

How to Install PHPMyAdmin on Linux?

written by sysadmin | 12 November 2025

By default, if you want to access the MariaDB database, then you have to access it using the CLI. However, this method is a bit difficult, especially for those who are not used to the CLI. So you need an application that can display the database in the GUI, and you can interact with the database using the application. One application that is often used is the PHPMyAdmin application.

Problem

How to install PHPMyAdmin on Linux?

Solution

phpMyAdmin is an open-source, free MySQL and MariaDB administration tool, and as of this writing (November 2025), this application has version 5.2.3. This application is a web application and is written using PHP. Before installing the PHPMyAdmin application, you must have installed the database on your Linux server, and you must have provided a password for the database. Here is how to install the PHPMyAdmin application on Linux, based on the distro:

Ubuntu/Debian

```
sudo apt update  
sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

RockyLinux/AlmaLinux/CentOS

```
yum install httpd  
yum install php-phpmysql
```

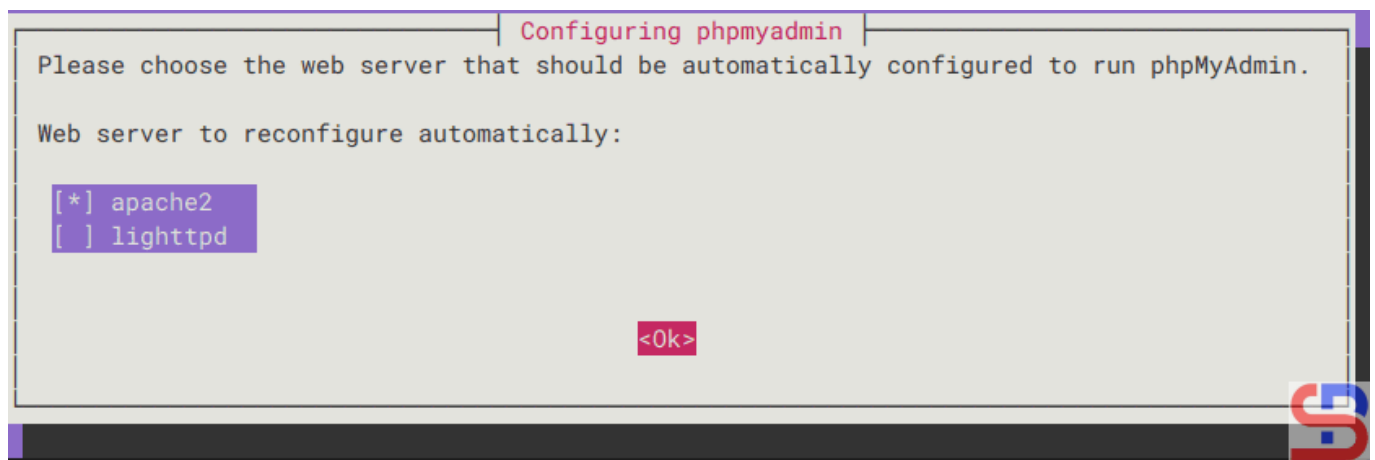
OpenSUSE

```
sudo zypper install apache2
sudo zypper install php7 php7-mysql apache2-mod_php7 phpMyAdmin
```

This article will install PHPMyAdmin on **Ubuntu 24.04** and run the command below:

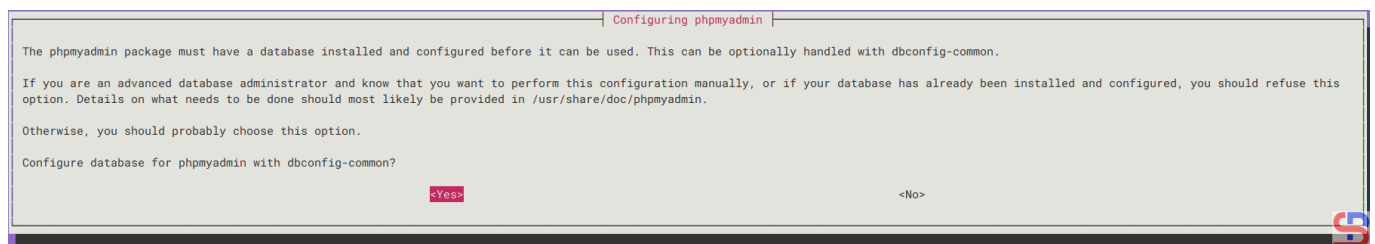
```
sudo apt update
sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

During installation, a pop-up will appear as below:



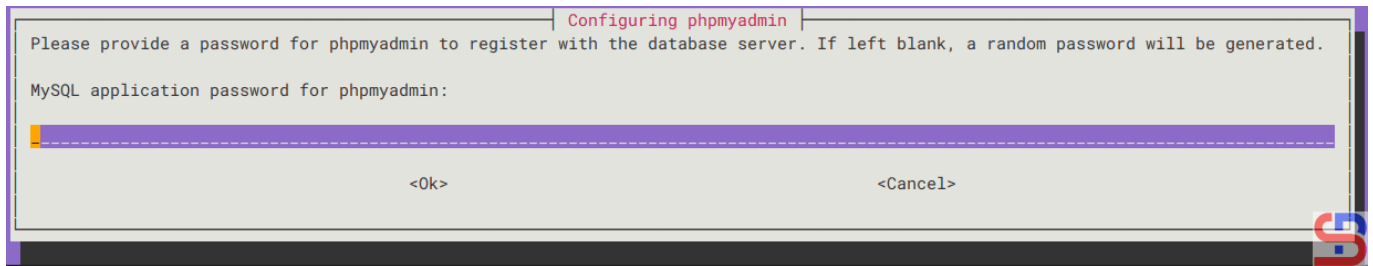
Select the web server

Select the web server used, and I use the Apache web server, and click **OK**. Then there will be another pop-up like below:



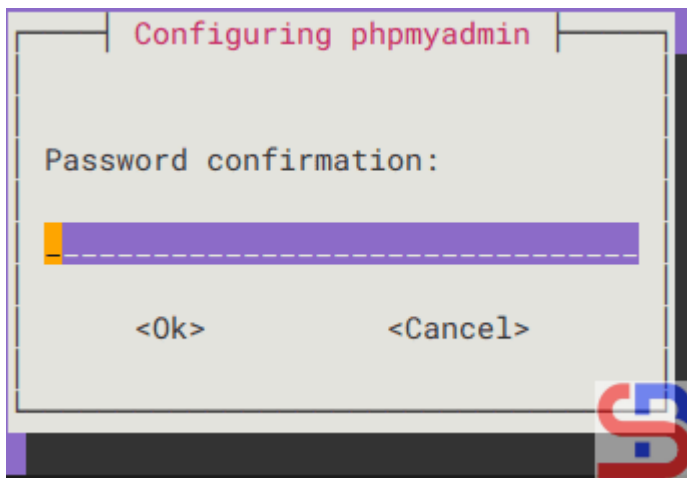
Configuring phpmyadmin

Choose **Yes**, and the installation will continue, but after that, you must enter the password for phpadmin as in the image below:



Enter the password

Enter your password, click **Ok**, and then there will be an email confirmation as in the image below:

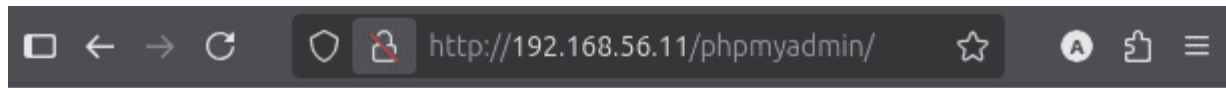


Password confirmation

Enter your password again, click **Ok**, and the installation process will continue. And you have to wait until the installation process is complete. After that, open your browser and type the URL below:

http://your_ip_server/phpmyadmin

There should be a display like the image below:



