

# MySQL: Getting started

*How to start using MySQL*

## Welcome on Stackhero's documentation!

Stackhero provides MySQL instances that are ready for production in just 2 minutes!

Including TLS encryption (aka HTTPS), customizable domain name, PhpMyAdmin, backups and updates in just a click.

Try our [managed MySQL cloud](#) in just 2 minutes

- [How to connect from PHP to MySQL](#)
  - [Connect PHP to MySQL with object-oriented style](#)
  - [Connect PHP to MySQL with procedural style](#)
  - [Connect PHP to MySQL with using credentials from environment variables](#)
- [Connect from Laravel to MySQL](#)
- [How to connect from Node.js to MySQL](#)
- [Connect from Node.js/NestJS/TypeORM to MySQL](#)
- [Create a user on MySQL using PhpMyAdmin](#)

## How to connect from PHP to MySQL

In these examples, we will use the "MySQL Improved Extension" known as `MySQLi`.

### Connect PHP to MySQL with object-oriented style

```
<?php

$hostname = '<STACKHERO_MYSQL_HOST>';
$user = 'root';
$password = '<STACKHERO_MYSQL_ROOT_PASSWORD>';
$database = 'mysql'; // You shouldn't use the "mysql" database. This is just for
the example. The recommended way is to create a dedicated database (and user) in
PhpMyAdmin and use it then here.

$mysqli = mysqli_init();
$mysqliConnected = $mysqli->real_connect($hostname, $user, $password, $database,
NULL, NULL, MYSQLI_CLIENT_SSL);
if (!$mysqliConnected) {
    die("Connect Error: " . $mysqli->connect_error());
}
```

```
echo 'Success... ' . $mysqli->host_info . "\n";

$mysqli->close();

?>
```

## Connect PHP to MySQL with procedural style

```
<?php

$hostname = '<STACKHERO_MYSQL_HOST>';
$user = 'root';
$password = '<STACKHERO_MYSQL_ROOT_PASSWORD>';
$database = 'mysql'; // You shouldn't use the "mysql" database. This is just for
the example. The recommended way is to create a dedicated database (and user) in
PhpMyAdmin and use it then here.

$mysqli = mysqli_init();
$mysqliConnected = mysqli_real_connect($mysqli, $hostname, $user, $password,
$database, NULL, NULL, MYSQLI_CLIENT_SSL);
if (!$mysqliConnected) {
    die("Connect Error: " . mysqli_connect_error($mysqli));
}

echo 'Success... ' . mysqli_get_host_info($mysqli) . "\n";

mysqli_close($mysqli);

?>
```

## Connect PHP to MySQL with using credentials from environment variables

We recommend to not keep your credentials in your source code but use environment variables in place.

Here is how to retrieve these credentials:

```
$hostname = parse_url(getenv('STACKHERO_MYSQL_HOST'));
$user = parse_url(getenv('STACKHERO_MYSQL_USER'));
$password = parse_url(getenv('STACKHERO_MYSQL_PASSWORD'));
$database = parse_url(getenv('STACKHERO_MYSQL_USER'));
```

## Connect from Laravel to MySQL

Edit the file `config/database.php` and replace the `mysql` part with the following configuration:

```
'mysql' => [
  'driver' => 'mysql',
  'host' => env('STACKHERO_MYSQL_HOST'),
  'port' => 3306,
  'username' => env('STACKHERO_MYSQL_USER'),
  'password' => env('STACKHERO_MYSQL_PASSWORD'),
  'database' => env('STACKHERO_MYSQL_USER'),
  'charset' => 'utf8mb4',
  'collation' => 'utf8mb4_unicode_ci',
  'prefix' => '',
  'prefix_indexes' => true,
  'strict' => true,
  'engine' => null,
  'sslmode' => 'require',
  'options' => extension_loaded('pdo_mysql') ? array_filter([
    PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
  ]) : [],
],
```

