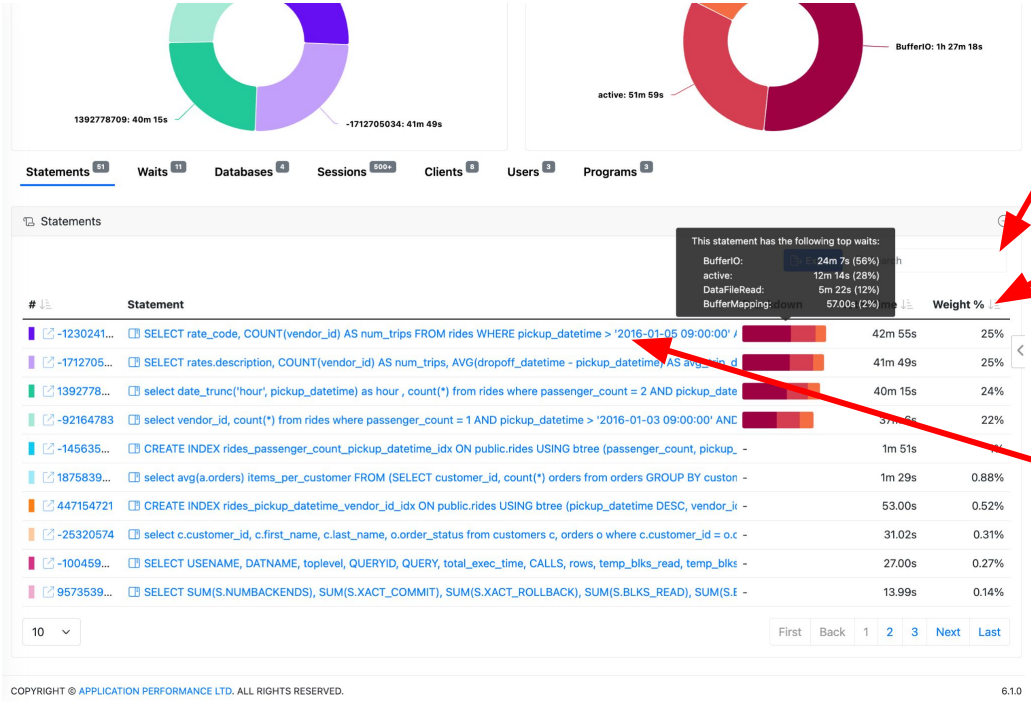
DBmarlin Co-pilot explains the problem very clearly, and offers advice for remediation

More details of DBmarlin Co-pilot in other sections

Copyright © 2026, Application Performance - Confidential

Analyse wait states

View wait states at statement level



Scroll down to see list of statements

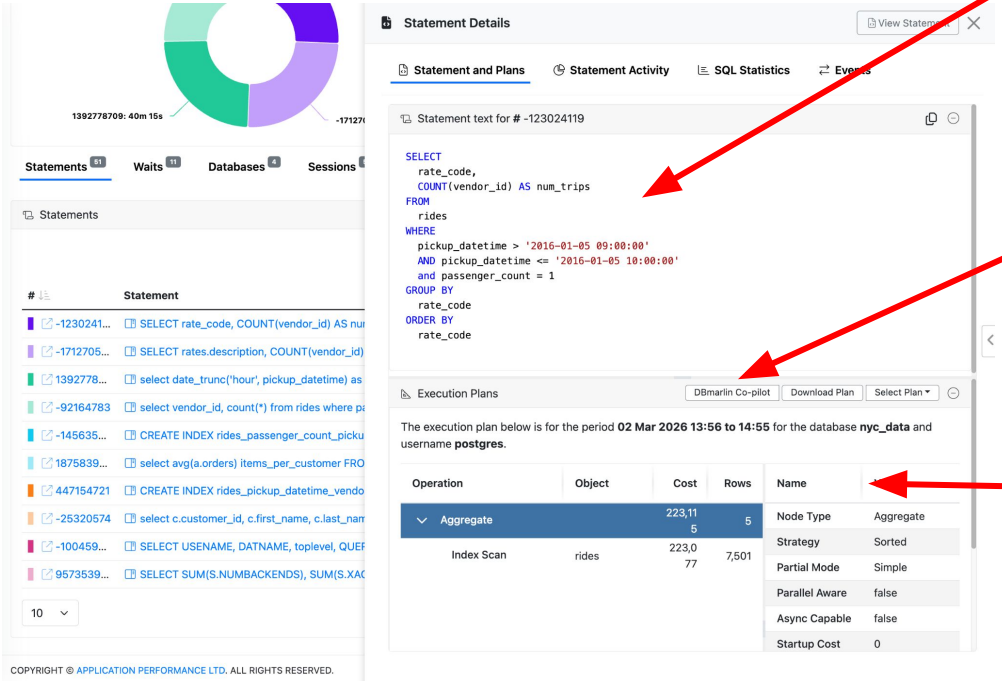
Top 4 statements have a breakdown of wait states for each statement. Hover over item in Breakdown column for details wait states.

Click on the top statement's SQL Query

Copyright © 2026, Application Performance - Confidential

Analyse wait states

View wait states at statement level



See statement text in clear indented form

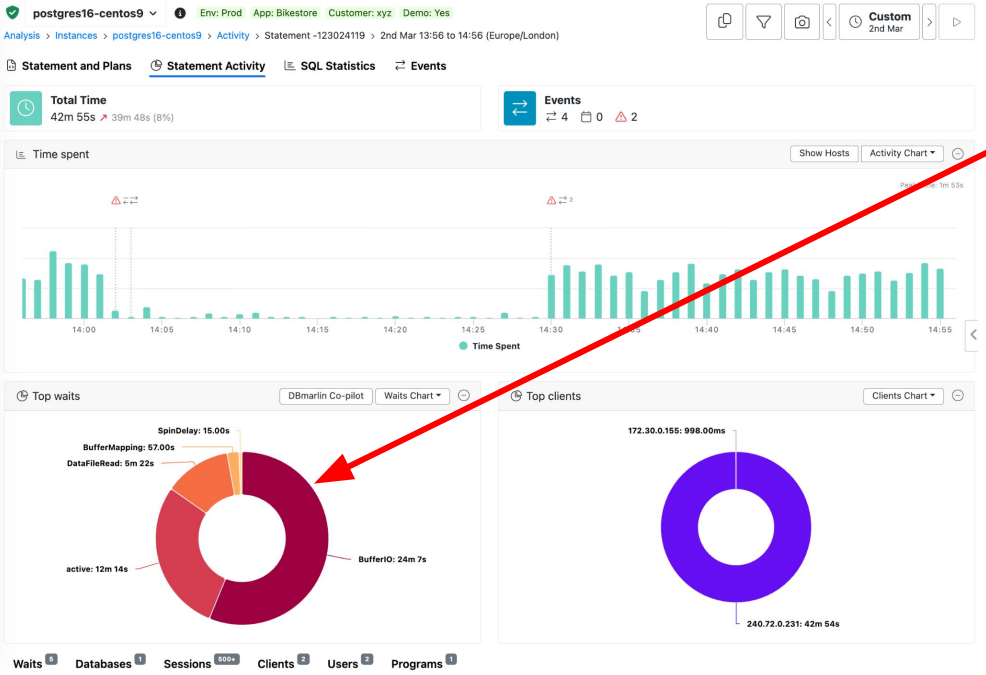
Click on View Statement (top right), and then “Statement Activity” (top middle)

See the Query execution plan for this statement

Copyright © 2026, Application Performance - Confidential

Analyse wait states

View wait states at statement level



See waitstates at statement level. Click on DBmarlin Co-pilot for more information.

Copyright © 2026, Application Performance - Confidential

4. Troubleshoot database locking

Problem

Blocked Queries are a significant problem for database performance. Inefficient query or database design and resource saturation can trigger blocking, which in turn can result in increased latency, and user dissatisfaction.

Identifying root blockers that can cause cascading locks on database resources is crucial to troubleshooting and addressing database performance issues.

Solution

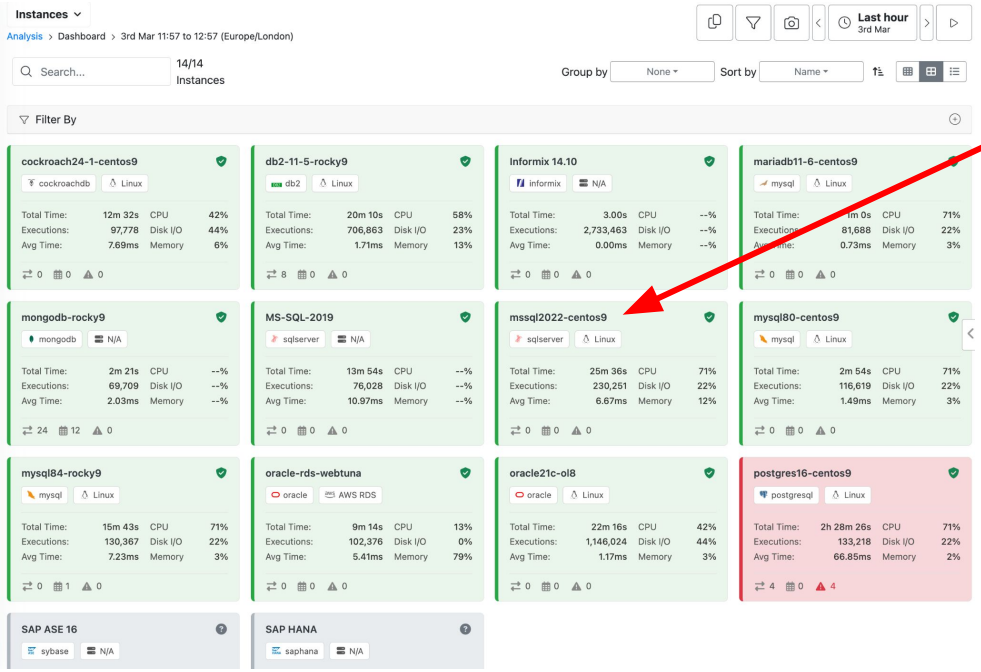
DBmarlin provides detailed visibility into root blocking queries, allowing you to efficiently survey and troubleshoot blocked connections and suboptimal performance across your entire database estate.

https://www.dbmarlin.com/blog/2023-05-31-troubleshoot-blocking-queries-with-dbmarlin

Copyright © 2026, Application Performance - Confidential

Troubleshoot database locking

Identify database to examine



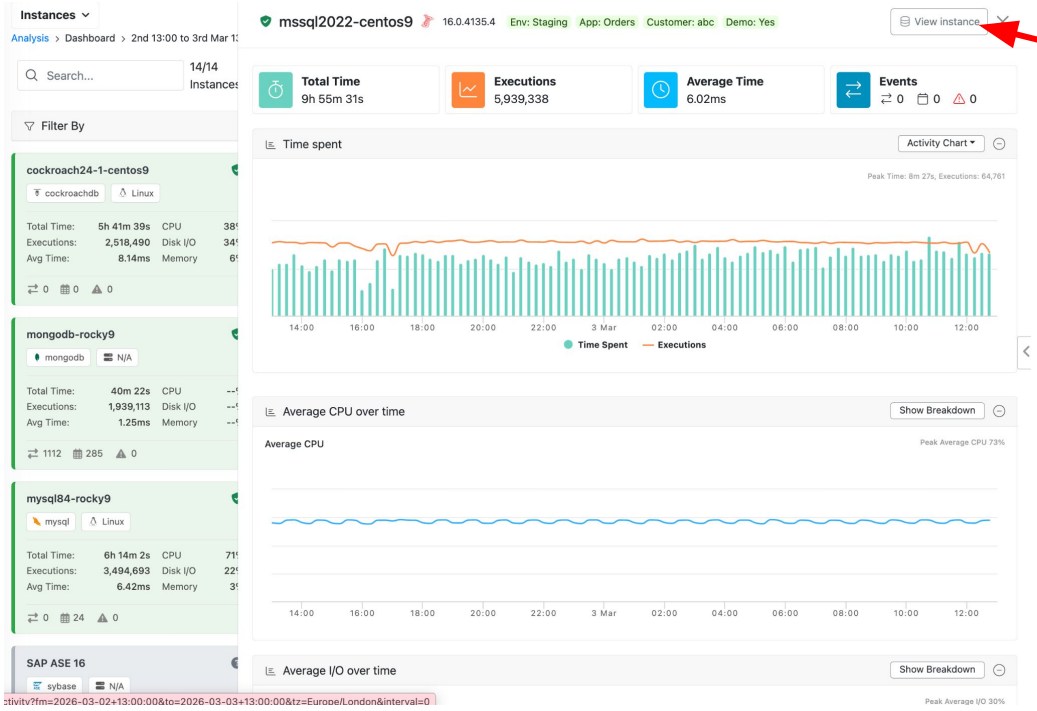
Select Database - in this case, MSSQL2022-centos contains some examples of locking. Click on the database name
And then click “View Instance”

Copyright © 2026, Application Performance - Confidential

DB marlin

Troubleshoot database locking

Using MS-SQL-2019



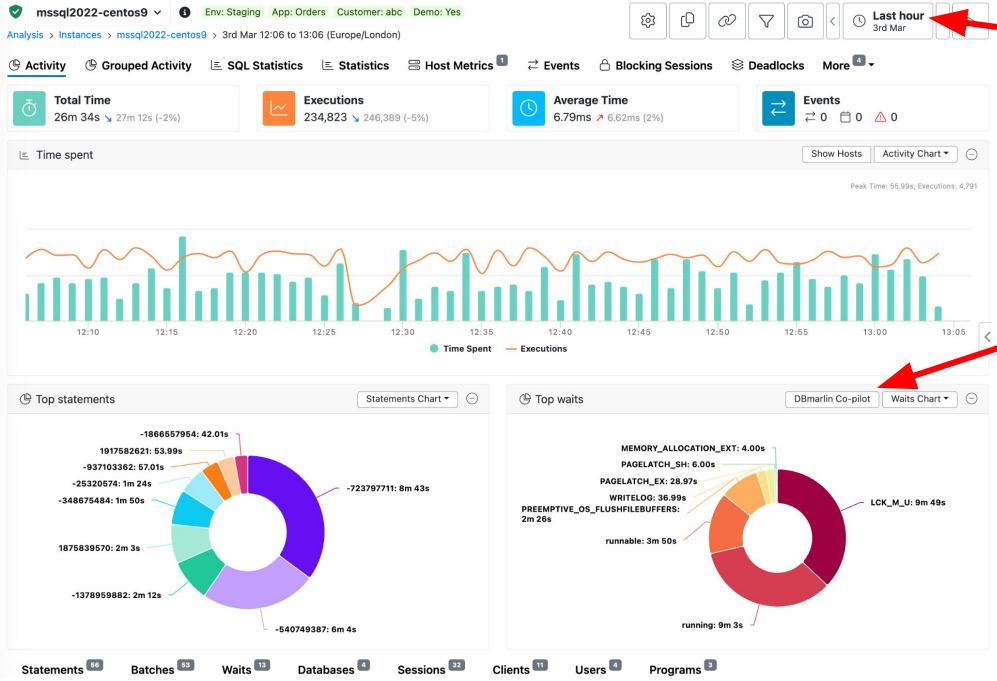
Select Database - in this case, MS-SQL-2019 contains some examples of locking. Click on the database name
And then click “View Instance”

Copyright © 2026, Application Performance - Confidential

DB marlin

Troubleshoot database locking

Using MS-SQL-2019



Set time selection to “Last Hour”