Language

English

Log in

Username:

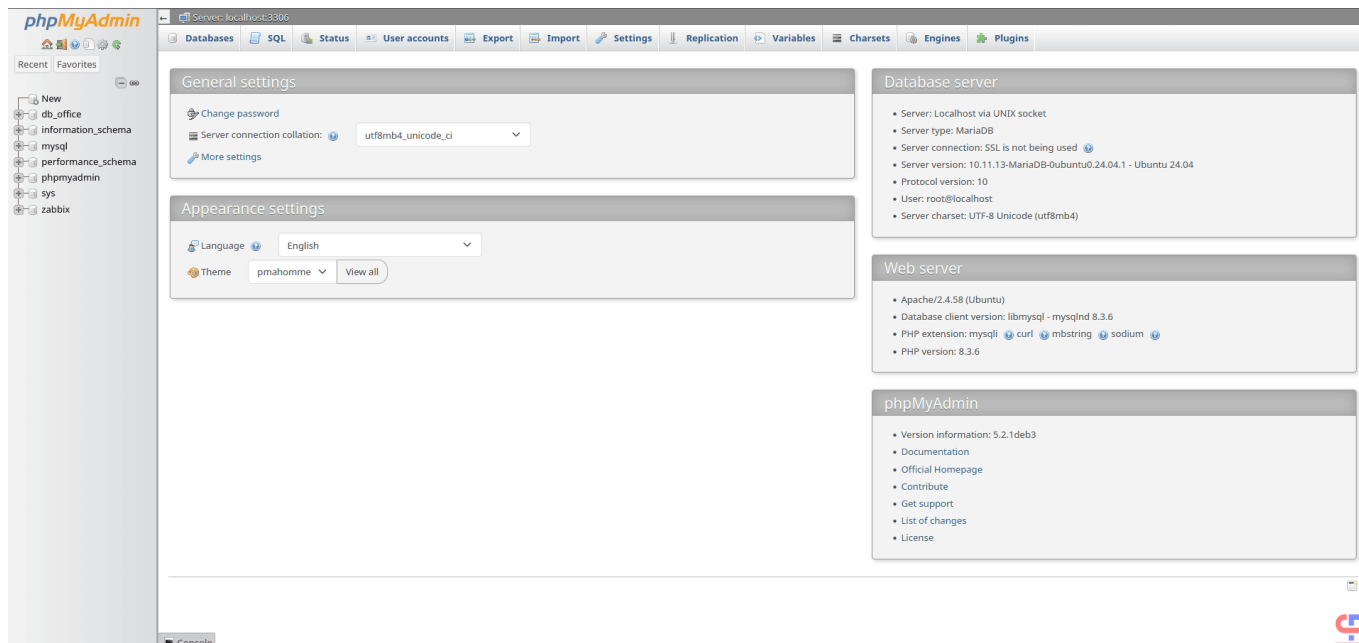
Password:

Log in



Enter username and password

If there is no display like the one above, it looks like you have to [open port](#) 80 on your server. Enter your username and password in your MySQL/MariaDB Database, and if there are no errors, then there will be a display like in the image below:



Access to PHPMyAdmin

To see all the databases in your database, click **Database** as shown in the image below:



Databases

Create database

☐ Check all

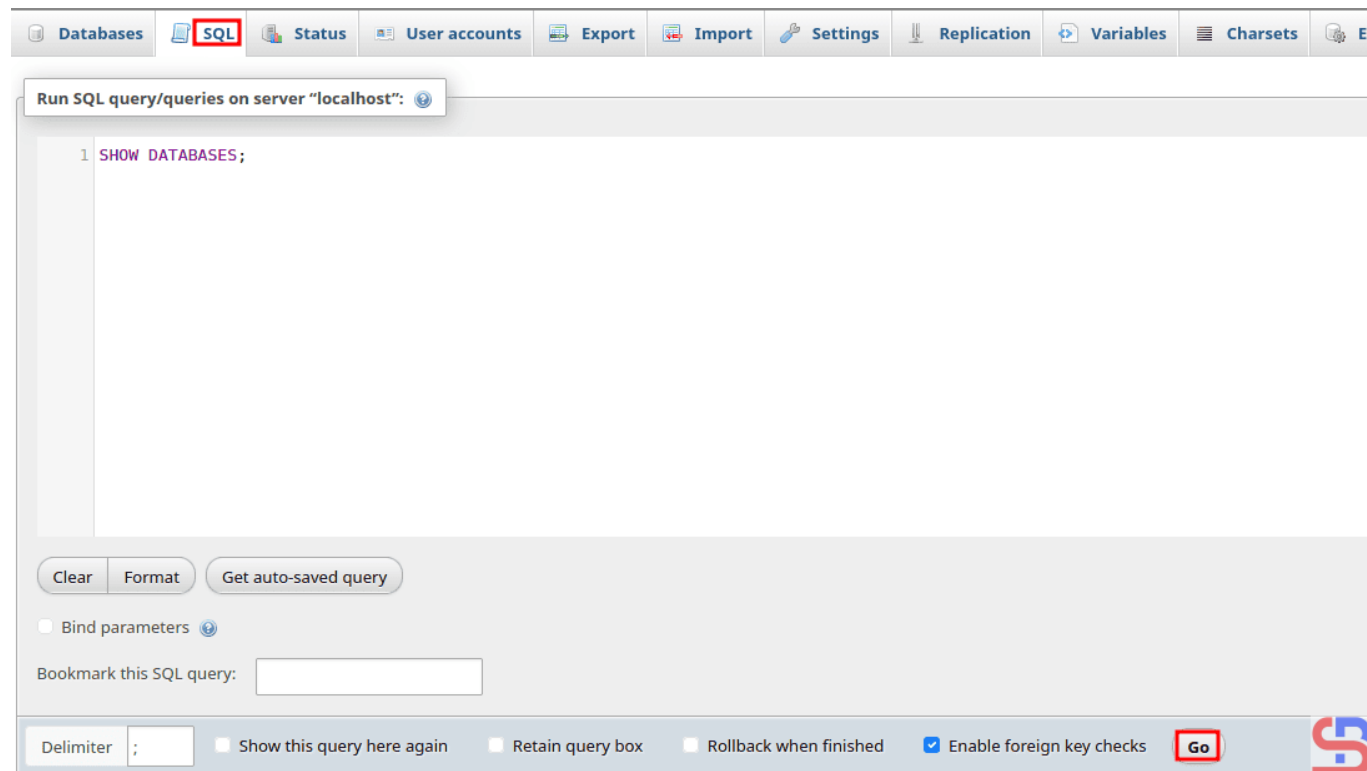
	Database	Collation	Action
☐	db_office	utf8mb4_general_ci	Check privileges
☐	information_schema	utf8mb3_general_ci	Check privileges
☐	mysql	utf8mb4_general_ci	Check privileges
☐	performance_schema	utf8mb3_general_ci	Check privileges
☐	phpmyadmin	utf8mb4_general_ci	Check privileges
☐	sys	utf8mb3_general_ci	Check privileges
☐	zabbix	utf8mb4_bin	Check privileges