## How to connect from Node.js to MySQL

In this example we will use the official `xdevapi` package that will use the new MySQL X protocol. To install it simply run this command: `npm install --save @mysql/xdevapi`

```
const mysqlx = require('@mysql/xdevapi');

(async () => {
  // Connection to MySQL using MySQL X Protocol
  const session = await mysqlx.getSession({
    host: '<STACKHERO_MYSQL_HOST>',
    user: 'root',
    password: '<STACKHERO_MYSQL_ROOT_PASSWORD>'
  });

  // Create a schema (database) if not exists
  const schemaExists = await
  session.getSchema('stackherotest').existsInDatabase();
  if (!schemaExists) {
    await session.createSchema('stackherotest');
  }

  // Create table "users" if not exists
  const tableExists = await session
```

```

.getSchema('stackherotest')
.getTable('users')
.existsInDatabase();
if (!tableExists) {
  await session
    .sql('CREATE TABLE `stackherotest`.`users` '
      + '('
      + '`userId` INT UNSIGNED NOT NULL,'
      + '`name` VARCHAR(128) NOT NULL,'
      + '`address` TEXT NOT NULL,'
      + '`email` VARCHAR(265) NOT NULL'
      + ') '
      + 'ENGINE = InnoDB;')
    .execute();
}

// Insert a fake user
await session
  .getSchema('stackherotest') // Database name
  .getTable('users') // Table name
  .insert('userId', 'name', 'address', 'email') // Columns names
  .values(
    Math.round(Math.random() * 100000), // Generate a fake userId
    'User name', // column 'name'
    'User address', // column 'address'
    'user@email.com' // column 'email'
  )
  .execute();

// Count number of rows in table users
const usersCount = await session
  .getSchema('stackherotest') // Database name
  .getTable('users')
  .count();

console.log(`There is now ${usersCount} entries in table "users"`);

// Close the connection to MySQL
await session.close();

})().catch(error => {
  console.error('');
  console.error('🚨 An error occurred!');
  console.error(error);
  process.exit(1);
});

```

# Connect from Node.js/NestJS/TypeORM to MySQL

To connect from Node.js/NestJS/TypeORM, you have to add the `ssl` option like in this example:

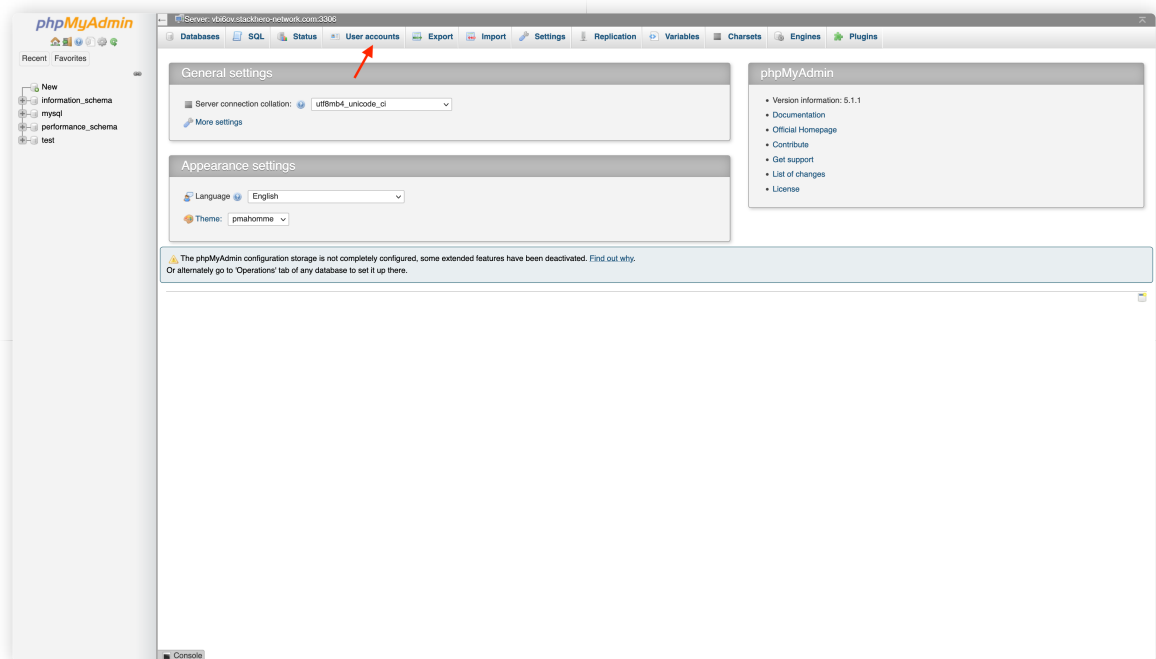
```
TypeOrmModule.forRoot({
  type: 'mysql',
  host: '<STACKHERO_MYSQL_HOST>',
  port: 3306,
  username: 'root',
  password: '<STACKHERO_MYSQL_ROOT_PASSWORD>',
  database: 'root',
  entities: [],
  synchronize: true,
  ssl: {}
})
```

## Create a user on MySQL using PhpMyAdmin

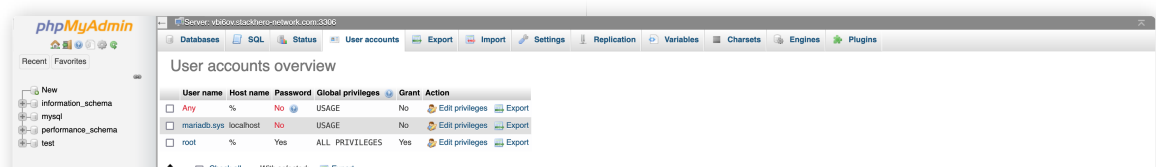
A best practice is to create a user other than the default "root" for your application.

