Updating database parameters in Amazon Lightsail

Last updated: January 3, 2020

Database parameters, also known as database system variables, define fundamental properties of a managed database in Amazon Lightsail. For example, you can define a database parameter to limit the number of database connections, or define another parameter to limit the database buffer pool size. This guide shows you how to get a list of the parameters for your managed database, and how to update them using the AWS Command Line Interface (AWS CLI).

Note
For more information about MySQL system variables, refer to the MySQL 5.6, MySQL 5.7, or MySQL 8.0 documentation. For more information about PostgreSQL system variables, refer to the PostgreSQL 9.6, PostgreSQL 10, or PostgreSQL 11 documentation.

Prerequisites

- 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.

Get a list of available database parameters

The database parameters differ depending on the database engine; therefore, you should get a list of the parameters available for your managed database. This will allow you to decide which parameter you want to modify, and the way in which that parameter becomes effective.

To get a list of available database parameters

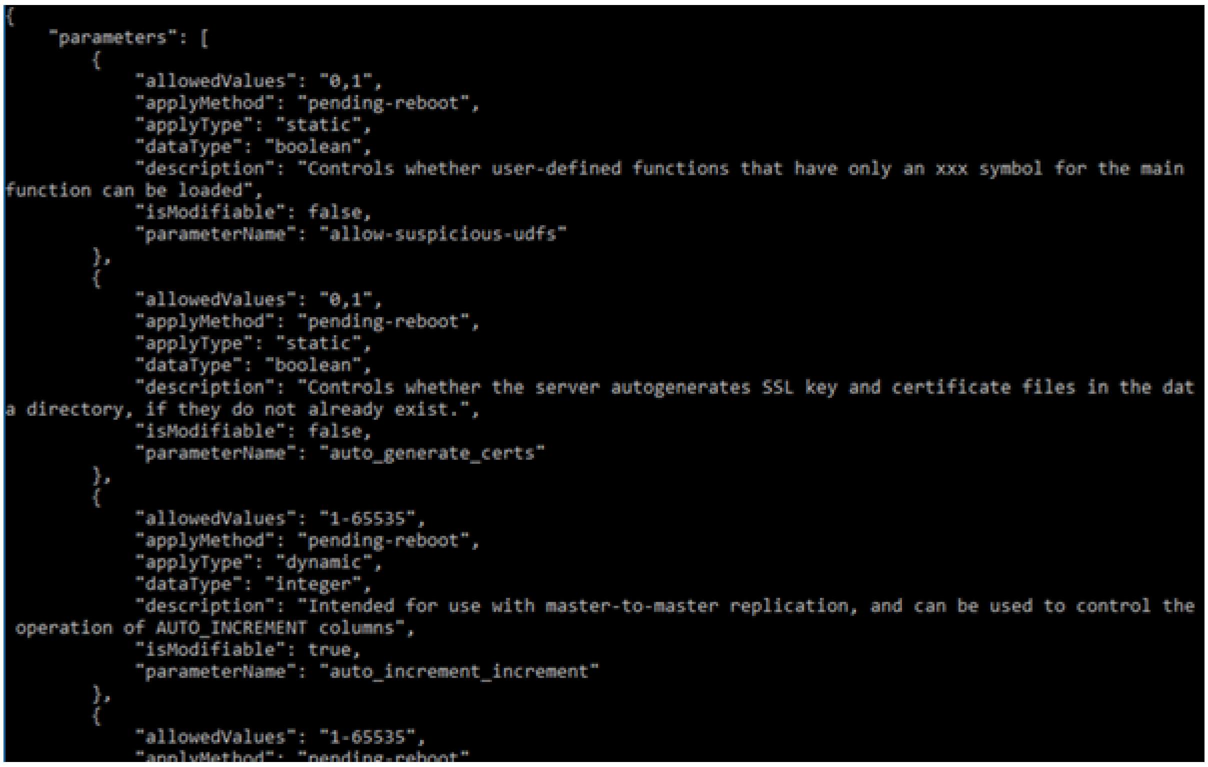
1. Open a Terminal or Command Prompt window.

2. Enter the following command to get a list of parameters for your database..

   ```
   aws lightsail get-relational-database-parameters --relational-database-name
   DatabaseName
   ```

   In the command, replace *DatabaseName* with the name of your database.

You should see a result similar to the following example:



```
{
    "parameters": [
        {
            "allowedValues": "0,1",
            "applyMethod": "pending-reboot",
            "applyType": "static",
            "dataType": "boolean",
            "description": "Controls whether user-defined functions that have only an xxx symbol for the main
function can be loaded",
            "isModifiable": false,
            "parameterName": "allow-suspicious-udfs"
        },
        {
            "allowedValues": "0,1",
            "applyMethod": "pending-reboot",
            "applyType": "static",
            "dataType": "boolean",
            "description": "Controls whether the server autogenerates SSL key and certificate files in the dat
a directory, if they do not already exist.",
            "isModifiable": false,
            "parameterName": "auto_generate_certs"
        },
        {
            "allowedValues": "1-65535",
            "applyMethod": "pending-reboot",
            "applyType": "dynamic",
            "dataType": "integer",
            "description": "Intended for use with master-to-master replication, and can be used to control the
 operation of AUTO_INCREMENT columns",
            "isModifiable": true,
            "parameterName": "auto_increment_increment"
        },
        {
            "allowedValues": "1-65535",
```

Note

A next page token ID is listed if the parameter results are paginated. Make note of the next page token ID and use it as shown in the next step to view the next page of parameter results.