Launch DBmarlin Co-pilot

Copyright © 2026, Application Performance - Confidential

DB marlin

Summary of DBmarlin Co-pilot recommendations

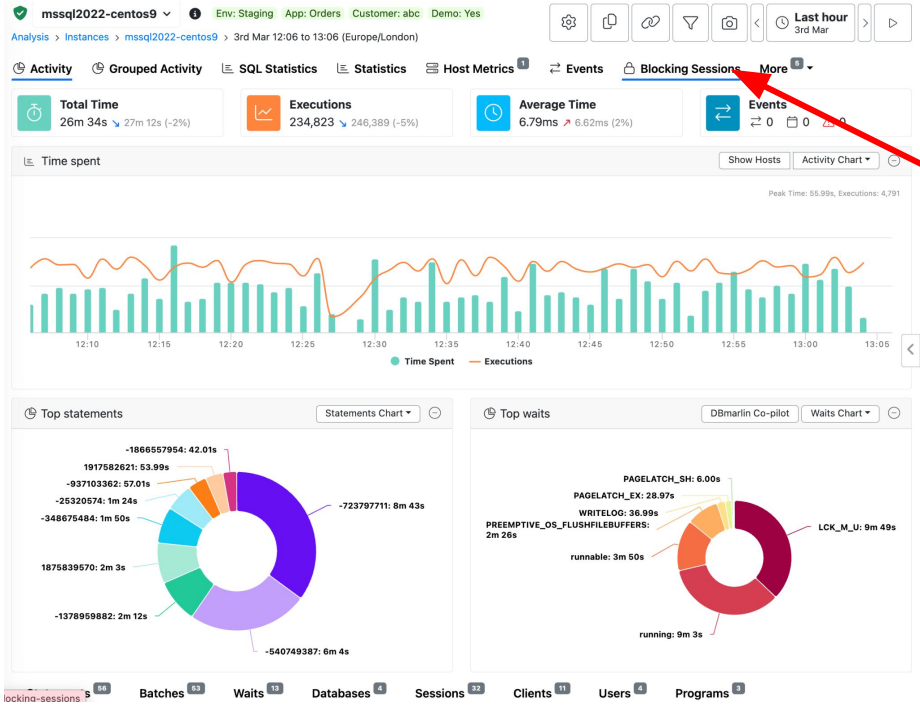
PDF is here: DBmarlin Co-Pilot for locking

1. Identify who is blocking whom
    a. Co-pilot recommends running a query to identify blockers
    This has to executed when the problem is apparent, which can be difficult, and by the time the query has been executed, it might miss the problem.
    DBmarlin can traverse locked sessions (and other statements that cause the lock anytime in the last 7 days)
2. Understand what is driving the sessions that are locking resources.
    a. Use DBmarlin to view statement and perhaps tune it.
    b. Check App Server / Web Server layers to see if lock logic is held there

Copyright © 2026, Application Performance - Confidential

Troubleshoot database locking

Examine Blocking sessions

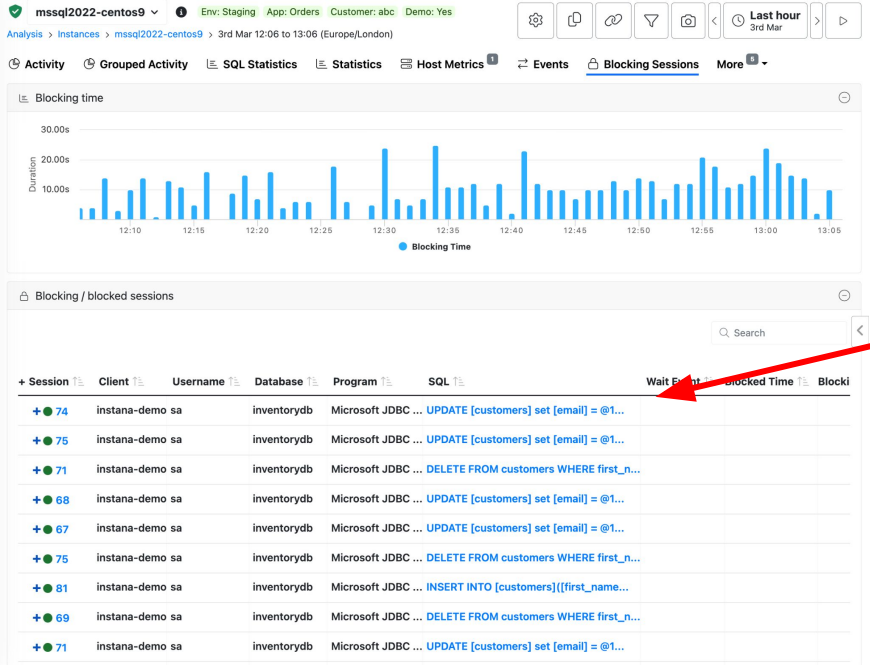


Select Blocking Sessions tab

Copyright © 2026, Application Performance - Confidential

Troubleshoot database locking

Examine Blocking sessions



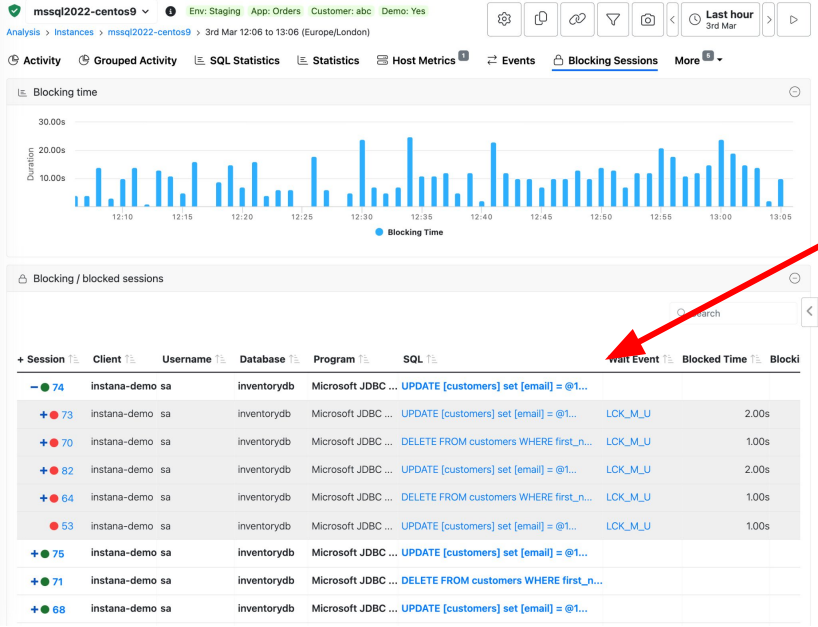
See list of recent blocked sessions

Copyright © 2026, Application Performance - Confidential

DB marlin

Troubleshoot database locking

Expand to see statements that are blocking



See list of recent blocked sessions

5 statements listed together with Blocked Time

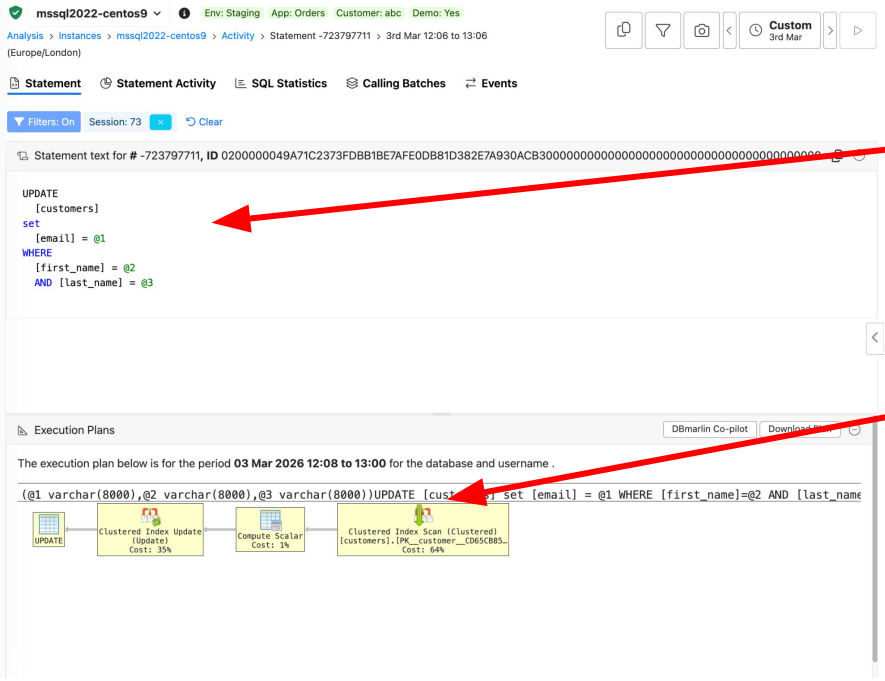
Click on Session 73 for more tuning advice

Copyright © 2026, Application Performance - Confidential

DB marlin

Troubleshoot database locking

Tuning one of the blocking sessions



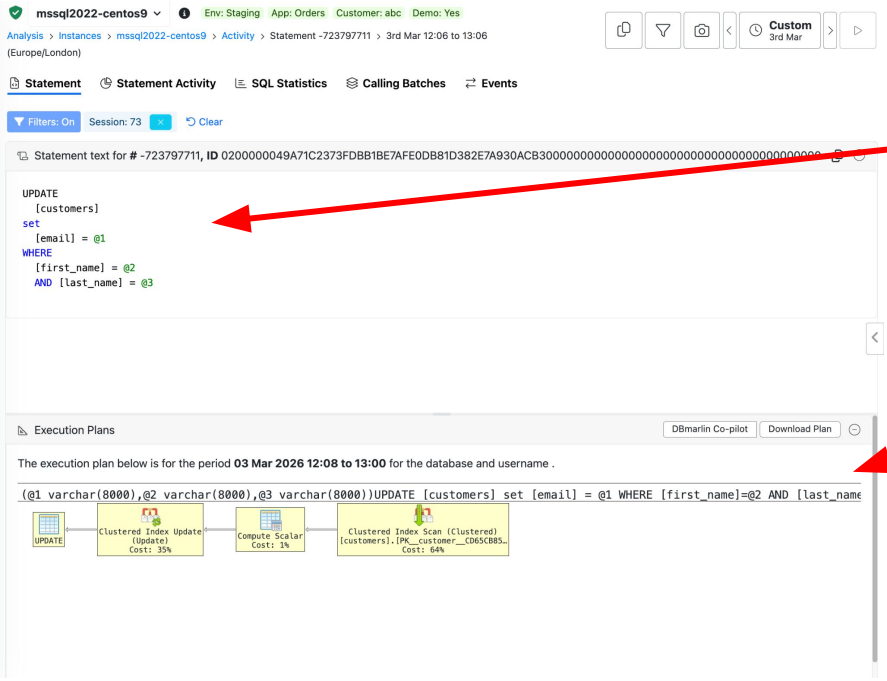
SQL Statement for one of the blockers

Execution Plan for the statement. Click on DBmarlin Co-pilot

Copyright © 2026, Application Performance - Confidential

Troubleshoot database locking

Tuning one of the blocking sessions



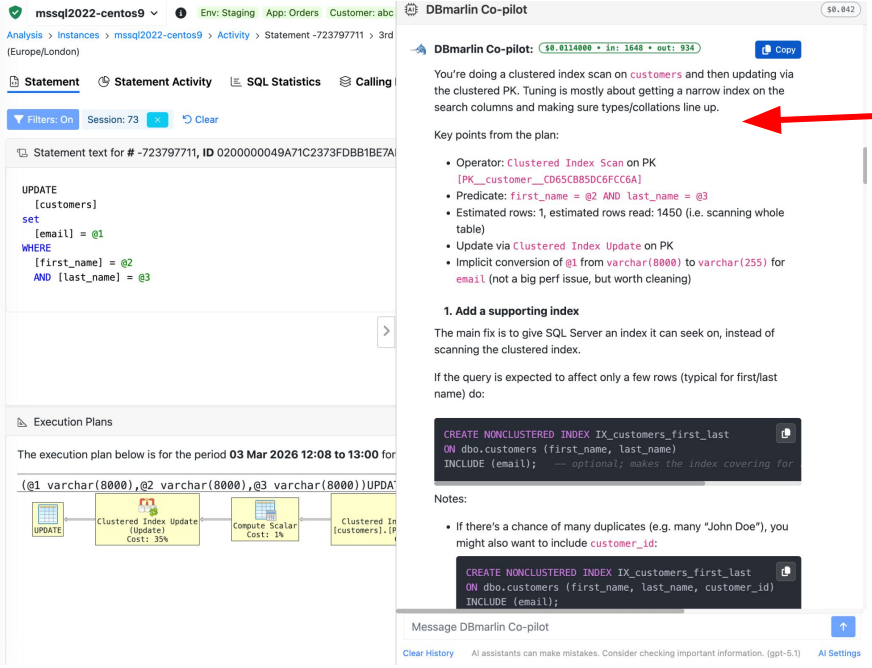
SQL Statement for one of the blockers

Execution Plan for the statement. Click on DBmarlin Co-pilot [Copilot findings for blocking sessions analysis]

Copyright © 2026, Application Performance - Confidential

Troubleshoot database locking

Tuning one of the blocking sessions



DBmarlin Co-pilot review. Attached here: Copilot findings for blocking sessions analysis

Copyright © 2026, Application Performance - Confidential

5. Configure DBmarlin co-pilot

- Problem

  A user of DBmarlin is trying to understand a performance issue. Maybe this is a new database platform. The user needs a deeper insight of what the data means,

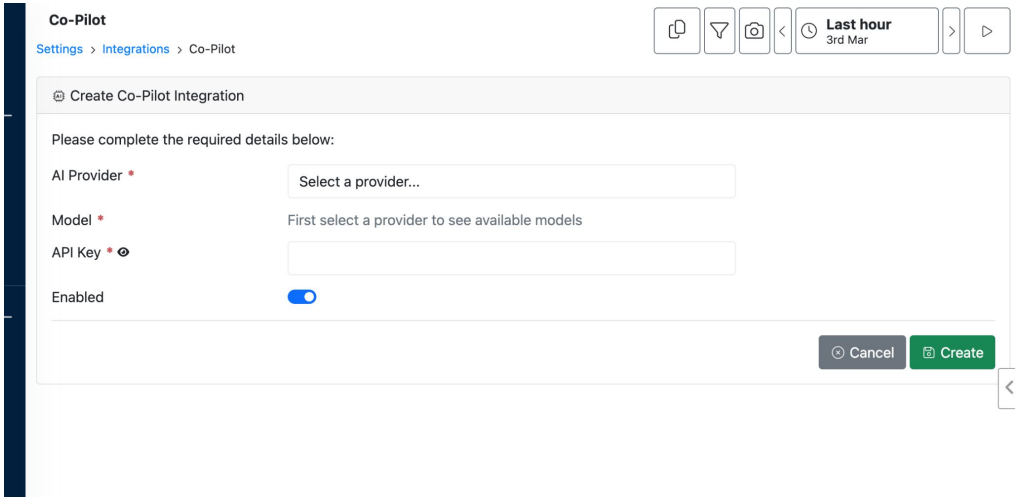
- Solution

  DBmarlin co-pilot can be configured to use a variety of available LLM’s at the user level or a system level. This example is set at the system level. The LLM will help to explain a database performance issue and remediation steps in a clear and concise way.

Copyright © 2026, Application Performance - Confidential

Configure DBmarlin co-pilot

Go to the “Integrations” screen

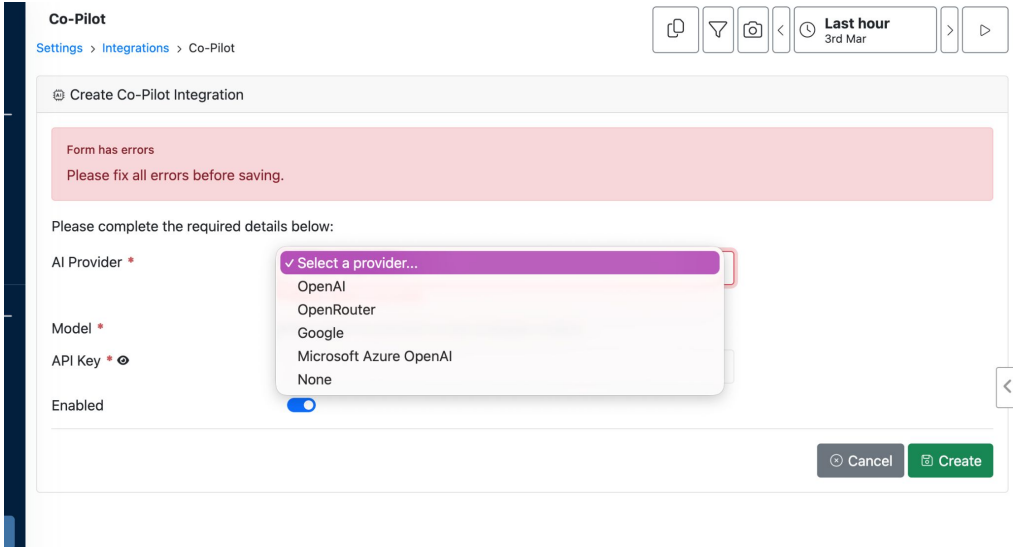


Under Settings, click on Integrations, Edit the “AI Co-pilot” section, and click Create.

