

# Installation and Configuration of phpMyAdmin in an Apache2 LXC (MariaDB in Separate Container)

---

## Introduction to phpMyAdmin

phpMyAdmin is a **web-based administration interface** for MySQL and MariaDB databases. It allows users to easily create, manage, and edit databases via a browser without having to use the command line directly.

Typical uses of phpMyAdmin include:

- **Database Administration:** Creating, deleting, and editing databases and tables.
- **Data Management:** Inserting, editing, or deleting data records.
- **SQL Queries:** Executing SQL commands and queries directly through the user interface.
- **Backup and Restore:** Exporting and importing databases for backups or migrations.
- **User and Privilege Management:** Creating database users and assigning permissions.

phpMyAdmin is primarily used to simplify database management for administrators and developers and is especially common in web server environments.

## Prerequisites

- Apache2 is installed and running in the container.
- MariaDB is running in a separate LXC container (`mariadb`) and is configured to accept **external connections** (e.g., `bind-address = 192.168.137.120`) and has an admin user.
- The containers are on the same network.

---

## 1. Install phpMyAdmin in the Web Server Container

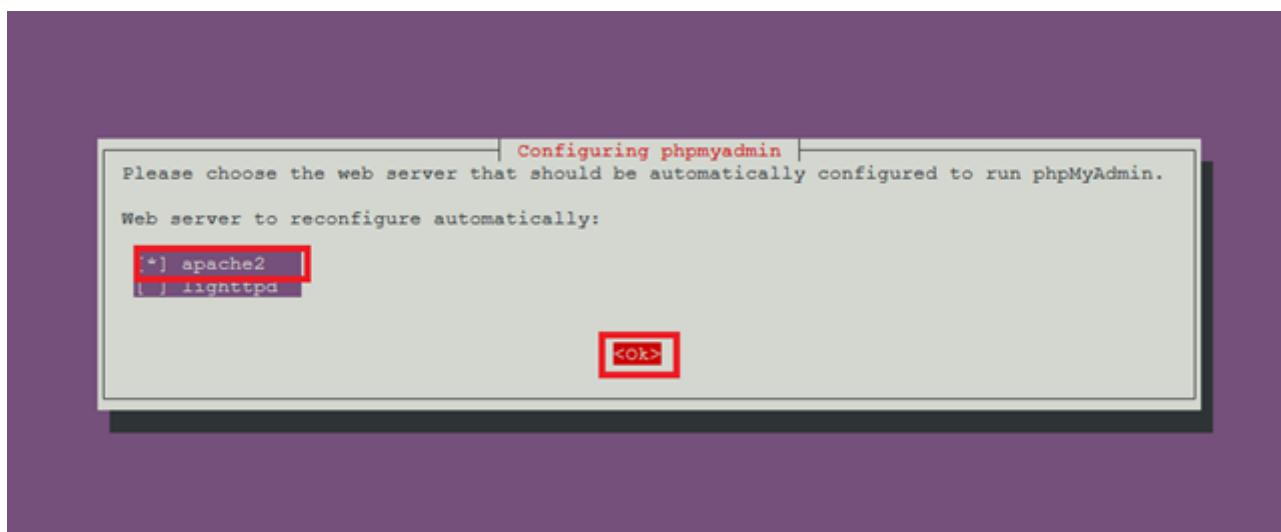
 Note: "phpMyAdmin" is a PHP web application that runs on an existing web server (e.g., Apache or Nginx). Therefore, phpMyAdmin cannot be started "alone" in a container—it always requires a web server and an existing database connection to function.

```
apt update  
apt install -y phpmyadmin php-mysql
```

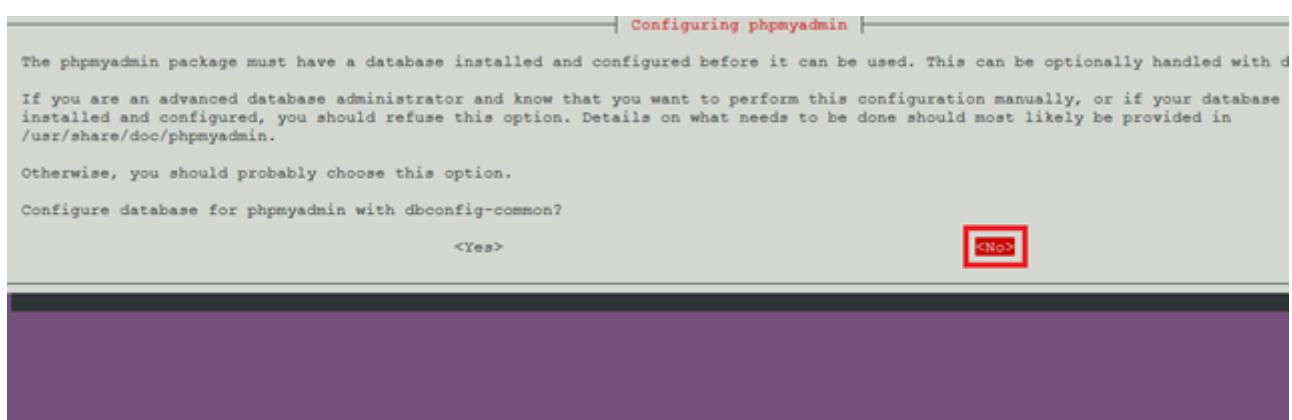
```
pdal@apache101:~$ sudo apt update
[sudo] password for pdal:
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1170 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-updates/main Translation-en [246 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [1279 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [272 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1092 kB]
Fetched 4311 kB in 3s (1515 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
pdal@apache101:~$ sudo apt install phpmyadmin php-mysql
```