3. If your results are paginated, use the following command to view the additional set of parameters. Otherwise, skip to the next step.

```
aws lightsail get-relational-database-parameters --relational-database-name
DatabaseName --page-token NextPageTokenID
```

In the command, replace:

- *DatabaseName* with the name of your database.
- *NextPageTokenID* with the next page token ID.

The result displays the following information for each database parameter:

- **Allowed values** — Specifies the valid range of values for the parameter.
- **Apply method** — Specifies when the parameter change is applied. Allowed options are `immediate` or `pending-reboot`. See the following apply type for more information about how to define the apply method.
- **Apply type** — Specifies the engine-specific submission type. If `dynamic` is listed, the parameter can be applied with an `immediate` apply method and the database will begin using the new parameter value immediately. If `static` is listed, the parameter can only be applied with a `pending-reboot` apply method and the database will begin using the new parameter only after it's restarted.
- **Data type** — Specifies the valid data type for the parameter.
- **Description** — Provides a description of the parameter.

  - **Is modifiable** — A Boolean value indicating whether the parameter can be modified. If `true` is listed, then the parameter can be modified.
  - **Parameter name** — Specifies the name of the parameter. Use this value together with the `update relational database` operation and the `parameter name` parameter.

4. Find the parameter you want to change, and make note of the parameter name, allowed values, and apply method. We recommend copying the parameter name to your clipboard to avoid entering it incorrectly. To do that, highlight the parameter name and press **Ctrl+C** if you're using Windows, or **Cmd+C** if you're using macOS, to copy it to your clipboard. Then, press **Ctrl+V** or **Cmd+V** as appropriate to paste it.

   After you identify the name of the parameter that you want to modify, continue to the next section of this guide to change the parameter to your desired value.

Update your database parameters

After you have the name of the parameter you want to change, perform the following steps to modify the parameter for your managed database in Lightsail:

To update your database parameters

- Enter the following command into a terminal or command prompt window to update a parameter for your managed database.

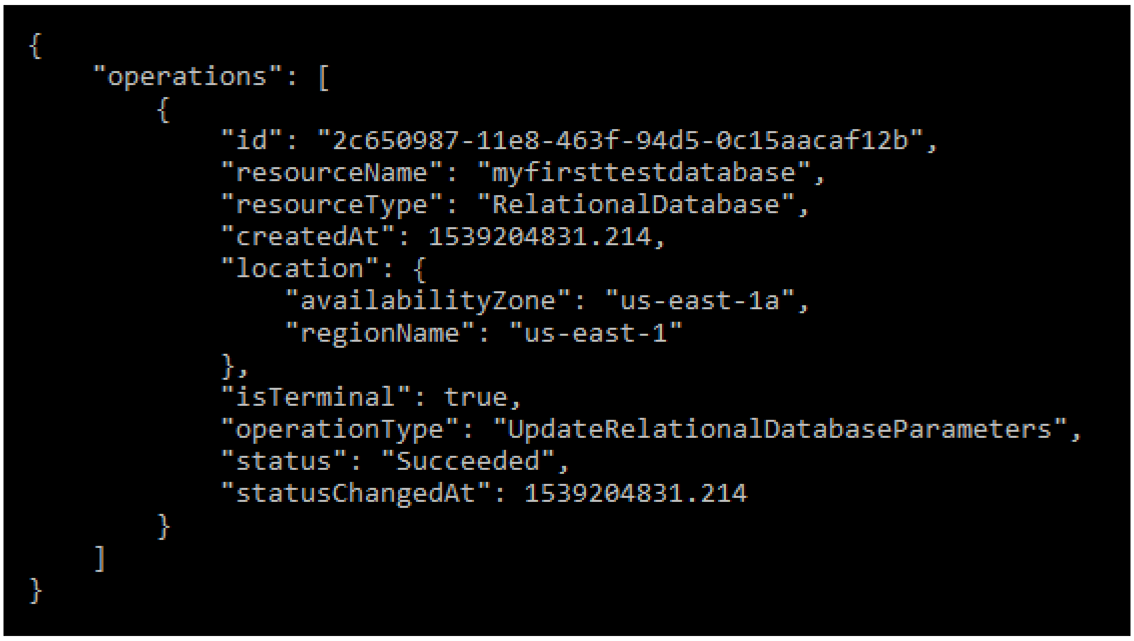
  ```
  aws lightsail update-relational-database-parameters --relational-database-name DatabaseName --parameters "parameterName=ParameterName,parameterValue=NewParameterValue,applyMethod=ApplyMethod"
  ```

  In the command, replace:

  - *DatabaseName* with the name of your database.
  - *ParameterName* with the name of the parameter you want to modify.
  - *NewParameterValue* with the new value of the parameter.
  - *ApplyMethod* with the apply method for the parameter.

    If the parameter's apply type is `dynamic`, the parameter can be applied with an `immediate` apply method and the database will begin using the new parameter value immediately. However, if the parameter apply type is `static`, the parameter can only be applied with a `pending-reboot` apply method and the database will begin using the new parameter only after it's restarted.

You should see a result similar to the following example:



```
{
    "operations": [
        {
            "id": "2c650987-11e8-463f-94d5-0c15aacaf12b",
            "resourceName": "myfirsttestdatabase",
            "resourceType": "RelationalDatabase",
            "createdAt": 1539204831.214,
            "location": {
                "availabilityZone": "us-east-1a",
                "regionName": "us-east-1"
            },
            "isTerminal": true,
            "operationType": "UpdateRelationalDatabaseParameters",
            "status": "Succeeded",
            "statusChangedAt": 1539204831.214
        }
    ]
}
```

The database parameter is updated depending on the apply method used.