Copyright © 2026, Application Performance - Confidential

Configure DBmarlin co-pilot
Select AI LLM provider

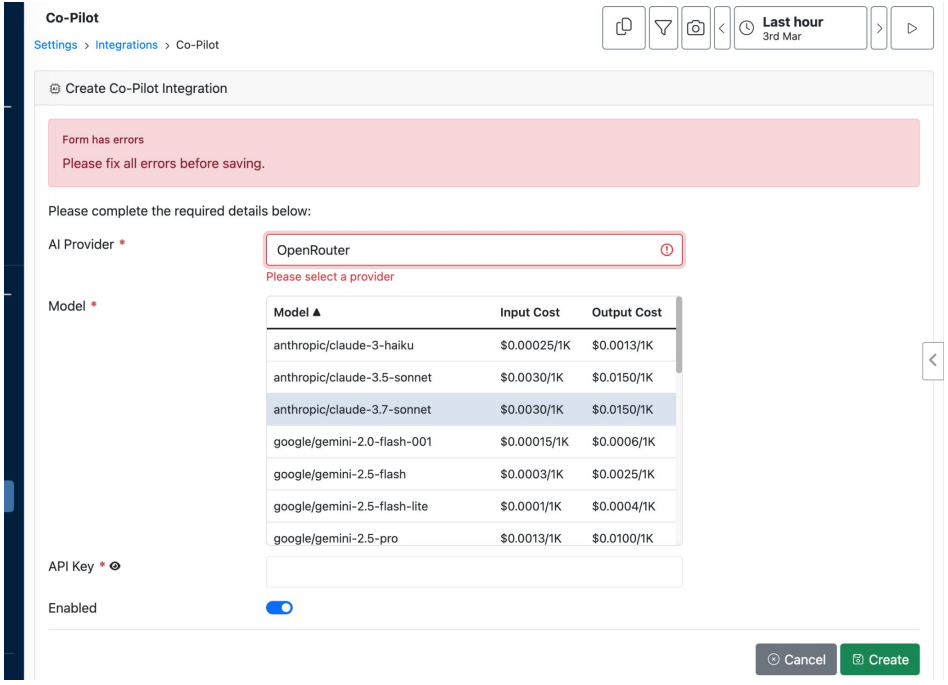
Select your provider



Copyright © 2026, Application Performance - Confidential

Configure DBmarlin co-pilot

Select AI Model



Select the model

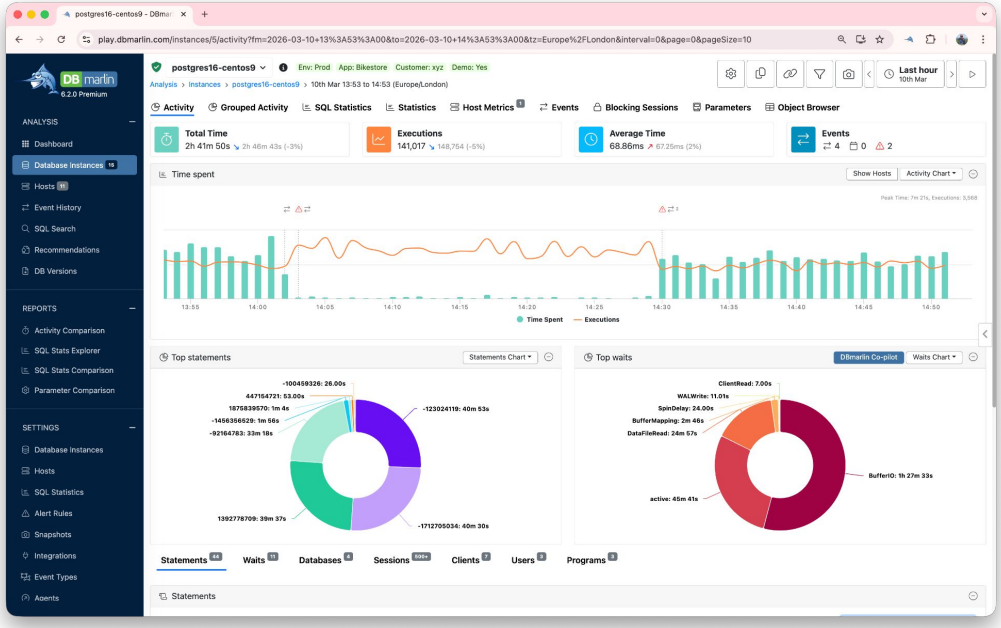
Enter the API Key for the provider

Click the Create button, and it is done.

Copyright © 2026, Application Performance - Confidential

Configure DBmarlin co-pilot

Test the configuration



For a database, open the instance, and click the “DBmarlin Co-Pilot” on the Top Waits pie chart

DBmarlin sends a question to your LLM, which provides the answer

Copyright © 2026, Application Performance - Confidential

6. Use DBmarlin co-pilot to tune SQL Query

Problem

DBmarlin has provided detail on the execution plan for a statement, but the plan is hard to interpret, and it is difficult to build a remediation plan from what is provided

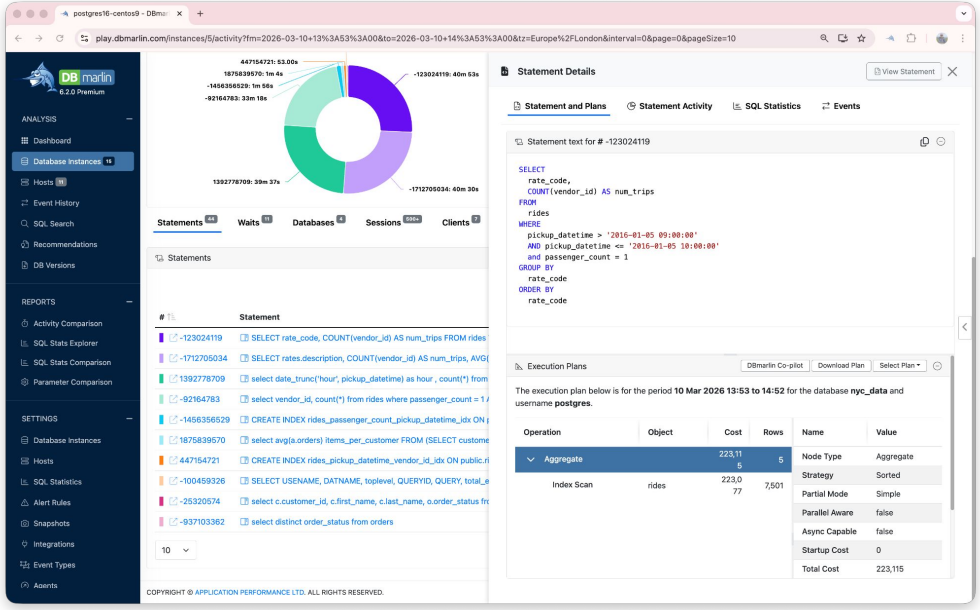
Solution

DBmarlin co-pilot will use the LLM of your choice to query how a SQL Statement is executed, and will make recommendations to improve performance.

Copyright © 2026, Application Performance - Confidential

Use DBmarlin co-pilot

Open up query.



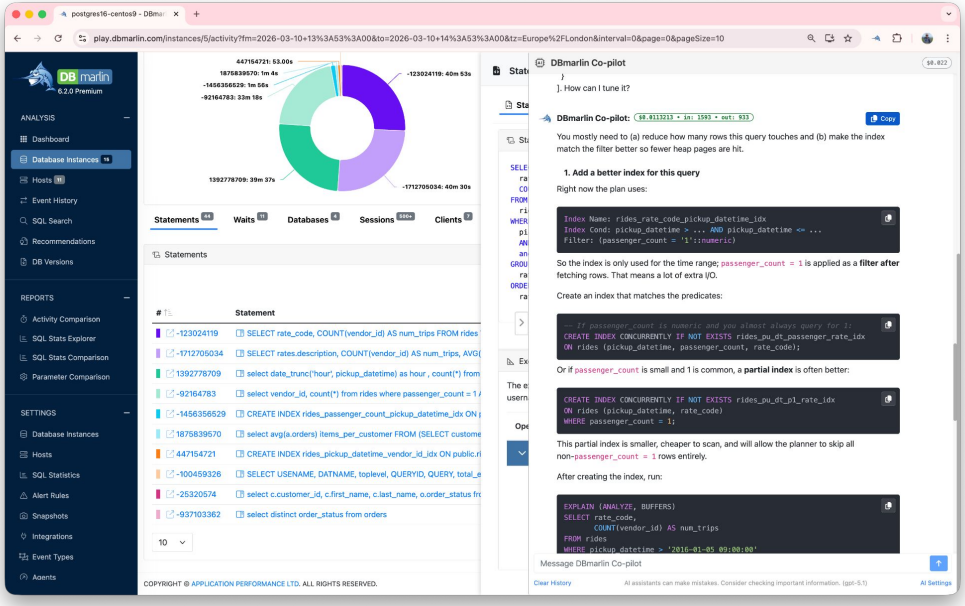
For a database, open the instance, scroll down to the statements. Click on one of the top statements

The Execution plan is provided. Click on DBmarlin Co-pilot to get a better understanding.

Copyright © 2026, Application Performance - Confidential

Use DBmarlin co-pilot

Review the findings



Co-pilot has produced a good set of recommendations. The main option is to use a better index.

It is possible to ask any follow up questions of DBmarlin Co-pilot

Copyright © 2026, Application Performance - Confidential

7. Use DBmarlin Co-pilot to understand wait states

Problem

DBmarlin has provided a chart highlighting the proportions of different wait states a database is experiencing. Optimising these to improve the performance is vitally important. Each database platform uses different terminology. The end user would like a detailed and concise definition of the wait states, and recommendations for remediation actions.

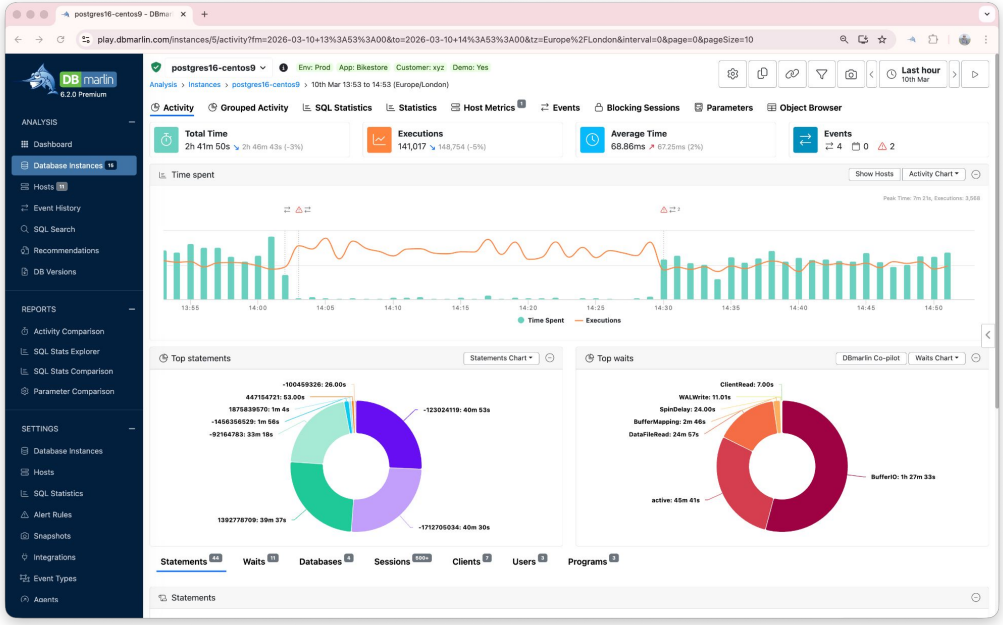
Solution

DBmarlin Co-pilot can provide a description of the wait state, and can provide recommendations for performance improvement.

Copyright © 2026, Application Performance - Confidential

Use DBmarlin co-pilot

Ask for Information



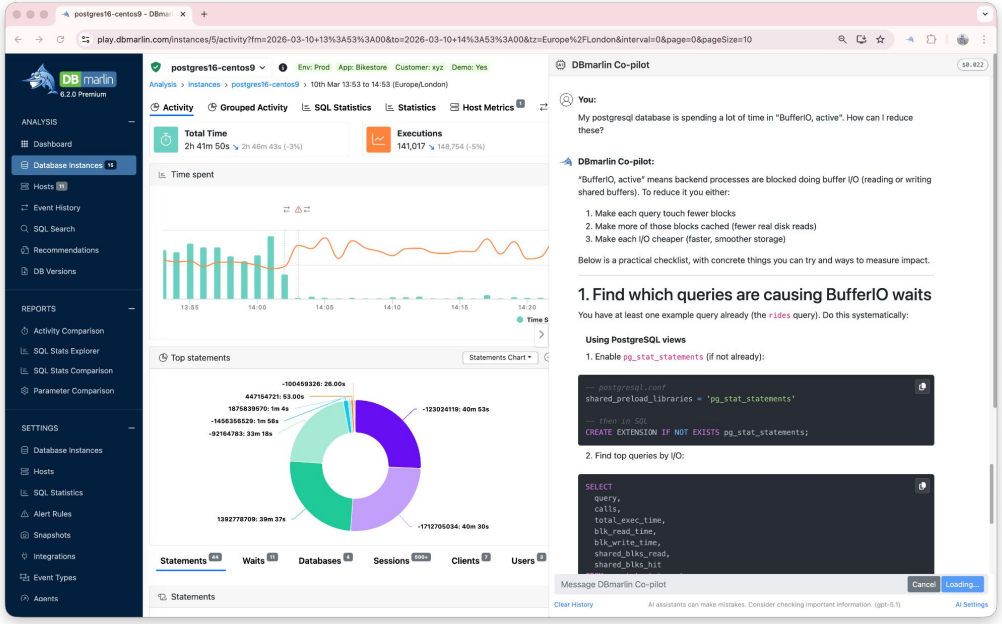
When viewing a database instance, there is a “Top Waits” chart.

Click on DBmarlin Co-pilot to get more information about top waits.

Copyright © 2026, Application Performance - Confidential

Use DBmarlin co-pilot

Interpret results



DBmarlin Co-pilot shows

- A definition of the top wait “BufferIO, active”
- A 5 point plan
  - Find which queries affected
  - Tune queries
  - Improve Cache Behaviour
  - Smooth and speed up I/O
  - Use DBmarlin to track impact

Copyright © 2026, Application Performance - Confidential

8. Use KB to understand wait states

Problem

DBmarlin has provided a chart highlighting the proportions of different wait states a database is experiencing. Optimising these to improve the performance is vitally important. Each database platform uses different terminology. The end user would like a detailed and concise definition of the wait states, and recommendations for remediation actions.