Total: 7



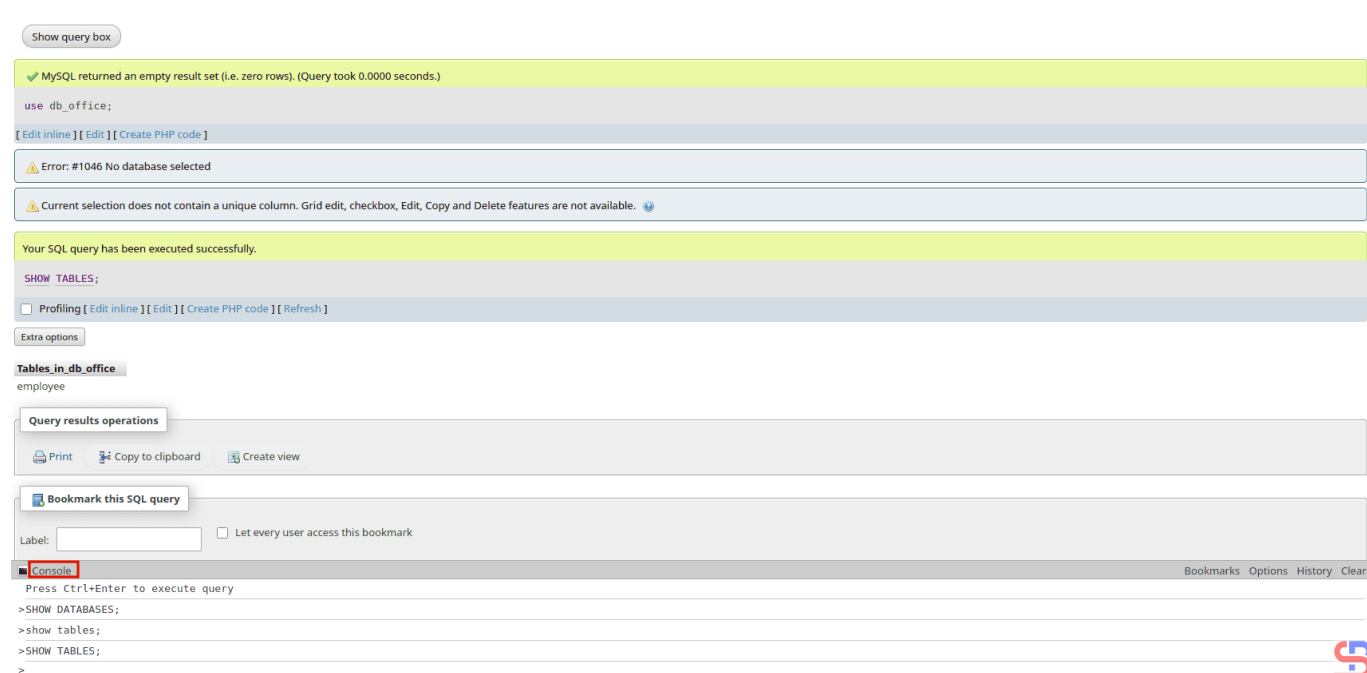
List all databases

You can also perform queries in the SQL section, as in the image below:



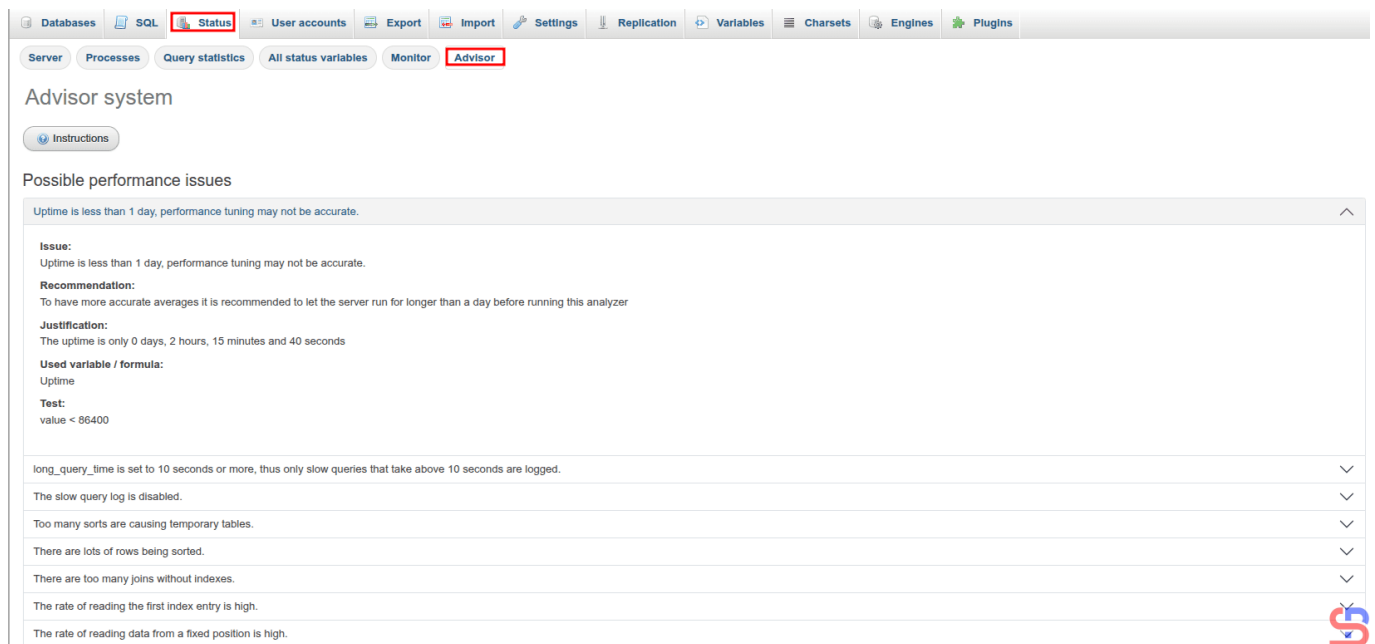
Execute query in PHPMyAdmin

And you can see the history of queries that you have done in PHPMyAdmin in the **console** section, as in the image below:



Show the history of the query in PHPMYAdmin

And in the console section, you can also perform queries by pressing **Ctrl+Enter** after you write the query. In fact, this application also provides an advisor feature to provide recommendations for your MariaDB database, as shown in the image below:



Recommendation for your database in phpMyAdmin

Note

It is highly recommended to restrict access to the PHPMYAdmin application. Therefore, this application should only be allowed on certain IPs, for example, office IPs. And you can make these restrictions in the **/etc/apache2/conf-available/phpmyadmin.conf** file in the **/usr/share/phpmyadmin** section. For example, if you want only IP 192.168.56.1 to be able to access this application, then add the script below:

```
# Limit IP
Order Deny,Allow
Deny from All
Allow from sysadminpedia.com
Allow from 192.168.56.1
Allow from 127.0.0.1
```

```
<Directory /usr/share/phpmyadmin>
Options SymLinksIfOwnerMatch
DirectoryIndex index.php

# limit libapache2-mod-php to files and directories necessary by pma
<IfModule mod_php7.c>
    php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp
    php_admin_value open_basedir /usr/share/phpmyadmin/:/usr/share/doc/phpmyadmin/:/etc/phpmyadmin/:/var/lib/phpmyadmin/:/usr/share/php/:/usr/share/javascript/
</IfModule>

# PHP 8+
<IfModule mod_php.c>
    php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp
    php_admin_value open_basedir /usr/share/phpmyadmin/:/usr/share/doc/phpmyadmin/:/etc/phpmyadmin/:/var/lib/phpmyadmin/:/usr/share/php/:/usr/share/javascript/
</IfModule>

# Limit IP
Order Deny,Allow
Deny from All
Allow from sysadminpedia.com
Allow from 192.168.56.1
Allow from 127.0.0.1
</Directory>
```

IP limitation in PHPMYAdmin

After that, restart your web server. Now you can't open the phpMyAdmin application except at the IP you have specified in the phpmyadmin.conf file

References

en.wikipedia.org
digitalocean.com
tutorialspoint.com
synaptica.info
linuxblog.io

How to Restore the Database in MariaDB?

written by sysadmin | 12 November 2025

[The previous articles](#) have explained how to perform a database backup in MariaDB. This article will explain how to perform a database restore in MariaDB.

