

Enabling the MySQL database general and slow query logs in Amazon Lightsail

Last updated: March 15, 2019

The [general and slow query logs](#) are disabled by default for MySQL databases in Amazon Lightsail. You can enable these logs, and begin collecting data, by updating a few database parameters. Update the database parameters by using the Lightsail API, AWS Command Line Interface (AWS CLI), or SDKs. In this guide, we show you how to use the AWS CLI to update your database parameters and enable the general and slow query logs. We also provide additional options for controlling the general and slow query logs, and how log data retention is handled.

Prerequisite

If you haven't done so already, install and configure the AWS CLI. For more information, see [Configuring the AWS Command Line Interface to work with Amazon Lightsail](#).

Enable the general and slow query logs in the Lightsail console

To enable the general and slow query logs in the Lightsail console, you must update the [general_log](#) and [slow_query_log](#) database parameters with a value of **1**, and the [log_output](#) parameter with a value of **FILE**.

To enable the general and slow query logs in the Lightsail console

1. Open a Terminal or Command Prompt window.
2. Enter the following command to update the [general_log](#) parameter to a value of **1**, which is true, or enabled.

```
aws lightsail update-relational-database-parameters --region Region --  
relational-database-name DatabaseName --parameters  
"parameterName=general_log,parameterValue=1,applyMethod=pending-reboot"
```

In the command, replace:

- *DatabaseName* with the name of your database.
 - *Region* with the AWS Region of your database.
3. Enter the following command to update the [slow_query_log](#) parameter to a value of **1**, which is true, or enabled.

```
aws lightsail update-relational-database-parameters --region Region --  
relational-database-name DatabaseName --parameters  
"parameterName=slow_query_log,parameterValue=1,applyMethod=pending-reboot"
```

In the command, replace:

- *DatabaseName* with the name of your database.
- *Region* with the AWS Region of your database.

4. Enter the following command to update the `log_output` parameter to a value of `FILE`, which writes the log data to a system file and enables it to be displayed in the Lightsail console.

```
aws lightsail update-relational-database-parameters --region Region --relational-database-name DatabaseName --parameters "parameterName=log_output,parameterValue=FILE,applyMethod=pending-reboot"
```

In the command, replace:

- *DatabaseName* with the name of your database.
- *Region* with the AWS Region of your database.

5. Enter the following command to reboot the database and make the changes effective.

```
aws lightsail reboot-relational-database --region Region --relational-database-name DatabaseName
```

In the command, replace:

- *DatabaseName* with the name of your database.
- *Region* with the AWS Region of your database.

At this point, your database becomes unavailable while it reboots. Wait a few minutes, then sign in to the [Lightsail console](#) to view the general and slow query logs for your database. For more information, see [Viewing your database logs and history in Amazon Lightsail](#). **Note**

For more information about updating database parameters, see [Updating database parameters in Amazon Lightsail](#).

Control additional database log options

To control additional options for the MySQL general and slow query logs, update the following parameters:

- `log_output` — Set this parameter to `TABLE`. This writes general queries to the `mysql.general_log` table, and slow queries to the `mysql.slow_log` table. You can also set the `log_output` parameter to `NONE` to disable logging. **Note**

Setting the `log_output` parameter to `TABLE` disables the general and slow query log data from displaying in the Lightsail console. Instead, you must refer to the `mysql.general_log` and `mysql.slow_log` tables on your database to view the log data.

- `long_query_time` — To prevent fast-running queries from being logged in the slow query log, specify a value for the shortest query execution time to be logged, in seconds. The default is 10 seconds, and the minimum is 0. If the `log_output` parameter is set to `FILE`, you can specify a floating point value that goes to microsecond resolution. If the `log_output` parameter is set to `TABLE`, you must specify an

integer value with second resolution. Only queries whose execution time exceeds the `long_query_time` parameter value are logged. For example, setting `long_query_time` to 0.1 prevents any query that runs for less than 100 milliseconds from being logged.

- `log_queries_not_using_indexes` — To log all queries that do not use an index to the slow query log, set to 1. The default is 0. Queries that do not use an index are logged even if their execution time is less than the value of the `long_query_time` parameter.

Log data retention

When logging is enabled, table logs are rotated, or log files are deleted, at regular intervals. This measure is a precaution to reduce the possibility of a large log file either blocking database use or affecting performance.

When the `log_output` parameter is set to `FILE` or `TABLE`, logging is handled as follows:

- When `FILE` logging is enabled, log files are examined every hour and log files older than 24 hours are deleted. In some cases, the remaining combined log file size after the deletion might exceed the threshold of 2 percent of a database's allocated space. In these cases, the largest log files are deleted until the log file size no longer exceeds the threshold.
- When `TABLE` logging is enabled, log tables are rotated every 24 hours in some cases.

This rotation occurs if the space used by the table logs is more than 20 percent of the allocated storage space or the size of all logs combined is greater than 10 GB.

If the amount of space used for a database is greater than 90 percent of the database's allocated storage space, then the thresholds for log rotation are reduced.

Log tables are then rotated if the space used by the table logs is more than 10 percent of the allocated storage space or the size of all logs combined is greater than 5 GB.

You can subscribe to the `low_free_storage` event to be notified when log tables are rotated to free up space.

- When log tables are rotated, the current log table is copied to a backup log table and the entries in the current log table are removed. If the backup log table already exists, then it is deleted before the current log table is copied to the backup. You can query the backup log table. The backup log table for the `mysql.general_log` table is named `mysql.general_log_backup`. The backup log table for the `mysql.slow_log` table is named `mysql.slow_log_backup`.
- You can rotate the `mysql.general_log` table by calling the `mysql.rds_rotate_general_logprocedure`. You can rotate the `mysql.slow_log` table by calling the `mysql.rds_rotate_slow_logprocedure`.
- Table logs are rotated during a database version upgrade.