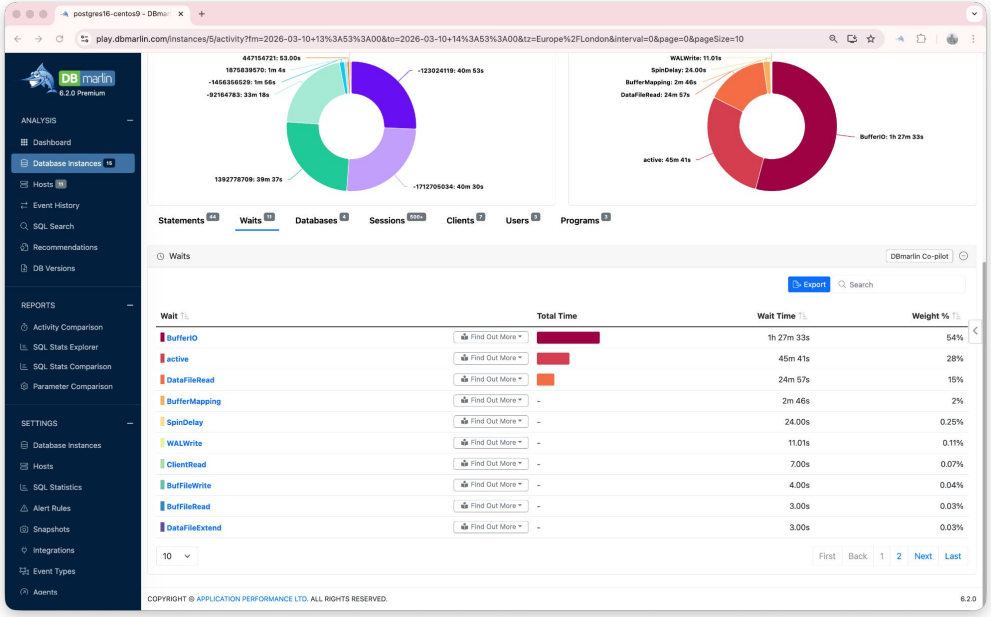
Solution

DBmarlin provides and maintains a knowledge base to help interpret the presented statistics for all of the databases that DBmarlin supports.

Copyright © 2026, Application Performance - Confidential

Use DBmarlin Knowledge Base

Ask for Information



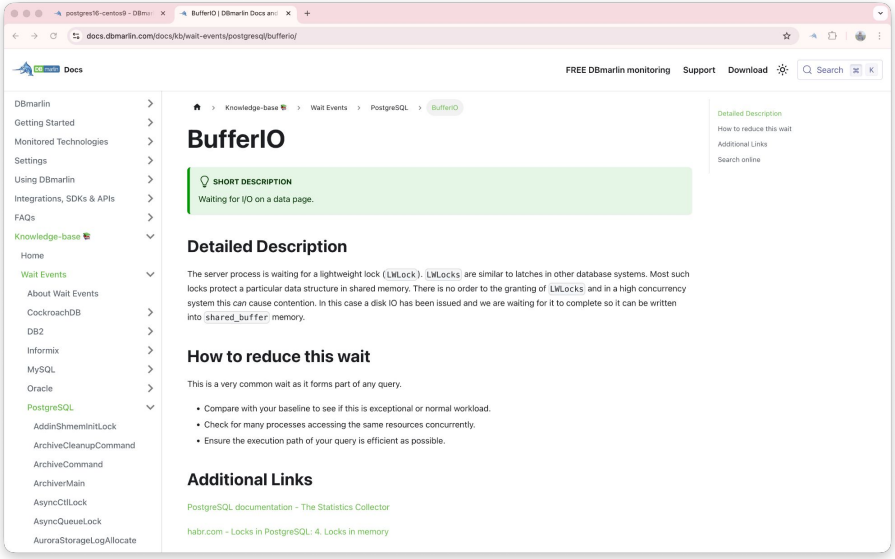
When viewing a database instance, scroll down to the list of Statements, click on Waits to view the top waits i a table.

Click on the “Find Out More” button and choose Knowledge Base”

Copyright © 2026, Application Performance - Confidential

Use DBmarlin Knowledge Base

Ask for Information



The DBmarlin knowledge base describes the wait state, presents options for reducing the wait, and additional links for more information.

Copyright © 2026, Application Performance - Confidential

9. Select a time period for analysis

Problem

A problem occurred in the past, and investigation can be difficult to explore and to reproduce.

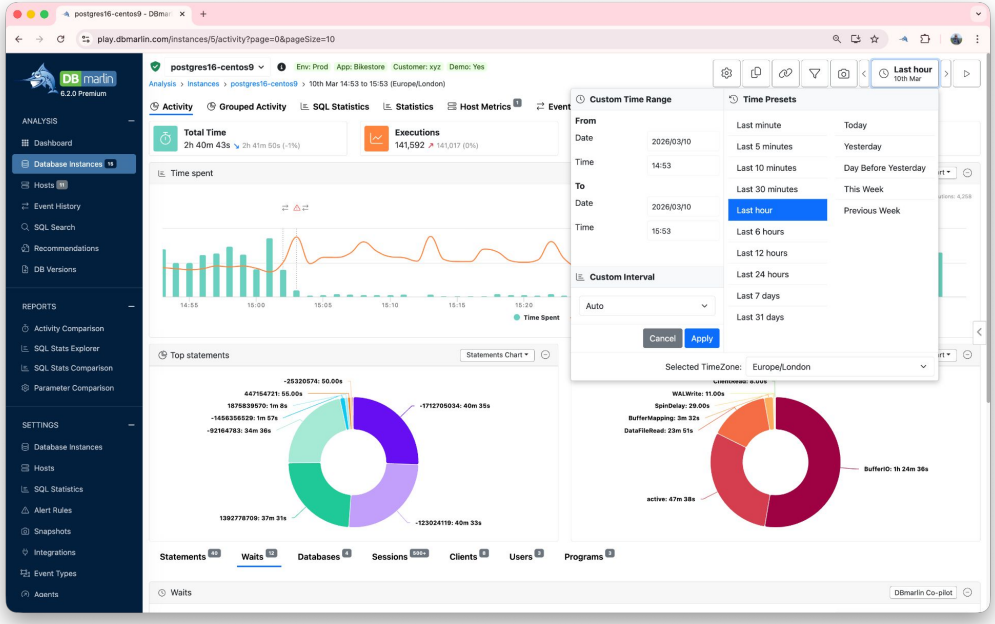
Solution

DBmarlin stores 1-second interval data for the past 7 days, and more summarised dta after that. DBmarlin offers a number of ways to select a previous period of time.

Copyright © 2026, Application Performance - Confidential

Select a time period for analysis

Choose a time period



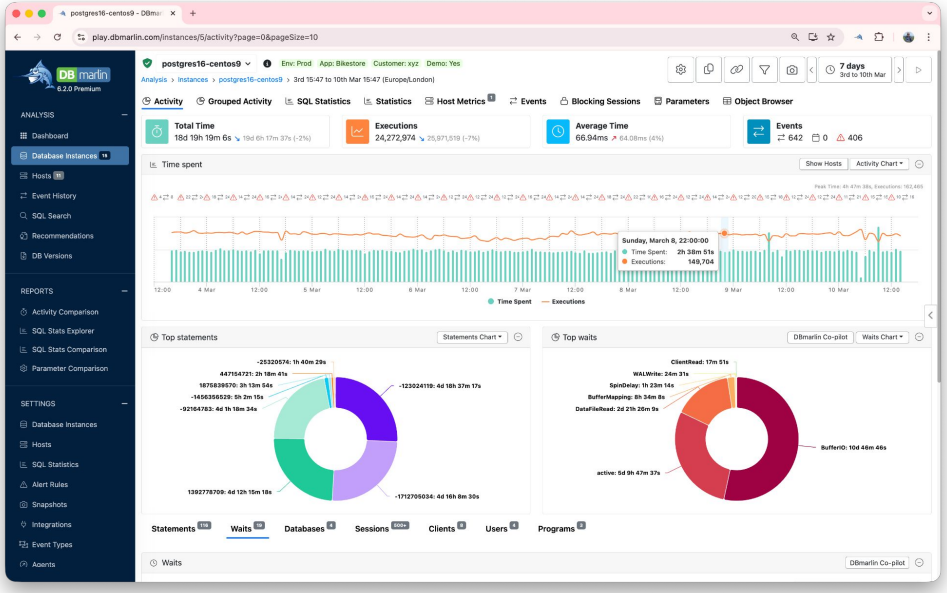
Click on time selector, and choose time period

- Presets such as last 5 minutes
- Custom Time Range or interval

Copyright © 2026, Application Performance - Confidential

Select a time period for analysis

Strategies for finding time.



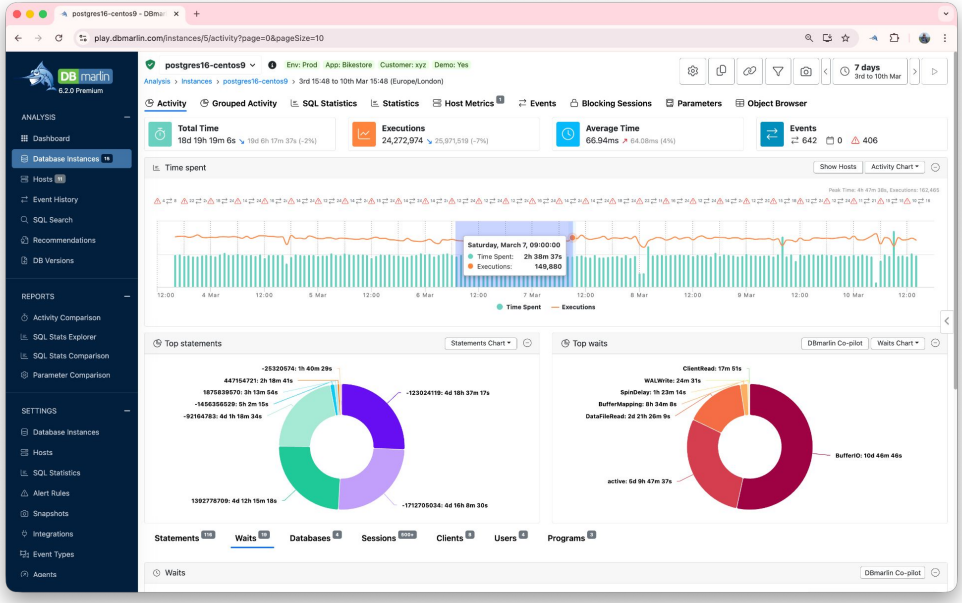
Expand time selection to past 7 days, Click and Drag inside timeline to find the exact time.

Copyright © 2026, Application Performance - Confidential

Select a time period for analysis

Strategies for finding time.

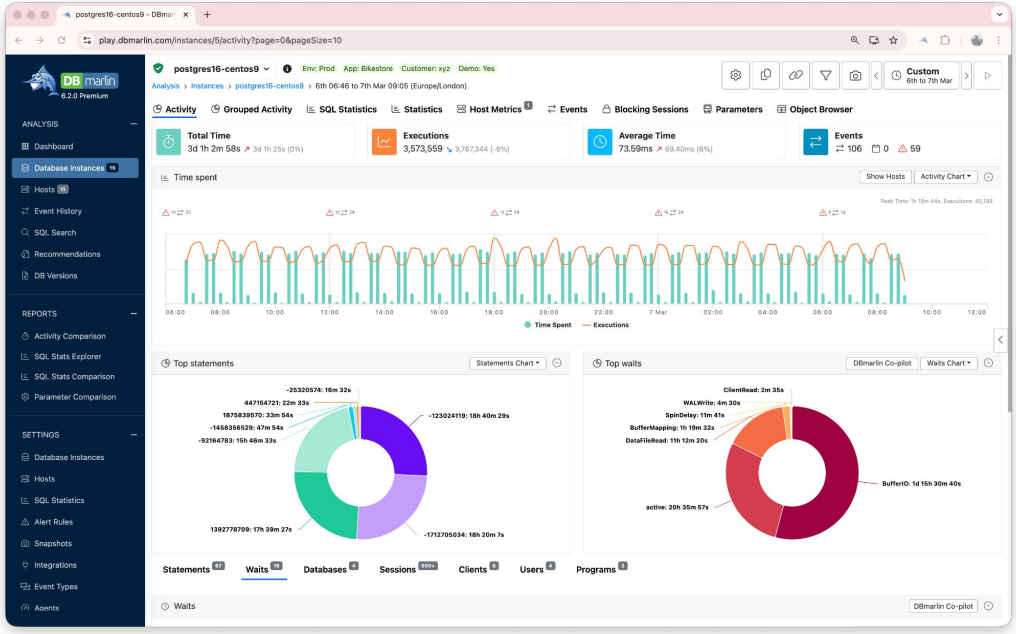
Time Period selected



Copyright © 2026, Application Performance - Confidential

Select a time period for analysis

Strategies for finding time.

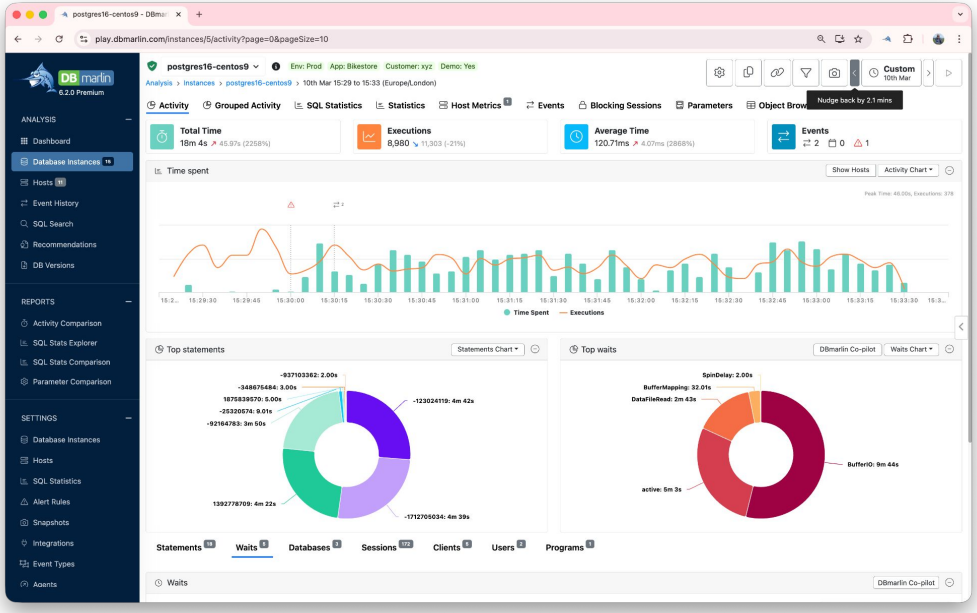


Zoomed in time period. This step can be performed until the time period is shown.

Copyright © 2026, Application Performance - Confidential

Select a time period for analysis

Nudge time back



If time period has interesting data a little bit earlier, click on the arrow to nudge it back.

Copyright © 2026, Application Performance - Confidential

10. Take a snapshot of database performance

Problem

Users need a way to capture a snapshot of database performance, so they can compare future behaviour against a known baseline. Without an easy mechanism to preserve key metrics, it is difficult to evaluate the impact of events such as major trading periods (e.g. Black Friday) or application releases.

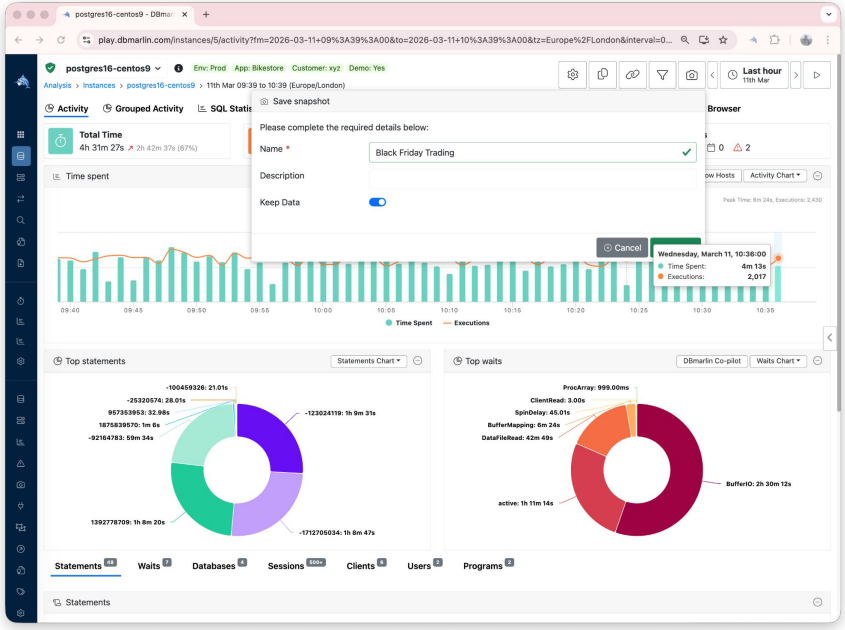
Solution

DBmarlin offers a mechanism to take a snapshot for any period of time, with the option to also capture the granular 1-second interval data for accurate evaluation at a later date.