Problem

How to restore the database in MariaDB?

Solution

There are several methods to perform the database restore in MariaDB:

1. Restore the entire database

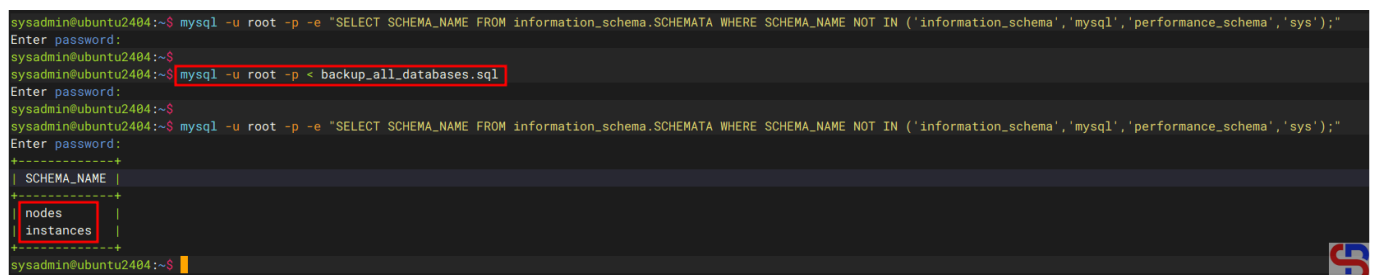
To restore the backup file of the entire database, use the format below:

```
mysql -u username -p < backup_file.sql
```

Use the command below if your backup file name is backup_all_databases.sql:

```
mysql -u root -p < backup_all_databases.sql
```

After you run the above command, the database will be restored in MariaDB as shown in the image below:

A terminal window showing the restoration of all databases in MariaDB. The user runs the command 'mysql -u root -p < backup_all_databases.sql'. The terminal shows the prompt 'Enter password:' and then the command is executed. The output shows a list of schemas: 'nodes' and 'instances'. The user then runs the command 'mysql -u root -p -e "SELECT SCHEMA_NAME FROM information_schema.SCHEMATA WHERE SCHEMA_NAME NOT IN ('information_schema','mysql','performance_schema','sys');"'. The terminal shows the prompt 'Enter password:' and then the command is executed. The output shows a list of schemas: 'nodes' and 'instances'.

Restore all databases in MariaDB

2. Restore a database

Use the following format to restore a database's backup file:

```
mysql -u username -p -e'create database new_database;' < backup_file.sql
```

If you want to restore the nodes database, then you can use the command below:

```
mysql -uroot -p -e'create database nodes;' < backup_nodes_db.sql
```

After you run the command above, MariaDB will restore the database like in the image below:

```
sysadmin@ubuntu2404:~$ mysql -u root -p -e "SELECT SCHEMA_NAME FROM information_schema.SCHEMATA WHERE SCHEMA_NAME NOT IN ('information_schema','mysql','performance_schema','sys');"
Enter password:
sysadmin@ubuntu2404:~$ mysql -u root -p -e "create database nodes;" < backup_nodes_db.sql
Enter password:
sysadmin@ubuntu2404:~$ mysql -u root -p -e "SELECT SCHEMA_NAME FROM information_schema.SCHEMATA WHERE SCHEMA_NAME NOT IN ('information_schema','mysql','performance_schema','sys');"
Enter password:
+-----+
| SCHEMA_NAME |
+-----+
| nodes       |
+-----+
```

Restore a database in MariaDB

3. Restore the table(s)

If you want to restore the table(s), you can follow the format below:

```
mysql -u username -p db_name < table_backup_file.sql
```

So, use the command below if you want to restore the tables in the nodes database:

```
mysql -u root -p nodes < table_backup_file.sql
```

4. Restore a compressed backup file

There are two methods to restore a compressed backup file:

a. Restore the .gz backup file

If you want to restore a .gz backup file, use the format below:

```
gunzip < database.sql.gz | mysql -u username -p -e 'create database new_database;'
```

For example, if you want to restore a database that uses .gz compression, then use the command below:

```
gunzip < nodes_backup_db.sql.gz | mysql -u root -p -e 'create database nodes;'
```

```

sysadmin@ubuntu2404:~$ mysql -u root -p -e "SELECT SCHEMA_NAME FROM information_schema.SCHEMATA WHERE SCHEMA_NAME NOT IN ('information_schema','mysql','performance_schema','sys');"
Enter password:
sysadmin@ubuntu2404:~$ gunzip < nodesdb_backup_file.sql.gz | mysql -u root -p -e 'create database nodes;'
Enter password:
sysadmin@ubuntu2404:~$ mysql -u root -p -e "SELECT SCHEMA_NAME FROM information_schema.SCHEMATA WHERE SCHEMA_NAME NOT IN ('information_schema','mysql','performance_schema','sys');"
Enter password:
+-----+
| SCHEMA_NAME |
+-----+
| nodes       |
+-----+
sysadmin@ubuntu2404:~$

```

Restore a database that is compressed using .gz compression

b. Restore the .gz backup file

If you want to restore a .bz2 backup file, use the format below (but make sure your server has already installed the bzip2 package):

```

bunzip2 < database.sql.bz2 | mysql -u username -p -e 'create database new_database;'

```

For example, if you want to restore a database that uses .bz2 compression, then use the command below:

```

bunzip2 < nodes.sql.bz2 | mysql -u root -p -e 'create database nodes;'

```

```

sysadmin@ubuntu2404:~$ mysql -u root -p -e "SELECT SCHEMA_NAME FROM information_schema.SCHEMATA WHERE SCHEMA_NAME NOT IN ('information_schema','mysql','performance_schema','sys');"
Enter password:
sysadmin@ubuntu2404:~$ bunzip2 < nodes_backup_db.sql.bz2 | mysql -u root -p -e 'create database nodes;'
Enter password:
sysadmin@ubuntu2404:~$ mysql -u root -p -e "SELECT SCHEMA_NAME FROM information_schema.SCHEMATA WHERE SCHEMA_NAME NOT IN ('information_schema','mysql','performance_schema','sys');"
Enter password:
+-----+
| SCHEMA_NAME |
+-----+
| nodes       |
+-----+
sysadmin@ubuntu2404:~$

```

Restore a database that is compressed using .bz2 compression

Note

You can restore a database backup file that is compressed using method number 4, whether you compress when backing up the database or after backing up the database.

References

mariadb.com
stackoverflow.com
serverfault.com

How to Back up the Database in MariaDB?

written by sysadmin | 12 November 2025

I want to back up the database in MariaDB.

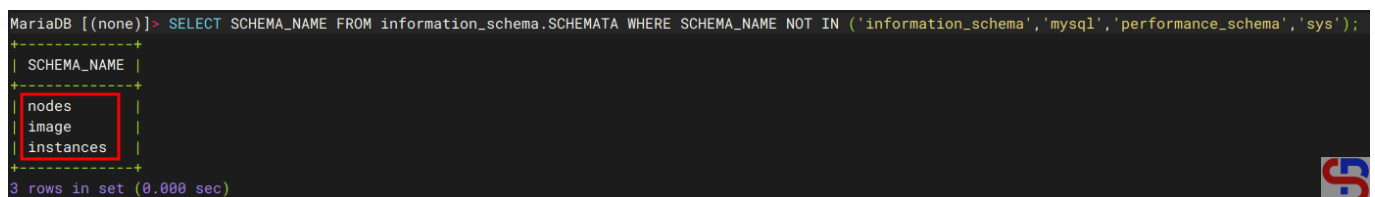
Problem

How to back up the database in MariaDB?