💡 During Installation:

Web server selection: If prompted, select **apache2** (Spacebar → Tab → OK).



Database configuration with dbconfig-common: → Select **No**, as MariaDB is running externally.



If the web server selection doesn't appear, manually link phpMyAdmin:

```
ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
```

Then restart Apache:

```
systemctl restart apache2
```

## ⚙️ 2. Configure Remote MariaDB Connection in phpMyAdmin

Edit the configuration file:

```
sudo nano /etc/phpmyadmin/config.inc.php
```

```
<?php
/**
 * Debian local configuration file
 *
 * This file overrides the settings made by phpMyAdmin interactive setup
 * utility.
 *
 * For example configuration see
 *   /usr/share/doc/phpmyadmin/examples/config.sample.inc.php
 * or
 *   /usr/share/doc/phpmyadmin/examples/config.manyhosts.inc.php
 *
 * NOTE: do not add security sensitive data to this file (like passwords)
 * unless you really know what you're doing. If you do, any user that can
 * run PHP or CGI on your webserver will be able to read them. If you still
 * want to do this, make sure to properly secure the access to this file
 * (also on the filesystem level).
 */

if (!function_exists('check_file_access')) {
    function check_file_access(string $path): bool
    {
        if (is_readable($path)) {
            return true;
        }

        if (!file_exists($path)) {
            return false;
        }

        error_log(
            "File $path does not exist or is not readable"
        );
    }
}
```

unterhalb der Zeile:

Add the following below the line:

```
if (empty($dbserver)) $dbserver = 'localhost';
```

Add:

```
$cfg['Servers'][$i]['host'] = '192.168.137.120'; // IP address of the MariaDB
container
```

```
$cfg['Servers'][$i]['connect_type'] = 'tcp';
```

```
/* Configure according to dbconfig-common if enabled */
if (!empty($dbname)) {
    /* Authentication type */
    $cfg['Servers'][$i]['auth_type'] = 'cookie';
    /* Server parameters */
    if (empty($dbserver)) $dbserver = 'localhost';
    $cfg['Servers'][$i]['host'] = '192.168.137.120'; // IP-Adress
    $cfg['Servers'][$i]['connect_type'] = 'tcp';

    if (!empty($dbport) || $dbserver != 'localhost') {
        $cfg['Servers'][$i]['connect_type'] = 'tcp';
        $cfg['Servers'][$i]['port'] = $dbport;
    }
    // $cfg['Servers'][$i]['compress'] = false;
```

❖ **Adjust the IP address** to the actual address of the MariaDB container.

Save and close **Ctrl + O -> Enter -> Ctrl + X**.

---

### ⌚ 3. Restart Apache

```
systemctl restart apache2
```

---

### 🌐 4. Access phpMyAdmin in the Browser

Open in the browser:

<http://IP-of-phpMyAdmin-Container/phpmyadmin>

Example:

<http://192.168.137.101/phpmyadmin>

Log in with the MySQL/MariaDB user who has access from the phpMyAdmin container.



## 📝 5. Troubleshooting

Note: This error should only occur if you have not completely executed the script "Installation and Configuration of MariaDB in the LXC Container." If you have created the user correctly, check connectivity to the database LXC.