Copyright © 2026, Application Performance - Confidential

Take a snapshot of database performance

Select time period, and click camera icon



Give the snapshot a descriptive name.

Optionally choose to keep data, so accurate (1 second) detail is available for later analysis

Copyright © 2026, Application Performance - Confidential

11. Compare database activity between snapshots or time periods

Problem

Users need a way to make an objective comparison of database performance now against a previous time window. This could help build readiness for major events (e.g. Black Friday), or could help to quantify performance improvements

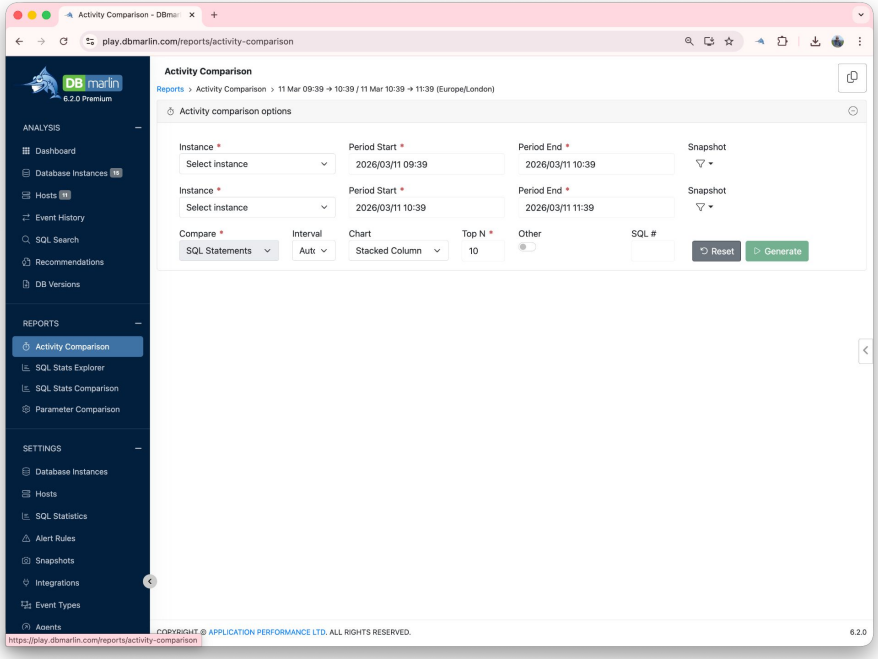
Solution

DBmarlin offers an Activity Comparison report that can compare performance between two different periods (either or both of these can be snapshots)

Copyright © 2026, Application Performance - Confidential

Compare database activity

Select the Activity Comparison report



Select the Instance, and then select time period, or select snapshot(s)

And click generate

Copyright © 2026, Application Performance - Confidential

Compare database activity

View activity comparison results

Activity Comparison Results

Metric	6th Sep 16:00 to 18:00 (Oracle RDS CPU bottleneck)	7th Sep 16:00 to 18:00 (Oracle RDS CPU bottleneck FIXED)	Difference
Instance Name	oracle-rds-webtuna	oracle-rds-webtuna	
Total DB Time	21h 5m 47s	22m	20h 43m 47s (-98%) ↘
Total Executions	330,427	243,911	86,516 (-26%) ↘
Average Execution Time	229.85ms	5.41ms	224.44ms (-98%) ↘
Host Average CPU	18%	14%	-4% ↘
Host Average Memory	82%	81%	-1% ↘
Detected Changes	1	0	1 ↘
Custom Events	0	0	0 =
Alerts	0	0	0 =
CPU Count	2	2	0 (0%) =

This comparison is for before / after a fix for a CPU bottleneck.

Average Execution time was down 98%

Copyright © 2026, Application Performance - Confidential

DB marlin

Compare database activity

View activity comparison results

Activity Comparison Results

Metric	6th Sep 16:00 to 18:00 (Oracle RDS CPU bottleneck)	7th Sep 16:00 to 18:00 (Oracle RDS CPU bottleneck FIXED)	Difference
Instance Name	oracle-rds-webtuna	oracle-rds-webtuna	
Total DB Time	21h 5m 47s	22m	20h 43m 47s (-98%) ↘
Total Executions	330,427	243,911	86,516 (-26%) ↘
Average Execution Time	229.85ms	5.41ms	224.44ms (-98%) ↘
Host Average CPU	18%	14%	-4% ↘
Host Average Memory	82%	81%	-1% ↘
Detected Changes	1	0	1 ↘
Custom Events	0	0	0 =
Alerts	0	0	0 =
CPU Count	2	2	0 (0%) =

This comparison is for before / after a fix for a CPU bottleneck.

Average Execution time was down 98%

Copyright © 2026, Application Performance - Confidential

DB marlin

Compare database activity

View activity comparison results



This comparison is for before / after a fix for a CPU bottleneck.

Top waits over time and Top statements over time reduced to negligible amounts

Copyright © 2026, Application Performance - Confidential

12. Explore SQL Statistics

Problem

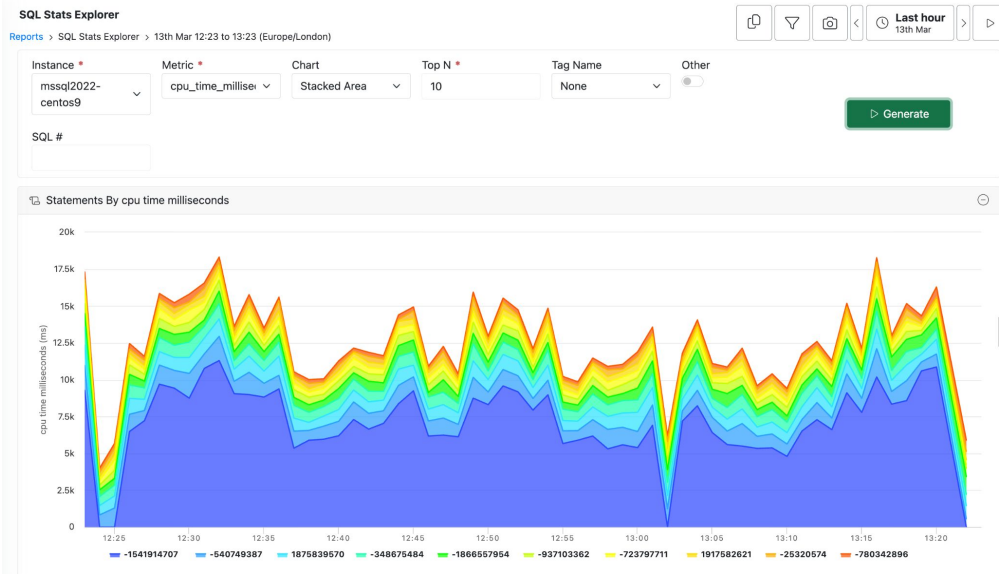
SQL Statistics are essential performance metrics that help in analysing and optimising the performance of SQL queries and database systems. These statistics provide insights into how the database engine processes SQL statements, enabling database administrators and developers to identify performance bottlenecks and optimize query execution.

Solution

DBmarlin presents relevant SQL Statistics in the dashboards, and it is also possible to create custom reports using SQL Stats Explorer, and to make comparisons between instances or between different times.

Copyright © 2026, Application Performance - Confidential

Explore SQL Statistics
Using SQL Stats Explorer



In Reports->SQL Stats Explorer, the user has selected cpu time (ms) for an MS SQL Server db, using a stacked area chart.

The largest area in the chart corresponds to the SQL Statement with the most execution time.

Copyright © 2026, Application Performance - Confidential

Compare database activity

View activity comparison results



From the menu, select Reports->SQL Stats Explorer

The user can choose their instance, Metric (in this case SQL Statement duration) as a stacked column, grouped by statement

Copyright © 2026, Application Performance - Confidential

DB marlin

13. Configure SQL Statistics

Problem

Each database platform can report on a wide variety of statistics. What can be done if a user can't see a statistic that they want?

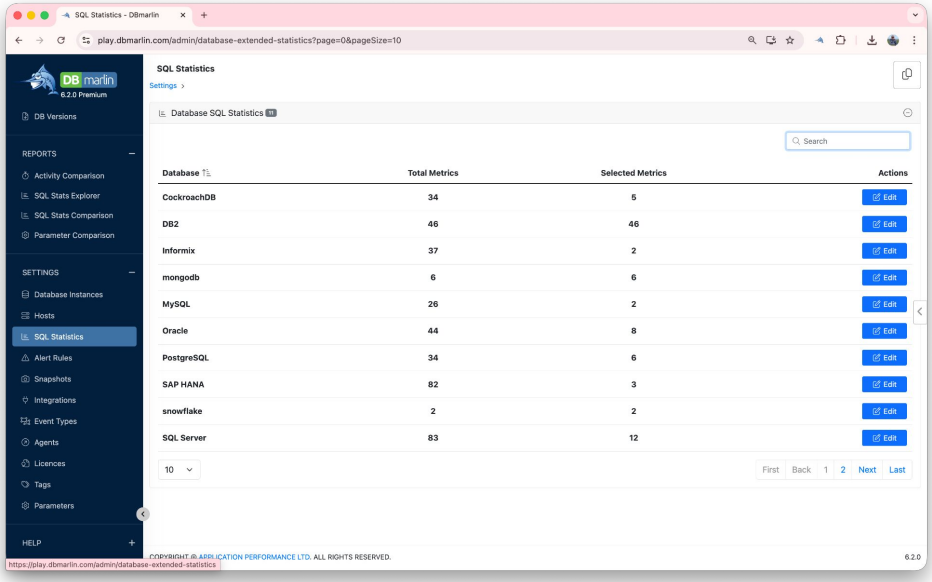
Solution

In DBmarlin's Settings->SQL Statistics page, users can configure which database statistics that they collect. New statistics can be added for future collection.

Copyright © 2026, Application Performance - Confidential

Configure SQL Statistics

Select database platform settings

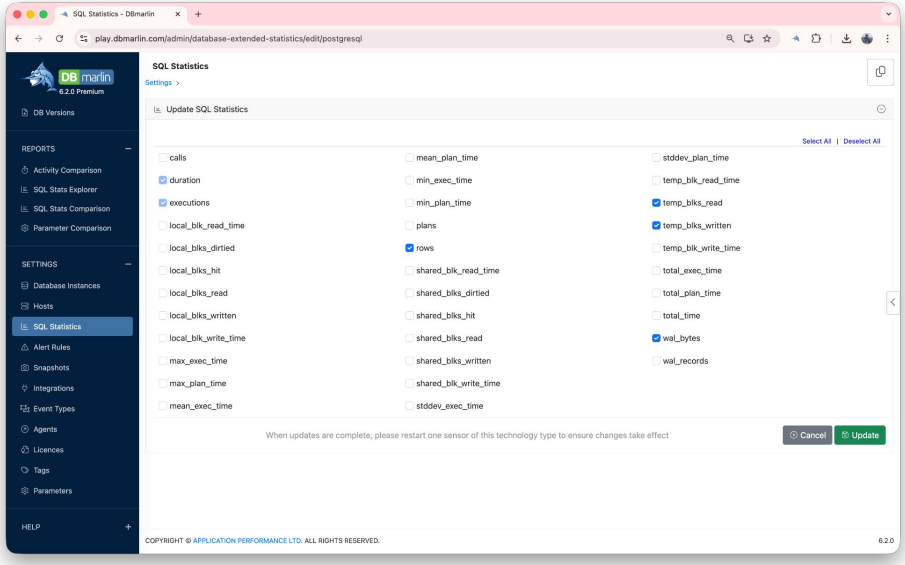


Select Settings->SQL Statistics to bring up the list of configured statistics. For the platform of interest, click on Edit.

Copyright © 2026, Application Performance - Confidential

Configure SQL Statistics

Make alterations to list of statistics gathered



Click to select or deselect statistics gathered. Each statistic has a CPU cost for gathering the information, so be careful making selections.

Copyright © 2026, Application Performance - Confidential

14. Compare SQL Statistics over time or between instances

Problem

Users need a way to make an objective comparison of database performance now against a previous time window. This could help build readiness for major events (e.g. Black Friday), or could help to quantify performance improvements

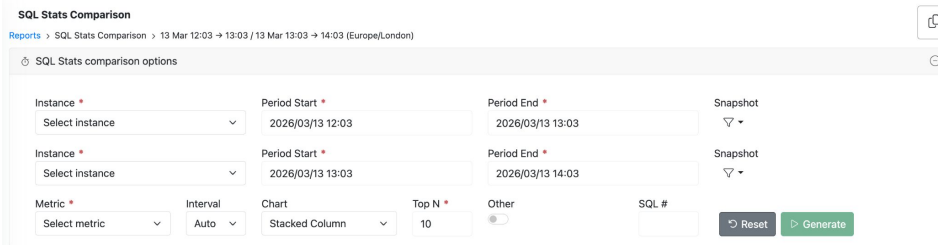
Solution

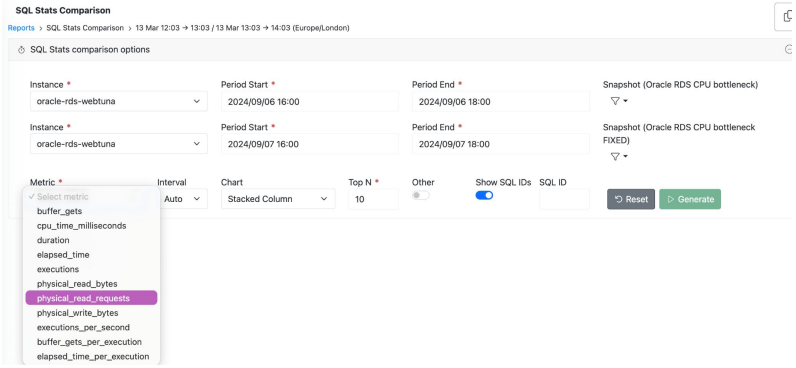
DBmarlin provides a SQL Stats Comparison report in order to explore and understand a comparison of recent performance and a well established benchmark time. This could also be used to contribute to a report showing before and after statistics.