Solution

Before doing a backup, it is highly recommended to ensure that there are no transactions in the database. Maybe you can turn off the application that is connected to the database or turn off the connection to the database so that it will produce a good backup file. In this article, I have 3 databases to use as an example, like the image below:

```
MariaDB [(none)]> SELECT SCHEMA_NAME FROM information_schema.SCHEMATA WHERE SCHEMA_NAME NOT IN ('information_schema','mysql','performance_schema','sys');
+-----+
| SCHEMA_NAME |
+-----+
| nodes       |
| image       |
| instances   |
+-----+
3 rows in set (0.000 sec)
```



Example of databases

And below are the commands to back up the MariaDB database:

1. Back up the entire database

If you want to back up all databases in MariaDB, use the format below:

```
mariadb-dump -u username -p --all-databases > backup_file_name.sql
```

So, run the command below to back up your entire database:

```
mariadb-dump -u root -p --all-databases > backup_all_databases.sql
```

2. Backup a database

If you want to back up a database in MariaDB, use the format below:

```
mariadb-dump -u username -p db_name > backup_file_name.sql
```

So, run the command below if you want to back up the nodes database:

```
mariadb-dump -u root -p nodes > backup_nodes_db.sql
```

3. Backup more than one database

If you want to back up more than one database, use the format below:

```
mariadb-dump -u username -p db1_name db2_name > backup_file_name.sql
```

So, if you want to back up the nodes and image databases, use the format below:

```
mariadb-dump -uroot -p --databases nodes image >  
backup_nodes_and_image_db.sql
```

4. Backup database only

When you back up a database, by default, the database will be backed up in its entirety, both the database and the database structure. But sometimes you just want to back up the database without the structure, so you can use the format below:

```
mariadb-dump -u username -p --no-create-db --no-create-info db_name >
```

dbname_db_only.sql

So, if you want to back up the database only for the nodes database, you can use the command below:

```
mariadb-dump -u root -p --no-create-db --no-create-info nodes >
nodes_db_only.sql
```

5. Backup the database structure only

If you want to back up the database structure only, without the need to back up the database, then you can use the format below:

```
mariadb-dump -u username -p --no-data db_name > dbname_structure_only.sql
```

So, use the command below if you want to back up the structure only for the nodes database:

```
mariadb-dump -u root -p --no-data nodes > nodes_structure_only.sql
```

6. Backup table only

If you want to back up the specific table only in a database, you can use the format below:

```
mariadb-dump -u username -p db_name table1 table2 >
backup_table1_table2_dbname.sql
```

So, if you want to back up 2 tables in the nodes database for babel and category tables, use the command below:

```
mariadb-dump -uroot -p nodes babel category > backup_babel_category_nodes.sql
```

7. Backup with compression

By default, when people perform database backups, they will usually use .sql as an extension of the database backup

file. But actually, you can compress the database backup files when you do a backup. There are 2 methods of compressing database backups, namely using the tar.gz method and the .bz2 method. If you want to use the first method, use the format below:

```
mariadb-dump -u username -p db_name | gzip -9 -c > db_backup_file.sql.gz
```

Suppose you want to compress the nodes database, then use the command below:

```
mariadb-dump -u root -p nodes | gzip -9 -c > nodesdb_backup_file.sql.gz
```

And if you want to use bz2 compression when backing up databases, then make sure that the bz2 package is already installed on your server. If the package is not already installed, use the command below:

RockyLinux/AlmaLinux/CentOS

```
dnf install bzip2
```

Ubuntu/Debian

```
sudo install bzip2
```

OpenSUSE

```
zypper install bzip2
```

After you install the package, try to back up your database using the format below:

```
mariadb-dump -u username -p db_name | bzip2 > database.sql.bz2
```

If you want to back up your database, for example nodes database, run the below command:

```
mariadb-dump -u root -p nodes | bzip2 > nodes_backup_db.sql.bz2
```

Below is a comparison image of the size of the backup file that does not use compression, uses .tar.gz compression, and .bz2 compression:

```
sysadmin@ubuntu2404:~$ ls -lhtr
total 1.8G
-rw-rw-r-- 1 sysadmin sysadmin 1.3G Sep  9 12:00 backup_all_databases.sql
-rw-rw-r-- 1 sysadmin sysadmin 292M Sep  9 12:04 backup_all_databases.sql.gz
-rw-rw-r-- 1 sysadmin sysadmin 228M Sep  9 12:07 backup_all_databases.sql.bz2
sysadmin@ubuntu2404:~$
```

Comparison size file

Warning

If you use compression when backing up the database, it will take longer than if you don't use compression. And if you use bz2 compression, then the time spent will be longer than using .tar.gz.

Note

You should know that the mariadb-dump command has many useful options, including the **--single-transaction** and **--lock-tables** options. If you have a database that uses the **InnoDB storage engine**, then you should use the **--single-transaction** option because this option starts a transaction before dumping and reads data from a consistent snapshot without locking the tables for extended periods, allowing concurrent reads and writes. However, if you are using the **MyISAM** storage engine, you can use the **--lock-tables** option when backing up the database in MariaDB. If your database has a different storage engine, it is recommended to back up the database partially; some tables that use InnoDB use the **--single-transaction** option, and some tables that use MyISAM use the **--lock-tables** option when backing up the database.

References

mariadb.com
tecmint.com
linode.com
serversforhackers.com

How to Display the Storage Engine in MariaDB?

written by sysadmin | 12 November 2025

If you are a user of the MariaDB database, you should know about the term storage engine.

Problem

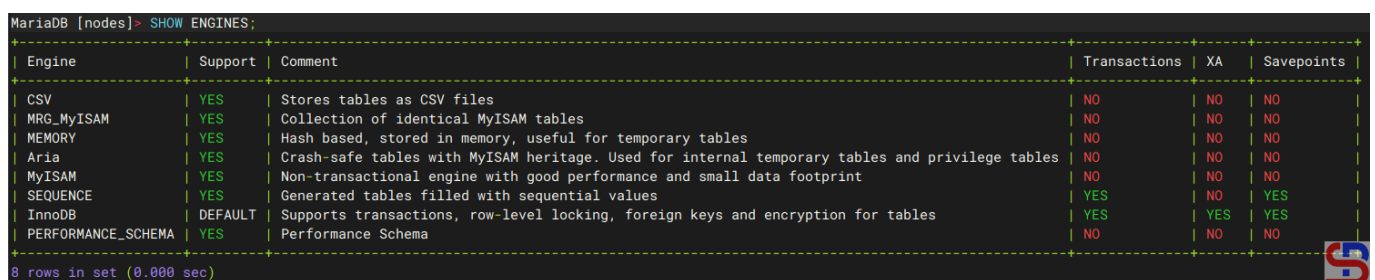
How to display the storage engine in MariaDB?