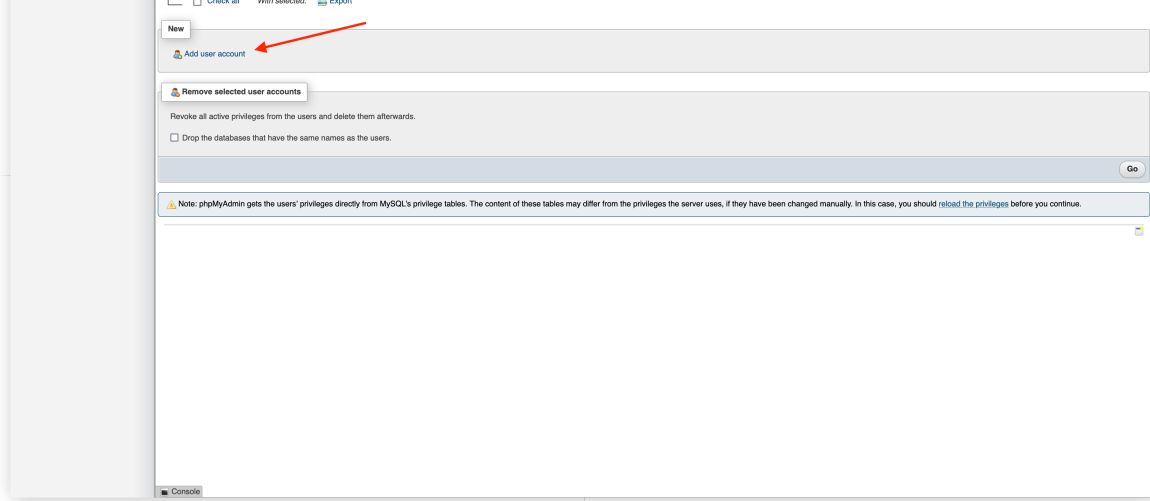
The easiest way to do that is to use PhpMyAdmin.

1. In PhpMyAdmin, click on `User accounts` on top.



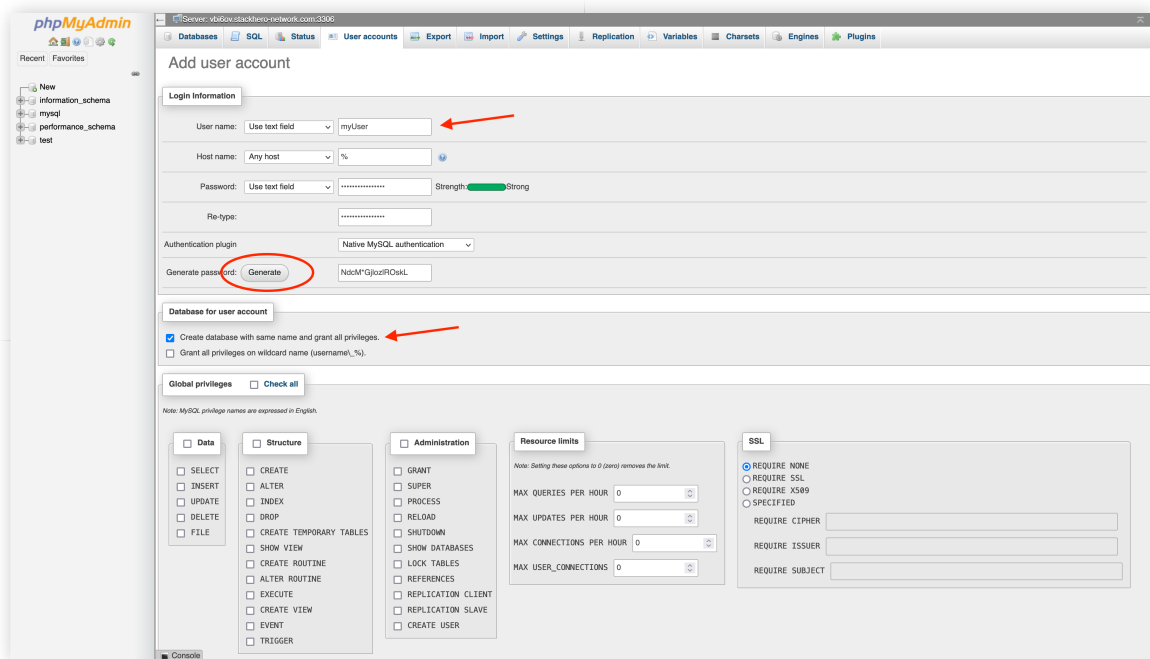
2. Click on `Add user account`.