Copyright © 2026, Application Performance - Confidential

Compare SQL Statistics

Select time periods and the metric to view





Click on Reports->SQL Stats Comparison

User has chosen a before / after tuning snapshot for an Oracle database, and they have chosen physical read requests as the metrics to display

Copyright © 2026, Application Performance - Confidential

Configure SQL Statistics

View Results

This is the first pane of the comparison. The selected statistic (executions per second) is showing as reduced from 5,448 to 4,063

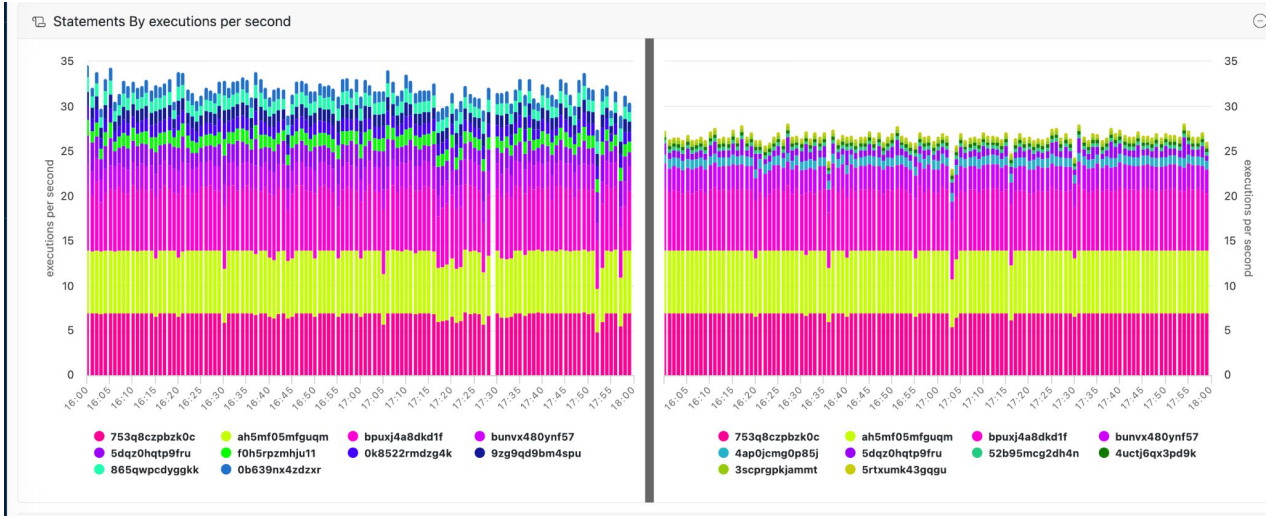
Metric	6th Sep 16:00 to 18:00	7th Sep 16:00 to 18:00	Period Difference
Instance Name	oracle-rds-webtuna	oracle-rds-webtuna	
Total Duration	20h 6m 54s	31m 14s	19h 35m 39s ↘
Total Executions	326,782	243,765	83,017 ↘
Average Execution Time	221.60ms	7.69ms	213.91ms ↘
CPU Count	2	2	0 =
executions_per_second	5,446	4,063	1,384 ↘

Copyright © 2026, Application Performance - Confidential

Configure SQL Statistics

View Results grouped by statements

This is the second pane of the comparison. The selected statistic (executions per second) is showing graphically before (on the left) and after (on the right)



Copyright © 2026, Application Performance - Confidential

15. Database recommendations

Problem

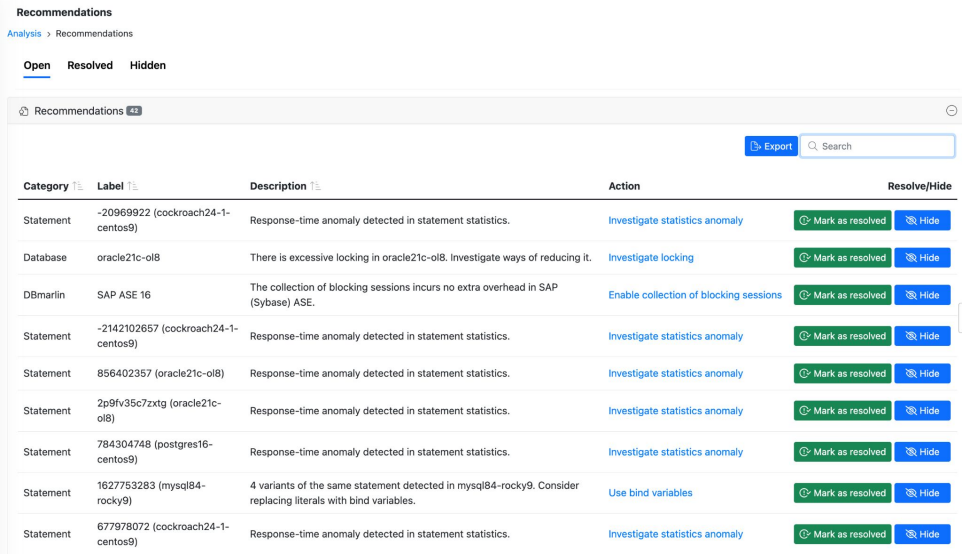
Users need a way to achieve a good configuration of their database platform, but with multiple platforms in use, and with less specialist knowledge for all them it can be very difficult to achieve this.

Solution

DBmarlin checks the database configurations, and will highlight recommendations for changes to improve configuration for security, operational health and performance

Copyright © 2026, Application Performance - Confidential

Recommendations
View all recommendations



This option is available from Analysis -> Recommendations. DBmarlin groups by Open, Resolved and Hidden.

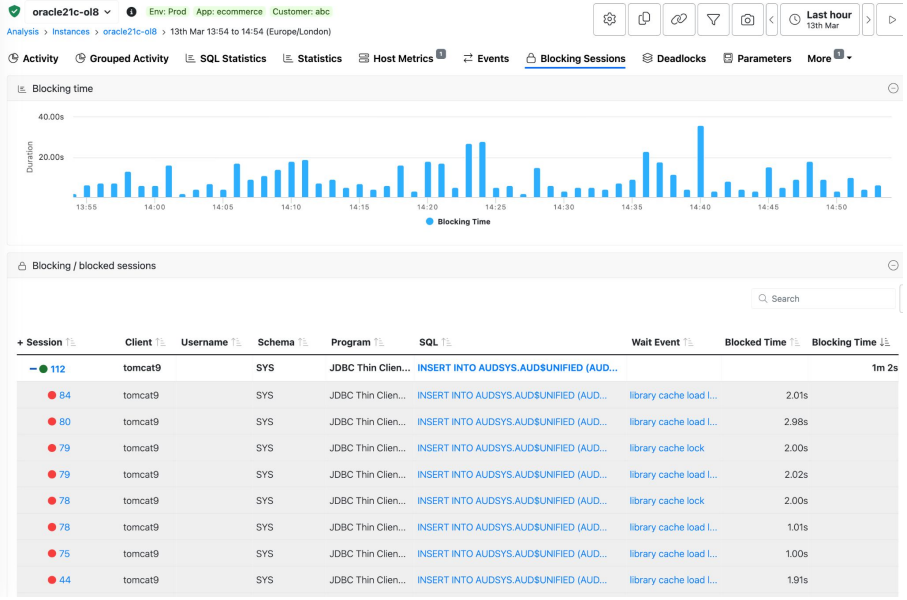
This analysis is an important step towards database configuration health for many different platforms.

The recommendation items can be managed, supporting any operational processes in place.

Copyright © 2026, Application Performance - Confidential

Recommendations

Dive into one recommendation



The deep dive shows recent locking activity for this database. The locked statements are displayed along with statements that are locking.

Previously, it has been very difficult to keep track of locking. DBmarlin provides this information, so that application developers can reduce the risk of locking.

Copyright © 2026, Application Performance - Confidential

16. Explore Database versions

Problem

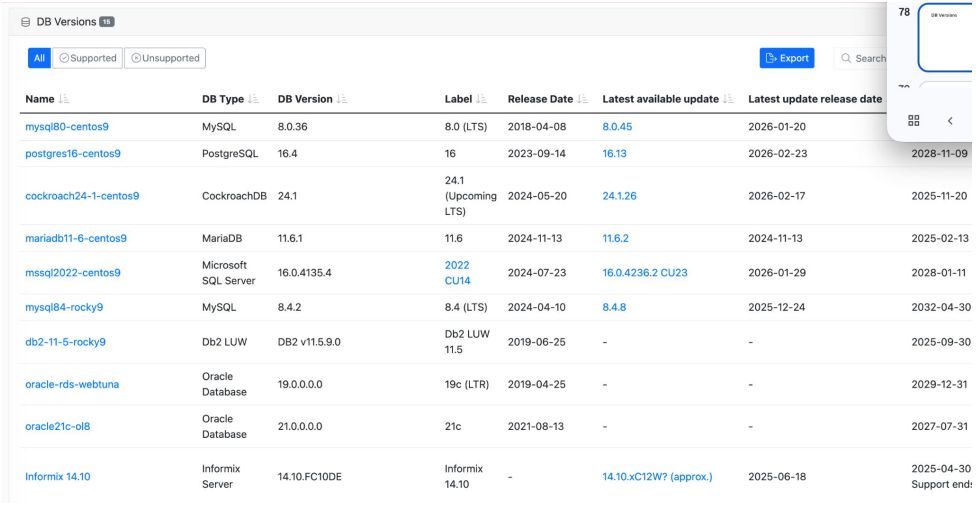
Old versions of software for database platform is a risk for any company. Not updating database software creates severe, high-level risks that go beyond simple technical inconveniences.

Solution

DBmarlin provides an overview of all configured databases with the release date, latest available update, end of mainstream support and end of extended support.

Copyright © 2026, Application Performance - Confidential

DB Versions



Name	DB Type	DB Version	Label	Release Date	Latest available update	Latest update release date	
mysql80-centos9	MySQL	8.0.36	8.0 (LTS)	2018-04-08	8.0.45	2026-01-20	
postgres16-centos9	PostgreSQL	16.4	16	2023-09-14	16.13	2026-02-23	2028-11-09
cockroach24-1-centos9	CockroachDB	24.1	24.1 (Upcoming LTS)	2024-05-20	24.1.26	2026-02-17	2025-11-20
mariadb11-6-centos9	MariaDB	11.6.1	11.6	2024-11-13	11.6.2	2024-11-13	2025-02-13
mssql2022-centos9	Microsoft SQL Server	16.0.4135.4	2022 CU14	2024-07-23	16.0.4236.2 CU23	2026-01-29	2028-01-11
mysql84-rocky9	MySQL	8.4.2	8.4 (LTS)	2024-04-10	8.4.8	2025-12-24	2032-04-30
db2-11-5-rocky9	Db2 LUW	DB2 v11.5.9.0	Db2 LUW 11.5	2019-06-25	-	-	2025-09-30
oracle-rds-webtuna	Oracle Database	19.0.0.0.0	19c (LTR)	2019-04-25	-	-	2029-12-31
oracle21c-ol8	Oracle Database	21.0.0.0.0	21c	2021-08-13	-	-	2027-07-31
Informix 14.10	Informix Server	14.10.FC10DE	Informix 14.10	-	14.10.xC12W? (approx.)	2025-06-18	2025-04-30 Support ends

This view shows all the databases under management, together with information about available updates, and support information for the databases.

This view can be used to ensure that all databases are supported and patched correctly, reducing risk of outage, data leakage or corruption.

Copyright © 2026, Application Performance - Confidential

17. Configure https access to DBmarlin

Problem

Sending data over the network without certificates and encryption leaves an organisation vulnerable to “man in the middle” and/or sensitive data collection over the wire.

Solution

DBmarlin allows easy configuration of SSL using the DBmarlin NGINX front end.

Copyright © 2026, Application Performance - Confidential

Steps to configure https

1. Obtain SSL Certificate
   a. Generate certificate and use the certificate files and private key file
2. Place the certificate files
   a. Copy the certificate into `/opt/dbmarlin/nginx/conf`
3. Configure NGINX for HTTPS
   a. Edit `/opt/dbmarlin/nginx/conf/nginx.conf`
4. Restart DBmarlin
5. Update agents

Reference documentation

https://docs.dbmarlin.com/docs/Getting-Started/ssl

Copyright © 2026, Application Performance - Confidential

18. Update DBmarlin

Problem

New releases of DBmarlin add new features, and new supported platforms as well as fixing vulnerabilities. How can I maintain an ideal cadence with minimal downtime and disruption

Solution

DBmarlin provides a clear process for upgrades, recent changes have reduced the time required for the update to the database. This reduces downtime opening up the possibility that an upgrade can be performed during office hours.

Documentation Linux or Windows

Copyright © 2026, Application Performance - Confidential

19. Collect logs

Problem

A user has encountered a problem, and the support organisation has asked for logs. How can I collect them?

Solution

From DBmarlin 6.10, admin users can generate a support bundle from the browser using the Self Monitoring screen. On older versions, run the support files script from the command line on Linux or Windows.

Copyright © 2026, Application Performance - Confidential

Linux process to capture logs

Log in, and set user to dbmarlin

```
user@dbmarlin:~$ sudo -i -u dbmarlin
[sudo] password for user:
dbmarlin@dbmarlin:~$
```

Run the command to capture the support files (this will take a few minutes)

```
dbmarlin@dbmarlin:~$ cd /opt/dbmarlin
dbmarlin@dbmarlin:~$ ./support_files.sh
```

The support files capture the necessary log files, and package them into a compressed archive. DBmarlin makes this available as a URL, so it can be easily downloaded

Since DBmarlin 6.10, you can also download the support files from the browser using the Self Monitoring screen.

Copyright © 2026, Application Performance - Confidential

Windows process to capture logs

Using a command prompt, cd to the right directory.

```
Microsoft Windows [Version 10.0.17763.8276]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd "\Program Files\DBmarlin"
```

Run the command to capture the support files (This will take a few minutes)

```
C:\Program Files\DBmarlin>support_files.bat
```

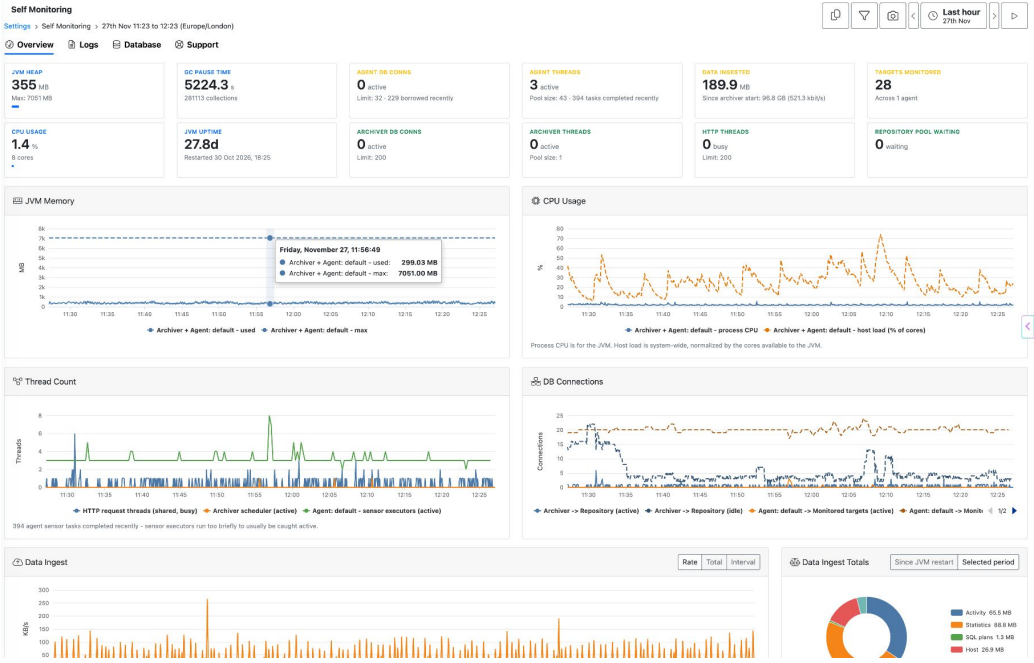
The support files capture the necessary log files, and package them into a compressed archive. DBmarlin makes this available as a URL, so it can be easily downloaded

Since DBmarlin 6.10, you can also download the support files from the browser using the Self Monitoring screen.

Copyright © 2026, Application Performance - Confidential

Collect logs

Use the Self Monitoring screen (6.10+)



Go to Help -> Self Monitoring and open the Support tab.

Click Download Support Files, wait for the status to complete, then click Download Archive.

The Logs tab lets you view and search Tomcat, PostgreSQL and nginx logs live while you reproduce a problem.