Solution

A database engine (or storage engine) is the underlying software component that a database management system (DBMS) uses to create, read, update, and delete (CRUD) data from a database. To see a list of all available storage engines on the server, use the command below in the MariaDB prompt:

```
SHOW ENGINES;
```

and it will appear as shown in the image below:



```
MariaDB [nodes]> SHOW ENGINES;
```

Engine	Support	Comment	Transactions	XA	Savepoints
CSV	YES	Stores tables as CSV files	NO	NO	NO
MRG_MyISAM	YES	Collection of identical MyISAM tables	NO	NO	NO
MEMORY	YES	Hash based, stored in memory, useful for temporary tables	NO	NO	NO
Aria	YES	Crash-safe tables with MyISAM heritage. Used for internal temporary tables and privilege tables	NO	NO	NO
MyISAM	YES	Non-transactional engine with good performance and small data footprint	NO	NO	NO
SEQUENCE	YES	Generated tables filled with sequential values	YES	NO	YES
InnoDB	DEFAULT	Supports transactions, row-level locking, foreign keys and encryption for tables	YES	YES	YES
PERFORMANCE_SCHEMA	YES	Performance Schema	NO	NO	NO

8 rows in set (0.000 sec)

Display all the storage engines in MariaDB

From the image above, you can see that InnoDB is the default storage engine in MariaDB. You can also use the command below to see the default storage engine in MariaDB:

```
SHOW GLOBAL VARIABLES LIKE 'default_storage_engine';
```

```
MariaDB [nodes]> SHOW GLOBAL VARIABLES LIKE 'default_storage_engine';
```

Variable_name	Value
default_storage_engine	InnoDB

```
1 row in set (0.006 sec)
```



Display the default storage engine in MariaDB

You can change the default storage engine by using the command below, which changes the default storage engine to MyISAM:

```
SET GLOBAL default_storage_engine='MyISAM';
```

Or you can use the below command if the session default storage engine supersedes the global default during this session:

```
SET SESSION default_storage_engine='MyISAM';
```

You should know that the storage engine is used per table and not per database, and to see it, you can use the command below:

```
SELECT TABLE_NAME, ENGINE
FROM information_schema.TABLES
WHERE TABLE_SCHEMA = 'nodes';
```

and the result will look like below:

```
MariaDB [nodes]> SELECT TABLE_NAME, ENGINE  
-> FROM information_schema.TABLES  
-> WHERE TABLE_SCHEMA = 'nodes';
```

```
+-----+-----+  
| TABLE_NAME          | ENGINE |  
+-----+-----+  
| flaggedrevs          | InnoDB |  
| flaggedrevs_statistics | InnoDB |  
| category             | InnoDB |  
| babel                | InnoDB |  
+-----+-----+  
4 rows in set (0.000 sec)
```



Display the storage engine used in the table

From the image above, it can be seen that all tables in the nodes database use InnoDB, so it can be said that the nodes database uses InnoDB. However, one database (schema) can have a table with a different engine, and the query below is the sample to create 2 tables that use different storage engines:

```
CREATE TABLE product (  
    id INT PRIMARY KEY,  
    name VARCHAR(100)  
) ENGINE=InnoDB;
```

```
CREATE TABLE access_log (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    time DATETIME,  
    user VARCHAR(50)  
) ENGINE=MyISAM;
```

where the product table uses InnoDB and the table access_log uses MyISAM. If a database uses various storage engines and you want to know how many storage engines are used, you can use the query below:

```
SELECT ENGINE, COUNT(*) AS Number_of_tables  
FROM information_schema.TABLES  
WHERE TABLE_SCHEMA = 'db_name'  
GROUP BY ENGINE;
```

You can change the storage engine of a table by using the query below, for example, change it to MyISAM:

```
ALTER TABLE database_name.table_name ENGINE=MyISAM;
```

```
MariaDB [nodes]> SELECT TABLE_NAME, ENGINE
-> FROM information_schema.TABLES
-> WHERE TABLE_SCHEMA = 'nodes';
+-----+-----+
| TABLE_NAME | ENGINE |
+-----+-----+
| flaggedrevs | InnoDB |
| flaggedrevs_statistics | InnoDB |
| category | InnoDB |
| babel | InnoDB |
+-----+-----+
4 rows in set (0.000 sec)
```

```
MariaDB [nodes]> ALTER TABLE nodes.babel ENGINE=MyISAM;
Query OK, 19570 rows affected (0.076 sec)
Records: 19570 Duplicates: 0 Warnings: 0
```

```
MariaDB [nodes]> SELECT TABLE_NAME, ENGINE FROM information_schema.TABLES WHERE TABLE_SCHEMA = 'nodes';
+-----+-----+
| TABLE_NAME | ENGINE |
+-----+-----+
| flaggedrevs | InnoDB |
| flaggedrevs_statistics | InnoDB |
| category | InnoDB |
| babel | MyISAM |
+-----+-----+
4 rows in set (0.000 sec)
```



Change the storage engine in the table

Note

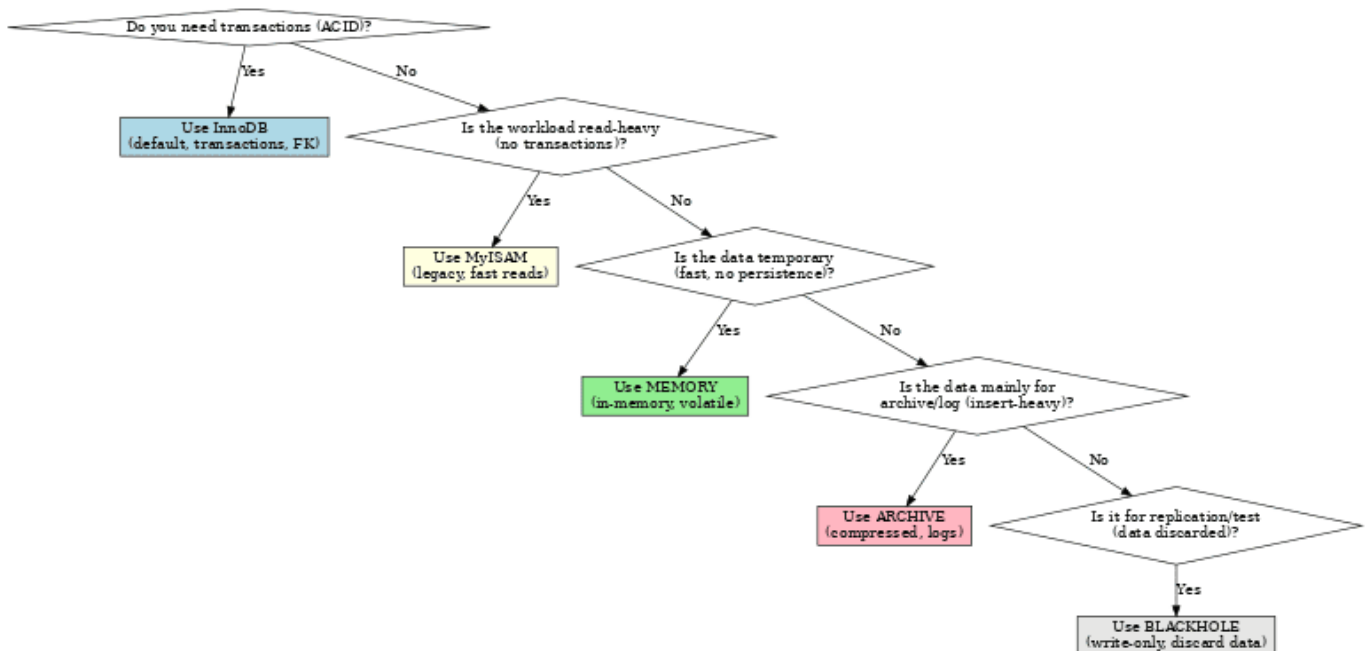
As explained above, there are many storage engines that you can use. But before you decide to use a storage engine, you should first find out the advantages and disadvantages of each storage engine, and to briefly see the differences between storage engines in the image below:

COMPARISON OF STORAGE ENGINES IN MARIADB/MYSQL

Feature / Engine	InnoDB (default)	MyISAM	MEMORY	ARCHIVE	BLACKHOLE
Transactions (ACID)	✓	✗	✗	✗	✗
Foreign Keys	✓	✗	✗	✗	✗
Locking	Row-level	Table-level	Table-level	Table-level	N/A
Crash recovery	Yes (redo /undo logs)	Limited (can lose data)	Data lost (in RAM)	Yes (insert only, compressed)	Accepts writes but discards data
Read/Write speed	Balanced	Very fast on reads, slower writes	Extremely fast (in-memory)	Optimized for inserts only	None
Storage usage	General purpose apps, OLTP, most modern apps	Read-footprint workloads, legacy apps	RAM only	Very small (compressed)	Replication (pass-through), testing

Comparison between the storage engines

And below is a flowchart to determine the storage engines you will use:



Flowchart to choose the storage engine

However, as far as I know, among the many choices of storage engines, generally people use 2 types of storage engines, namely InnoDB and MyISAM. Maybe you can see the difference between the two through [this site](#).