3. Fill the user creation form:

- Define an account name (generally your application name)
- Click on "Generate password" to get a secured password (copy it to your clipboard)
- Check the "Create database with same name and grant all privileges"

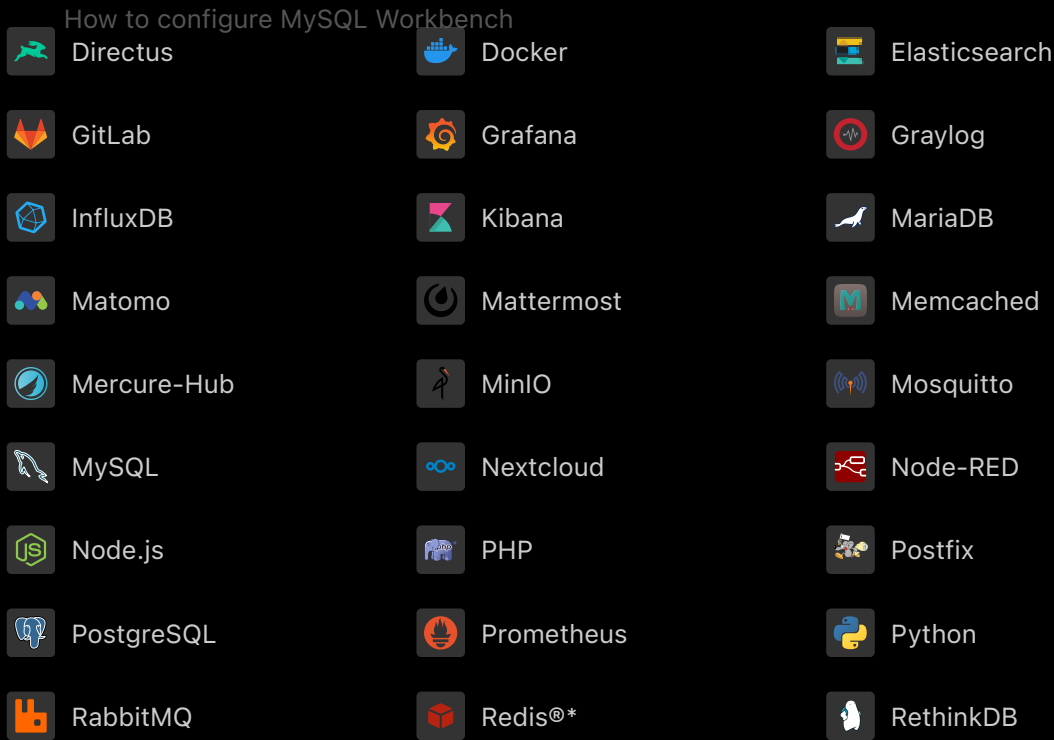


Once validated, the user will be created as its database which will get the same name as the username.


## Other articles about MySQL that might interest you

- [Introduction](#)  
Introduction to MySQL
- [Advanced usage](#)  
How to configure, optimize, export or import your MySQL data
- [Troubleshooting](#)  
MySQL common issues and solutions
- [Workbench](#)

## Our Managed Services



- [Terms of Service](#)
- [Privacy Policy](#)
- [Documentations](#)
- [Support](#)
- [Status](#)

 [English](#)

 [Global](#)



Directus, Docker, Elasticsearch, GitLab, Grafana, Graylog, InfluxDB, Kibana, MariaDB, Matomo, Mattermost, Memcached, Mercure-Hub, MinIO, MongoDB, Mosquitto, MySQL, Nextcloud, Node-RED, Node.js, PHP, Postfix, PostgreSQL, Prometheus, Python, RabbitMQ, Redis®, RethinkDB are trademarks and property of their respective owners. All product and service names used on this website are for identification purposes of their open sourced products only and do not imply endorsement. Stackhero is not affiliated to these trademarks or companies.

\*Redis is a registered trademark of Redis Ltd. Any rights therein are reserved to Redis Ltd. Any use by Stackhero is for referential purposes only and does not indicate any sponsorship, endorsement or affiliation between Redis and Stackhero

Some icons of this website are made by Dimitry Miroljubov.

© Stackhero. All rights reserved.