Copyright © 2026, Application Performance - Confidential

DB marlin

20. Use tags to classify database instances

Problem

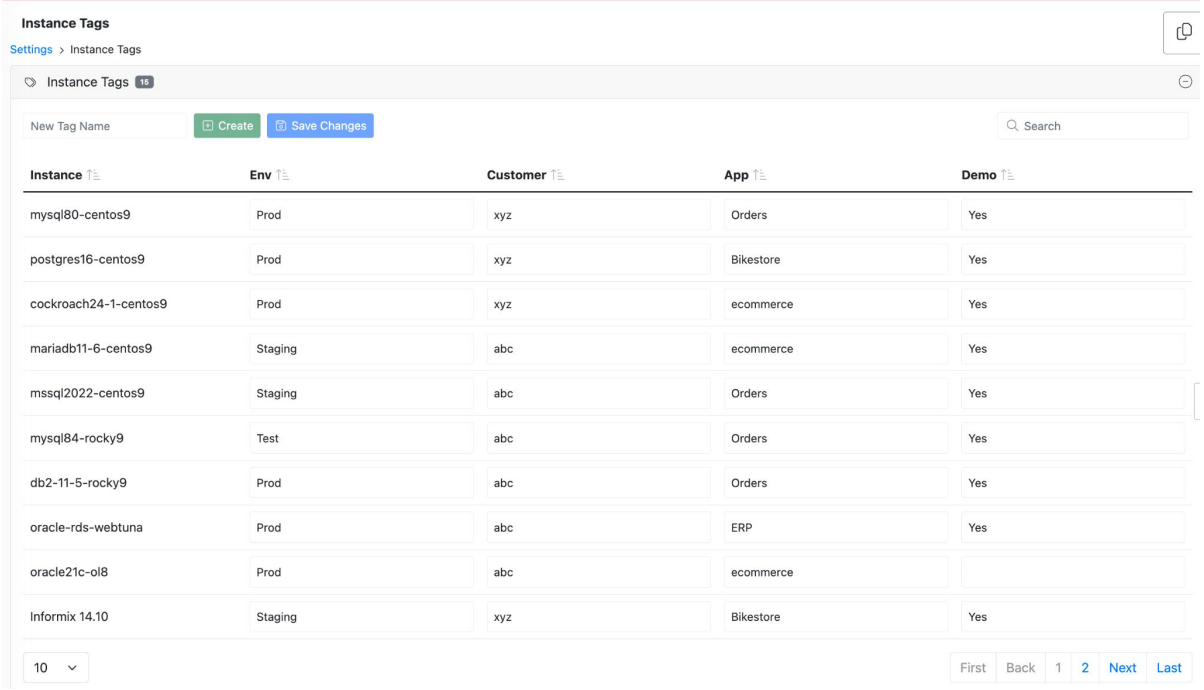
I have many different instances under management, it can be difficult to identify the right database depending on the environment (QA, DEV or Prod) or the application name (ABC, XYZ)

Solution

DBmarlin provides the ability to associate a name/value pair to a database instance, together with options to report on it, or group by it.

Copyright © 2026, Application Performance - Confidential

Configuring Tags

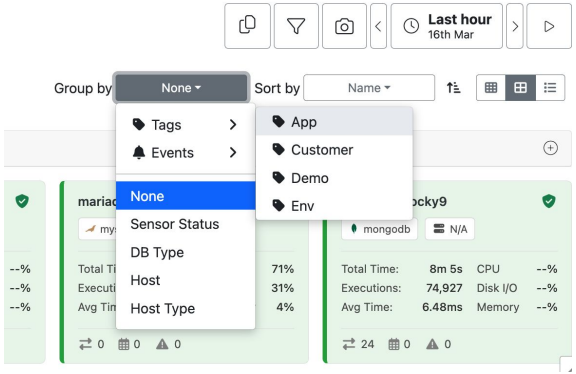


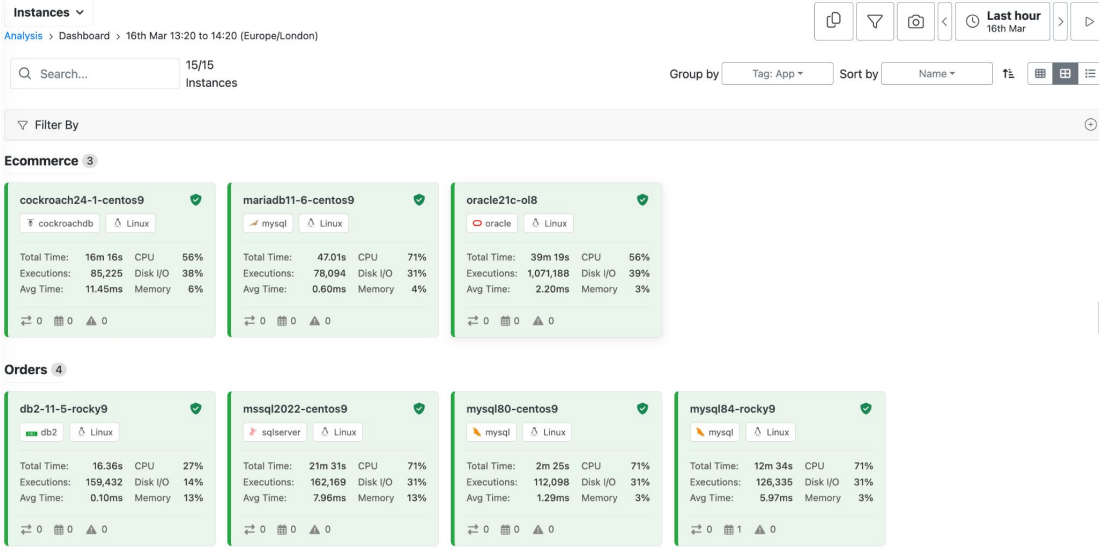
- Go to Settings -> Tags
  - Create a new tag if necessary
  - Fill in values for existing tabs to help classify each instance

Copyright © 2026, Application Performance - Confidential

Using Tags

At the main dashboard (Analysis->Dashboard), change group by to Tags->App





Copyright © 2026, Application Performance - Confidential

DB marlin

21. Configure Authentication

Problem

You want to control access to DBmarlin by enforcing password access.

Solution

DBmarlin provides basic authentication using an `htpasswd` file mechanism. Documentation is provided for this purpose here

Copyright © 2026, Application Performance - Confidential

22. Configure Role-based Access Control

Problem

You want to define what access each user has to databases within DBmarlin

Solution

DBmarlin provides Role-based Access Control (RBAC), Access control is tag based (See this section for how to use tags) and is documented here

Copyright © 2026, Application Performance - Confidential

Instance administrators (6.10+)

1. Log in as an admin user and go to Settings -> Logins
2. Edit the login and tick Instance administrator
3. Assign the tags that select the instances the user should manage
4. Save the login

Instance administrators can change collection settings and start or stop monitoring, but only on instances that match their tags. Creating and deleting instances is still limited to full admins.

Copyright © 2026, Application Performance - Confidential

23. Configure Single sign-on (SSO)

Problem

You want to allow users to log in to DBmarlin using your organisation's existing identity provider (idP), such as Microsoft Entra, Google or Okta, instead of managing local usernames and passwords

Solution

DBmarlin supports Single sign-on using OAuth 2.0 for authorisation and OIDC for identity.

Copyright © 2026, Application Performance - Confidential

Step 1 - Configure Authentication

Configure authentication first
See this recipe to configure authentication.

You must add one local user before setting up SSO

Copyright © 2026, Application Performance - Confidential

Step 2 - Register with your IdP

Create an app registration for DBmarlin with your identity provider.

Step-by-step guides are available for Microsoft Entra, Google and Okta:
https://docs.dbmarlin.com/docs/getting-started/access-control/sso

Copyright © 2026, Application Performance - Confidential

Step 3 - Add SSO parameters

Linux: add the `-Dsso` parameters to the systemd service definition for `dbmarlin-tomcat`, then restart the service.

Windows: append the `-Dsso` parameters to the Java Options in `DBmarlinTomcatw.exe` (in `DBmarlin\Tomcat\bin`), then restart the DBmarlin Tomcat service.

The parameters for each provider are listed in the documentation.

Copyright © 2026, Application Performance - Confidential

24. Ask an AI assistant about database performance

Problem

You want to investigate a performance problem without clicking through many screens, or you want your AI assistant (Claude, Cursor or VS Code) to answer questions about your databases using real monitoring data.

Solution

The optional DBmarlin MCP server (6.10+) lets AI assistants query DBmarlin in plain language, for example "triage my SQL Server instance between 9 and 10am this morning". All tools are read-only, and each user only sees the instances their access control allows.

Copyright © 2026, Application Performance - Confidential

Install the DBmarlin MCP server

1. Deploy the MCP server
   a. Copy `mcp##<version>.war` from the `extras` folder to `tomcat/webapps`. Tomcat deploys it automatically.
2. Check the nginx configuration
   a. `nginx/conf/shared_server.conf` contains a `location ~ ^/mcp(/|$)` block (shipped from 6.10)
3. Enable authentication (strongly recommended)
   a. Each user then only sees the instances they are allowed to see

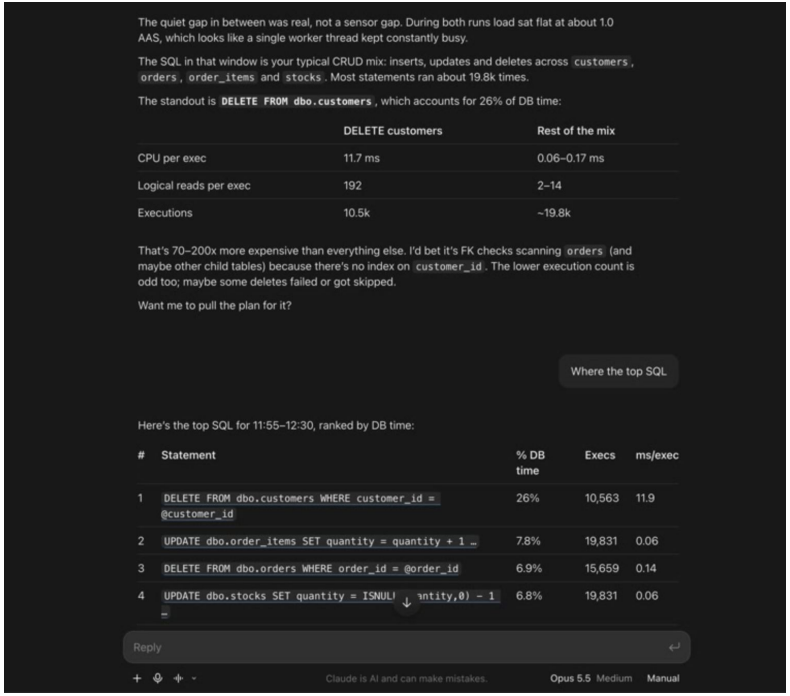
Reference documentation

https://docs.dbmarlin.com/docs/integrations/mcp-server

Copyright © 2026, Application Performance - Confidential

Ask an AI assistant

Triage an instance



The quiet gap in between was real, not a sensor gap. During both runs load sat flat at about 1.0 AAS, which looks like a single worker thread kept constantly busy.

The SQL in that window is your typical CRUD mix: inserts, updates and deletes across `customers`, `orders`, `order_items` and `stocks`. Most statements ran about 19.8k times.

The standout is `DELETE FROM dbo.customers`, which accounts for 26% of DB time:

	DELETE customers	Rest of the mix
CPU per exec	11.7 ms	0.06–0.17 ms
Logical reads per exec	192	2–14
Executions	10.5k	~19.8k

That's 70–200x more expensive than everything else. I'd bet it's FK checks scanning `orders` (and maybe other child tables) because there's no index on `customer_id`. The lower execution count is odd too; maybe some deletes failed or got skipped.

Want me to pull the plan for it?

Where the top SQL

Here's the top SQL for 11:55–12:30, ranked by DB time:

#	Statement	% DB time	Execs	ms/exec
1	`DELETE FROM dbo.customers WHERE customer_id = @customer_id`	26%	10,563	11.9
2	`UPDATE dbo.order_items SET quantity = quantity + 1 …`	7.8%	19,831	0.06
3	`DELETE FROM dbo.orders WHERE order_id = @order_id`	6.9%	15,659	0.14
4	`UPDATE dbo.stocks SET quantity = ISNULL ntity,0) - 1 =`	6.8%	19,831	0.06

Reply

Claude is AI and can make mistakes. Opus 5.5 Medium Manual

Ask in plain language: "Triage my SQL Server instance between 9 and 10am this morning".

The assistant compares with the same time last week and the period before, then drills into SQL, waits and changes.

The report includes links back to the matching DBmarlin screens.

Copyright © 2026, Application Performance - Confidential

DB marlin

25. Monitor SQL Server Always On availability groups

Problem

You run SQL Server Always On availability groups and need to know straight away if a replica disconnects, data stops moving or a failover happens, and how much data you would lose if a secondary took over now.

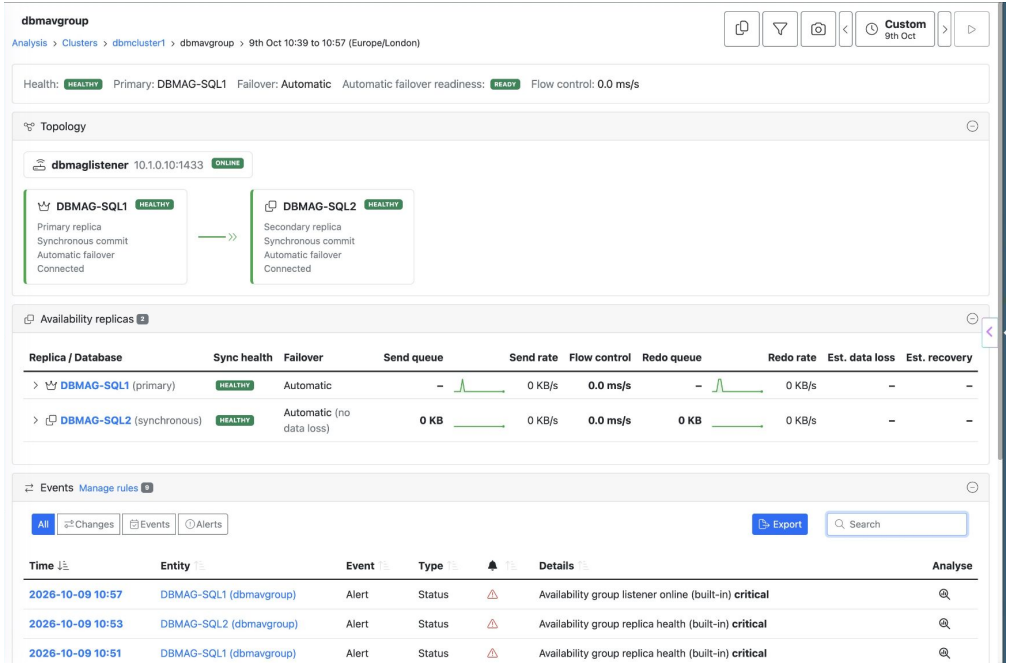
Solution

DBmarlin (6.9+) finds clusters and availability groups automatically from your monitored SQL Server instances. It shows replica health, send and redo queues, estimated data loss (RPO) and recovery time (RTO), and can alert on failovers and health changes.

Copyright © 2026, Application Performance - Confidential

Monitor availability groups

Find your clusters



Go to Analysis -> Clusters. Clusters are found automatically, there is nothing to set up.

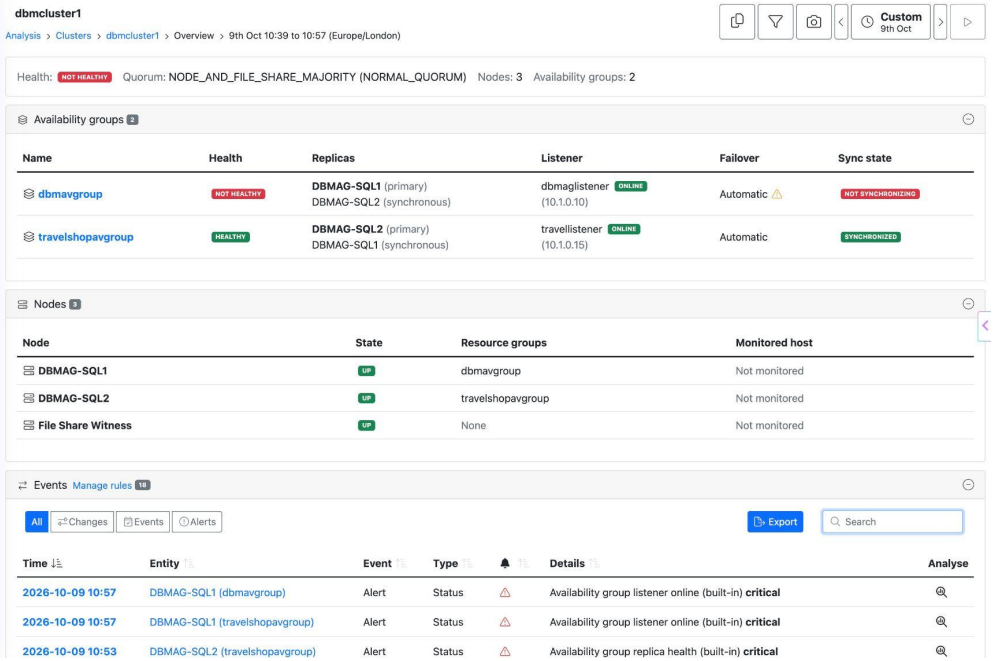
The badge shows how many clusters were found.