References

en.wikipedia.org
slideshare.net
medium.com
support.sages.com
mariadb.com
dev.mysql.com
managedserver.eu

[How to Display All Databases Created by the User in MariaDB?](#)

written by sysadmin | 12 November 2025

By default, in MariaDB, 4 databases are automatically created by the database system, namely **information_schema**, **mysql**, **performance_schema**, and **sys**. But I just want to display a database created by the user and not the default system databases.

Problem

How to display all databases created by the user in MariaDB?

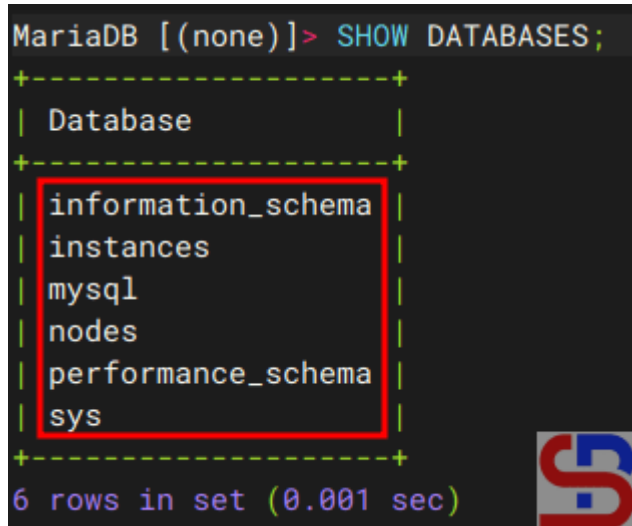
Solution

If you use the **show databases** command after you access successfully accessing MariaDB, it will display all databases created by the user and default system databases, as shown below:

```

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| instances |
| mysql |
| nodes |
| performance_schema |
| sys |
+-----+
6 rows in set (0.001 sec)

```



Display all databases

If you want to list only User-Created Databases, you can use a query as below:

```

SELECT SCHEMA_NAME
FROM information_schema.SCHEMATA
WHERE SCHEMA_NAME NOT IN
('information_schema','mysql','performance_schema','sys');

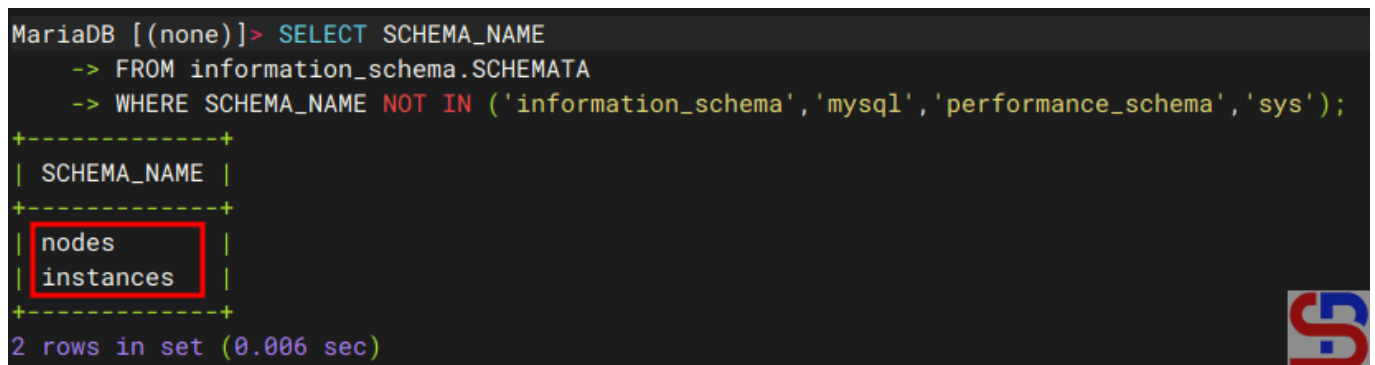
```

and it will display like the image below:

```

MariaDB [(none)]> SELECT SCHEMA_NAME
-> FROM information_schema.SCHEMATA
-> WHERE SCHEMA_NAME NOT IN ('information_schema','mysql','performance_schema','sys');
+-----+
| SCHEMA_NAME |
+-----+
| nodes |
| instances |
+-----+
2 rows in set (0.006 sec)

```



Display only User-Created Databases

Note

If you want to run the Linux command to display all databases created by the user only, use the command below:

```

mysql -u root -p -e "SELECT SCHEMA_NAME FROM information_schema.SCHEMATA
WHERE SCHEMA_NAME NOT IN
('information_schema','mysql','performance_schema','sys');"

```

And you will see the result like the image below:

```
sysadmin@ubuntu2404:~$ mariadb -u root -p -e "SELECT SCHEMA_NAME FROM information_schema.SCHEMATA WHERE SCHEMA_NAME NOT IN ('information_schema','mysql','performance_schema','sys');"
Enter password:
+-----+
| SCHEMA_NAME |
+-----+
| nodes       |
| instances   |
+-----+
```

Display only User-Created Databases using the Linux command

References

stackoverflow.com
kinsta.com

How to Display the Total Size of the Entire MariaDB Database in the CLI?

written by sysadmin | 12 November 2025

I want to know the total size of the entire MariaDB database.

Problem

How to display the total size of the entire MariaDB database in the CLI?

Solution

There are two methods for displaying the total size of the entire MariaDB database in the CLI:

1. Using a query

If you have entered the MariaDB database, you can use the command below to display the total size of a MariaDB database:


```
SELECT table_schema AS "Database",
ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS "Size (MB)"
FROM information_schema.TABLES
GROUP BY table_schema;
```

The above query will display the total size of the entire MariaDB database in MegaBytes (MB) as shown in the image below:

```
MariaDB [(none)]> SELECT table_schema AS "Database",
-> ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS "Size (MB)"
-> FROM information_schema.TABLES
-> GROUP BY table_schema;
```

Database	Size (MB)
information_schema	0.20
instances	488.72
mysql	3.33
nodes	540.02
performance_schema	0.00
sys	0.03

```
6 rows in set (0.013 sec)
```

Display the total size of the databases using the query in Megabytes

If you want to display in Gigabytes (GB), use the command below:

```
SELECT table_schema AS "Database",
ROUND(SUM(data_length + index_length) / 1024 / 1024 / 1024, 2) AS "Size (GB)"
FROM information_schema.TABLES
GROUP BY table_schema;
```

so that it will display as shown in the image below:

```

MariaDB [(none)]> SELECT table_schema AS "Database",
-> ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS "Size (GB)"
-> FROM information_schema.TABLES
-> GROUP BY table_schema;
+-----+-----+
| Database          | Size (GB) |
+-----+-----+
| information_schema | 0.00      |
| instances          | 0.48      |
| mysql             | 0.00      |
| nodes             | 0.53      |
| performance_schema | 0.00      |
| sys               | 0.00      |
+-----+-----+
6 rows in set (0.012 sec)

```



Display the total size of the databases using the query in GigaBytes

2. Using Linux Command

If you want to display the total size of the entire MariaDB database using Linux commands, use the command below:

```

mariadb -u root -p -e "
SELECT table_schema AS 'Database',
ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS 'Size (MB)'
FROM information_schema.TABLES
GROUP BY table_schema;"