Tip: add every replica as a DBmarlin instance for the most accurate rates.

Copyright © 2026, Application Performance - Confidential

Monitor availability groups

Check availability group health



Click an availability group name, or the availability group badge in a SQL Server instance header.

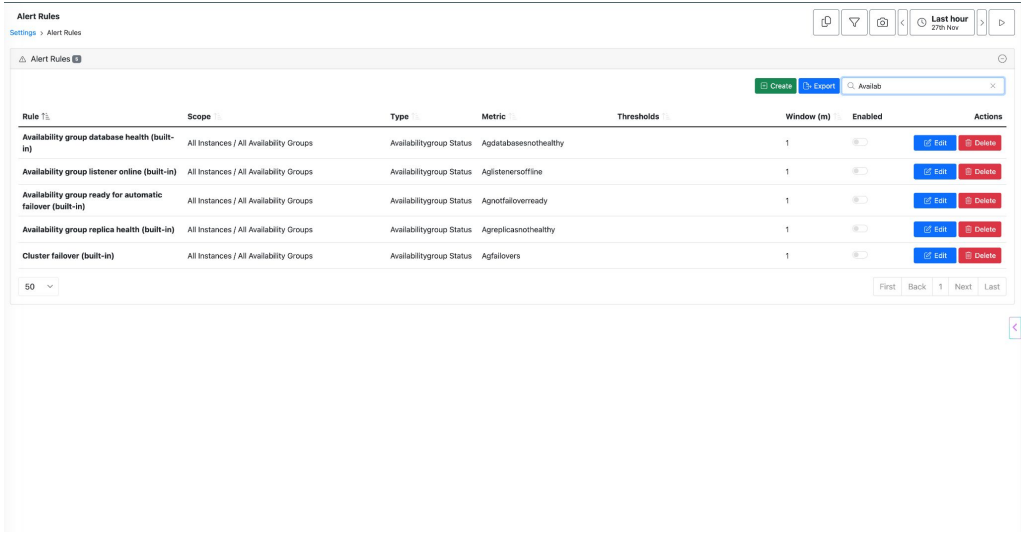
The topology shows the listener, the primary (crown icon) and secondaries. A cross on an arrow means data movement has stopped.

Est. data loss and Est. recovery show your current RPO and RTO.

Copyright © 2026, Application Performance - Confidential

Monitor availability groups

Turn on the alerts



Open Alert Rules in the settings menu.

Enable the five built-in Availability Group rules: replica health, database health, listener online, ready for automatic failover and cluster failover.

Built-in rules are disabled by default.

Copyright © 2026, Application Performance - Confidential

26. Monitor SQL Server backups

Problem

A database backup fails silently, or log backups stop, and nobody notices until a restore is needed. You want to know which databases are not backed up and how much data you could lose.

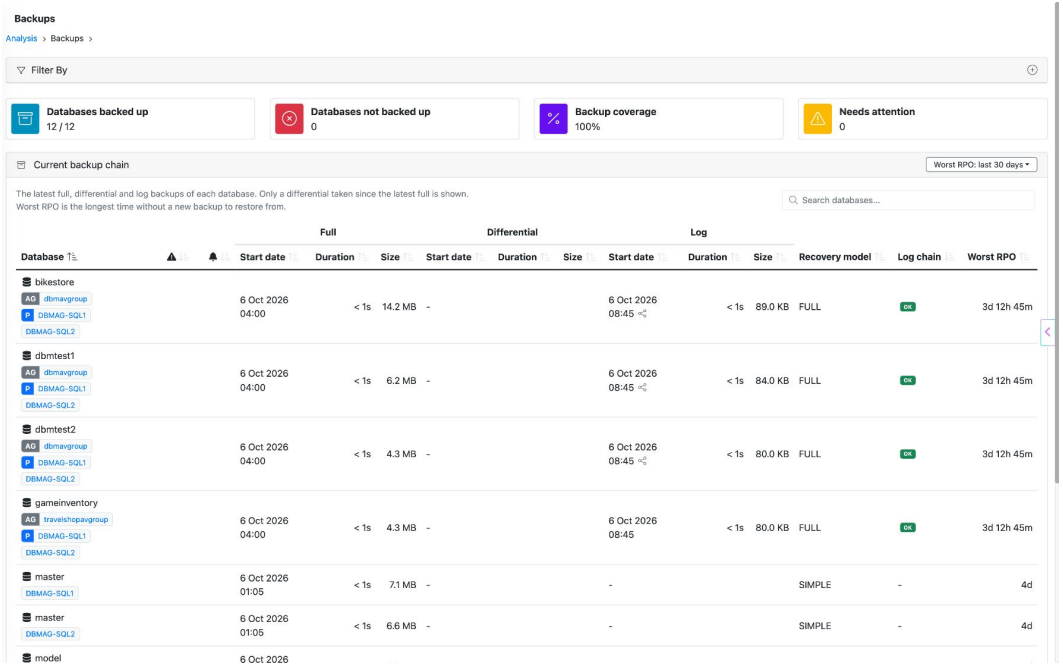
Solution

DBmarlin (6.10+) monitors SQL Server for failed backups, overdue backups and broken backup chains, and shows how far back each database can be restored. Backup history is collected by default.

Copyright © 2026, Application Performance - Confidential

Monitor SQL Server backups

Open the Backups screen



Go to Analysis -> Backups.

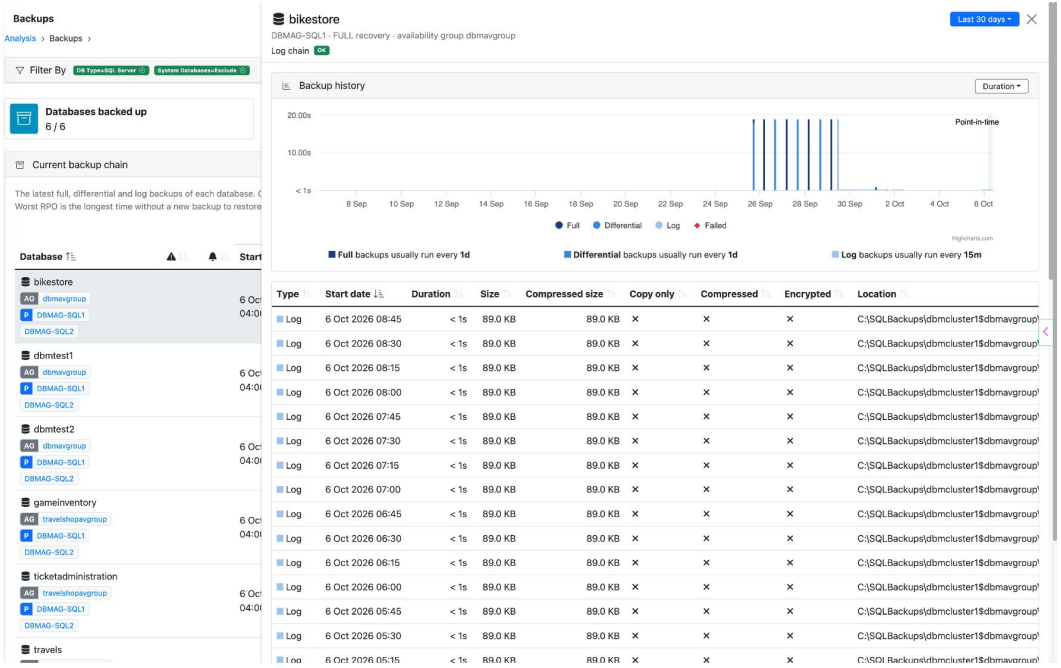
Summary cards show databases backed up, not backed up, backup coverage and Needs attention. Click a card to filter the table.

Worst RPO shows the most data you could have lost.

Copyright © 2026, Application Performance - Confidential

Monitor SQL Server backups

Check the log chain and history



Log chain shows OK, Broken, Warning or Unverified. Hover for the reason and the point-in-time restore window.

Click a database to see its backup history, with failed backups and chain breaks marked on the chart.

Copyright © 2026, Application Performance - Confidential

Turn on backup alerts

1. Open Alert Rules in the settings menu
2. Enable the built-in backup rules
    a. Backup failed
    b. Full backup overdue (warning 25 hours, critical 49 hours)
    c. Log backup overdue (warning 60 minutes, critical 240 minutes)
    d. Backup chain broken
3. Change the thresholds to match your backup schedule
4. Backup alerts also appear on the Events screen

Reference documentation

https://docs.dbmarlin.com/docs/using-dbmarlin/backups

Copyright © 2026, Application Performance - Confidential

27. Find repeated query patterns with Grouped SQL

Problem

Your top statements list shows many similar queries that differ only by their literal values. No single statement looks expensive, but together they use a lot of database time.

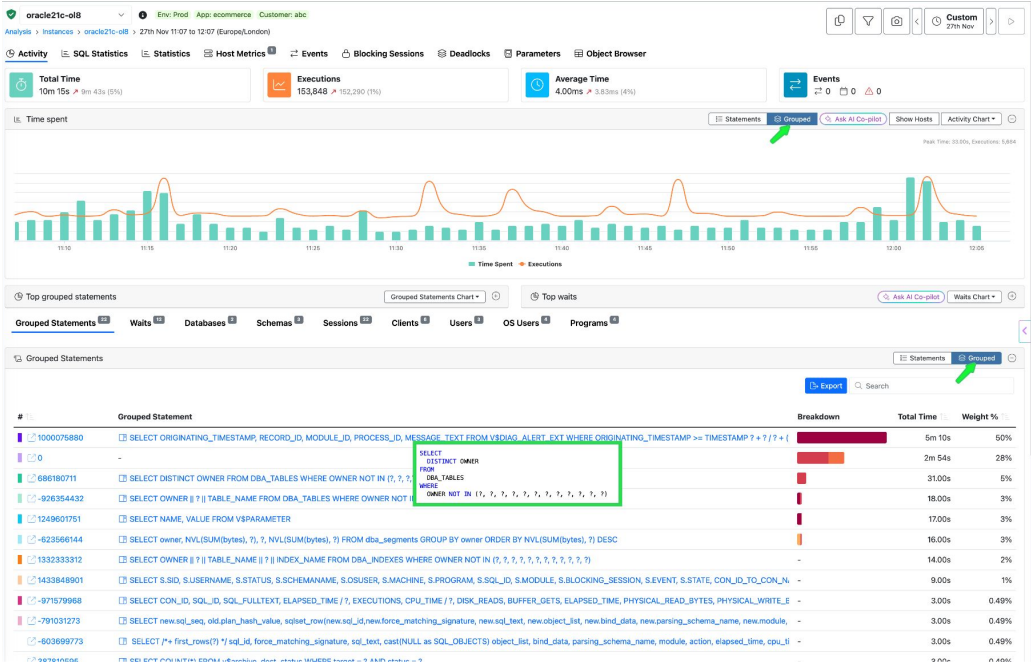
Solution

The Statements / Grouped toggle (6.10+) on the Activity and SQL Statistics views groups SQL statements that differ only by literal values, so you can spot a repeated query pattern that runs with many different parameters.

Copyright © 2026, Application Performance - Confidential

Find repeated query patterns

Switch from Statements to Grouped



Open an instance and scroll down to the list of statements.

Switch the toggle from Statements to Grouped. Statements that differ only by literal values are added together.

A pattern with many executions is a good candidate for bind variables or caching.

Copyright © 2026, Application Performance - Confidential

28. Monitor Oracle multitenant (CDB and PDBs)

Problem

Your Oracle databases run as pluggable databases (PDBs) inside a container database (CDB). You want to monitor them all without adding each PDB separately, and still see which PDB is using the time.

Solution

DBmarlin (6.10+) supports Oracle CDBs and their PDBs. Point DBmarlin at the CDB and break down activity by Database (CDB/PDB), Schema and OS User.

Copyright © 2026, Application Performance - Confidential

Create a common user in the CDB

1. Create a user with the `C##` prefix in the CDB
   a. `CREATE USER C##dbmarlin IDENTIFIED BY <password> default tablespace SYSTEM temporary tablespace TEMP;`
2. Grant access across all containers
   a. `GRANT CREATE SESSION to c##dbmarlin CONTAINER=ALL;`
   b. `GRANT SELECT_CATALOG_ROLE to c##dbmarlin CONTAINER=ALL;`
   c. `ALTER USER c##dbmarlin SET CONTAINER_DATA=ALL CONTAINER=CURRENT;`
   d. `GRANT SELECT ANY DICTIONARY TO C##dbmarlin CONTAINER=ALL;`
3. Add the CDB as a database instance in Settings -> Database Instances

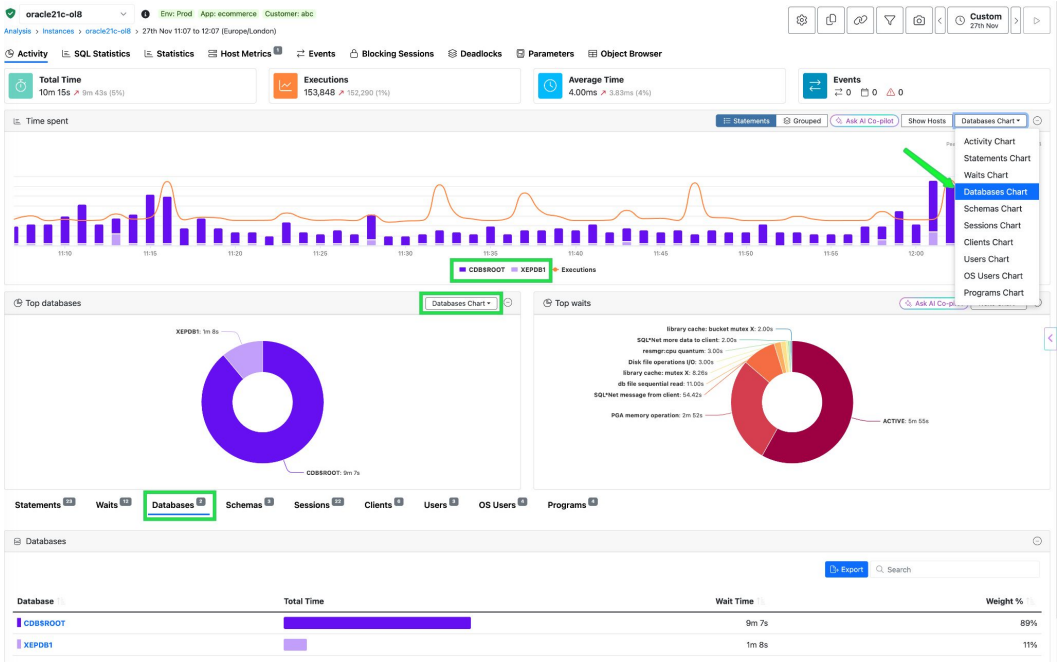
Reference documentation

https://docs.dbmarlin.com/docs/monitored-technologies/databases/oracle

Copyright © 2026, Application Performance - Confidential

Monitor Oracle multitenant

Break down activity by PDB



Open the Oracle CDB instance.

Use the Databases tab to see which PDB is using the time.

Drill into a PDB, then break down by Schema or OS User.

Copyright © 2026, Application Performance - Confidential

Questions?



Copyright © 2026, Application Performance - Confidential