```

If you run the above command, it will display like the image below:

```

root@ubuntu2404:~# mariadb -u root -p -e "
SELECT table_schema AS 'Database',
       ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS 'Size (MB)'
FROM information_schema.TABLES
GROUP BY table_schema;"
Enter password:
+-----+-----+
| Database          | Size (MB) |
+-----+-----+
| information_schema | 0.20      |
| instances          | 488.72    |
| mysql             | 3.33      |
| nodes             | 540.02    |
| performance_schema | 0.00      |
| sys               | 0.03      |
+-----+-----+
root@ubuntu2404:~# █

```

Display the total size of the databases using the Linux command in Megabytes

If you want to display it in GigaBytes, use the command below:

```

mariadb -u root -p -e "
SELECT table_schema AS 'Database',
       ROUND(SUM(data_length + index_length) / 1024 / 1024 / 1024, 2) AS 'Size (GB)'
FROM information_schema.TABLES
GROUP BY table_schema;"

```

and it will display like the image below:

```

root@ubuntu2404:~# mariadb -u root -p -e "
SELECT table_schema AS 'Database',
       ROUND(SUM(data_length + index_length) / 1024 / 1024 / 1024, 2) AS 'Size (GB)'
FROM information_schema.TABLES
GROUP BY table_schema;"
Enter password:
+-----+-----+
| Database          | Size (GB) |
+-----+-----+
| information_schema | 0.00      |
| instances          | 0.48      |
| mysql             | 0.00      |
| nodes             | 0.53      |
| performance_schema | 0.00      |
| sys               | 0.00      |
+-----+-----+
root@ubuntu2404:~# █

```

Display the total size of the databases using the Linux command in Gigabytes

Or in a short time, you can use the Linux command below to see the total size of the entire database in MariaDB:

```
cd /var/lib/mysql/  
du -sh *
```

```
root@ubuntu2404:~# cd /var/lib/mysql/  
root@ubuntu2404:/var/lib/mysql#  
root@ubuntu2404:/var/lib/mysql# du -sh *  
412K    aria_log.00000001  
4.0K    aria_log_control  
16K     ddl_recovery.log  
0       debian-10.11.flag  
4.0K    ib_buffer_pool  
76M     ibdata1  
96M     ib_logfile0  
12M     ibtmp1  
543M    instances  
0       multi-master.info  
3.7M    mysql  
4.0K    mysql_upgrade_info  
610M    nodes  
8.0K    performance_schema  
600K    sys  
root@ubuntu2404:/var/lib/mysql#
```

Display the total size of the databases using the Linux command

Note

If you want to display the total size of the entire MariaDB database in KiloBytes, then in the **ROUND(SUM(data_length + index_length))** section, simply divide it by 1024 so that it becomes as follows:

```
mariadb -u root -p -e "  
SELECT table_schema AS "Database",  
ROUND(SUM(data_length + index_length) / 1024 , 2) AS "Size (KB)"  
FROM information_schema. TABLES  
GROUP BY table_schema;"
```

And if you want to display in TeraBytes size, then in the **ROUND(SUM(data_length + index_length)** section, divide by 1024 4 times, so that the command is as follows:

```
mariadb -u root -p -e "  
SELECT table_schema AS 'Database',  
ROUND(SUM(data_length + index_length) / 1024 / 1024 / 1024 / 1024 , 2) AS  
'Size (KB)'  
FROM information_schema.TABLES  
GROUP BY table_schema;"
```

References

[a2hosting.com](https://a2hosting.com/database/guide)
database.guide
tecmin.com

How to Reset MariaDB Root Password?

written by sysadmin | 12 November 2025

I want to access the MariaDB database using the root user, but I forgot the password of the root user.

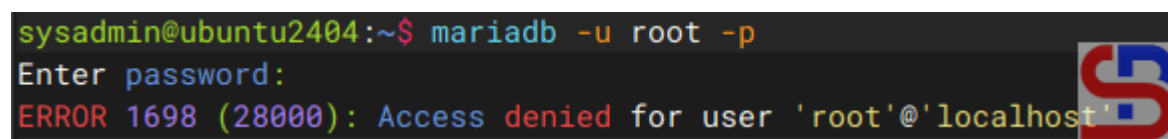
Problem

How to reset MariaDB root password?

Solution

When I insert my password root to access MariaDB, here is the error:

```
sysadmin@ubuntu2404:~$ mariadb -u root -p  
Enter password:  
ERROR 1698 (28000): Access denied for user 'root'@'localhost'
```



Error when accessing MariaDB

Here are the steps to reset the root password in MariaDB:

1. Stop the service

Use the command below to stop MariaDB's service:

```
sudo systemctl stop mariadb
```

2. Running Mariadb in Safe Mode

Run the command below to run the MariaDB service without access rights and networks:

```
sudo mysqld_safe --skip-grant-tables --skip-networking &
```

3. Change the password

Enter MariaDB using the command:

```
sudo mariadb -uroot -p
```

Press the **Enter** button and after that, type the commands below, and I use the **qwerty** password as my new password:

```
FLUSH PRIVILEGES;  
ALTER USER root@localhost IDENTIFIED BY 'qwerty';  
FLUSH PRIVILEGES;  
exit
```

4. Stop MariaDB

Use the command below to stop MariaDB's service:

```
sudo mysqladmin -u root -p shutdown
```

Enter the password you just created if you are asked to enter a password.

5. Running the Service

Run the command below to turn on MariaDB's service:

```
sudo systemctl start mariadb
```

6. Try a new password

Enter MariaDB by using the password that you created previously using the command below:

```
sudo mariadb -u root -p
```

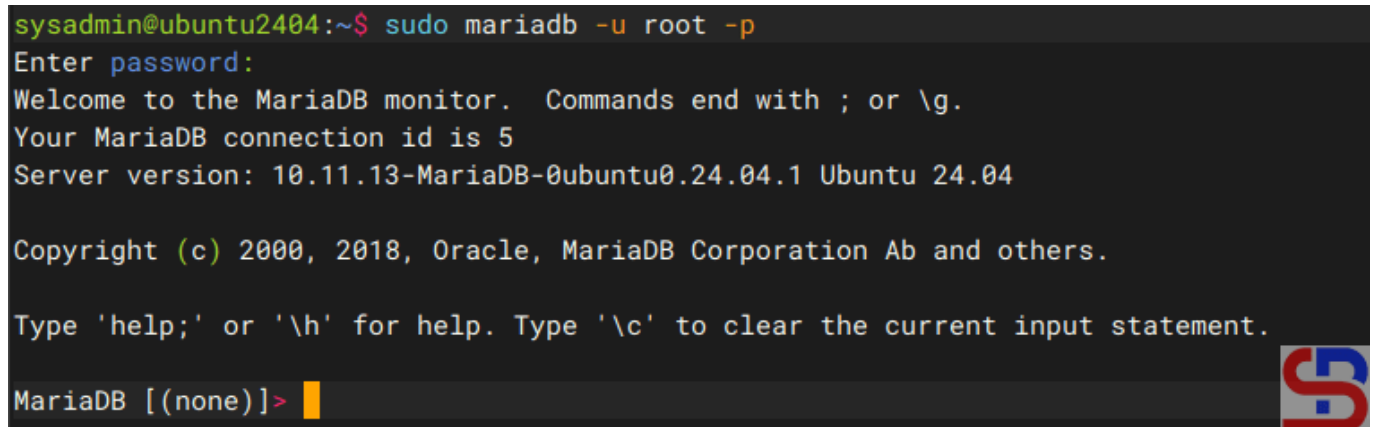
Insert your new password, and if you type the correct password, then you should be able to enter Mariadb as in the image below:

```
sysadmin@ubuntu2404:~$ sudo mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.11.13-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> 
```



Test your new password to access MariaDB

Note

Maybe as an alternative to storing passwords for the root user in MariaDB, you can store the password in a file, but using [an encrypted Vim editor](#).

References

musaamin.web.id
serverspace.io
vexxhost.com

stackoverflow.com