### ✗ Access denied

Switch to the MariaDB container and ensure that the user in MariaDB is correctly enabled for our local network:

```
CREATE USER 'pdal'@'192.168.137.120' IDENTIFIED BY 'JadeHS20';
GRANT ALL PRIVILEGES ON *.* TO 'pdal'@'192.168.137.120' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

### ☑ Conclusion

phpMyAdmin is now ready and connects to the external MariaDB database. You can manage databases, create users, make backups, etc., via the web interface.

Currently, a notice is displayed in the phpMyAdmin WebGUI indicating that the configuration storage is not fully configured. This will be explained in the following steps.

The screenshot shows the phpMyAdmin configuration interface. On the left, there's a sidebar with 'phpMyAdmin' branding and a tree view of databases: 'information\_schema', 'mysql', 'performance\_schema', and 'test'. The main content area has several tabs: 'Datenbanken', 'SQL', 'Status', 'Benutzerkonten', 'Exportieren', 'Importieren', 'Einstellungen' (which is selected), 'Replication', 'Variablen', 'Zeichensätze', 'Formate', and 'Erweiterungen'. The 'Einstellungen' tab contains sections for 'Allgemeine Einstellungen' (with 'Passwort ändern', 'Zeichensatz/Kollation der Verbindung zum Server' set to 'utf8mb4\_unicode\_ci', and 'Weitere Einstellungen'), 'Anzeige-Einstellungen' (with 'Sprache (Language)' set to 'Deutsch - German'), and 'Webserver' (listing Apache/2.4.58 (Ubuntu), MariaDB-Client Version: libmysql - mysqld 8.3.6, PHP-Erweiterung myopgi, and PHP-Version 8.3.6). A large red warning box at the bottom states: 'Der phpMyAdmin-Konfigurationsspeicher ist nicht vollständig konfiguriert, einige erweiterte Funktionen wurden deaktiviert. [Finden Sie heraus warum.](#) Oder wechseln Sie in einer beliebigen Datenbank zum Tab „Operationen“, um die Einstellung dort vorzunehmen.' Below the warning box is a green bar with 'Konsole' and a taskbar with various icons.

! Der phpMyAdmin-Konfigurationsspeicher ist nicht vollständig konfiguriert, einige erweiterte Funktionen wurden deaktiviert. [Finden Sie heraus warum.](#) Oder wechseln Sie in einer beliebigen Datenbank zum Tab „Operationen“, um die Einstellung dort vorzunehmen.

## 6. Setting up the phpMyAdmin Configuration Storage (Advanced Features)

### Goal

Set up the internal phpMyAdmin configuration storage to enable advanced features such as Designer, Bookmarks, and Relation display.

### Prerequisites

- **Apache Container (where phpMyAdmin runs):** CTID 101, Hostname: **apache101**
- **MariaDB Container (where the MySQL database runs):** e.g., CTID 120
- **Database access user exists** with all privileges
- Access data for phpMyAdmin configuration storage:
  - Username: **pdal**
  - Password: **JadeHS20**

### Step 6.1: Copy SQL file from Apache Container to Host

#### 💡 **Explanation:**

phpMyAdmin requires specific internal tables in the database to support features like Bookmarks, Relations, or PDF export. These tables are defined by the file **create\_tables.sql**. Since in our environment

**phpMyAdmin runs in the Apache container but the MariaDB runs in a separate container, phpMyAdmin cannot create the tables directly itself.**

We therefore use the safe and simple path via the Host (Proxmox):

```
pct pull 101 /usr/share/phpmyadmin/sql/create_tables.sql /tmp/create_tables.sql
```

```
Linux proxmox 6.8.12-11-pve #1 SMP PREEMPT_DYNAMIC PMX 6.8.12-11 (2025-05-22T09:39z) x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

```
Last login: Mon Jun 30 14:51:41 CEST 2025 on pts/0
root@proxmox:~# pct pull 101 /usr/share/phpmyadmin/sql/create_tables.sql /tmp/create_tables.sql
root@proxmox:~#
```

- The file is retrieved from CT 101 to the Host under [/tmp/create\\_tables.sql](#).
- It is now outside the containers on the Host and can be further used as needed.

## Step 6.2: Copy SQL file to the MariaDB Container

```
pct push 120 /tmp/create_tables.sql /tmp/create_tables.sql
```

```
root@proxmox:~# pct push 120 /tmp/create_tables.sql /tmp/create_tables.sql
```

- Pushes the file to CT 120 so that MariaDB can execute it.
- The tables can then be created using MySQL commands.

### 💡 Alternative Methods:

- Direct transfer from container to container over the network (e.g., via [scp](#))
- Use of a shared directory mounted by both containers
- Direct download of the SQL file within the MariaDB container

These methods are possible but require more network or configuration effort. The **Host intermediate step** is therefore **simple, safe, and reproducible**.

## Step 6.3: Create Configuration Database in MariaDB

### 💡 Explanation:

Although the MariaDB database runs in a separate container (CT 120), we access the database **via the Proxmox Host** instead of entering the container directly:

- **Central Access:** From the Host, we can reach every container directly without having to log in first.
- **No unnecessary container login:** MariaDB accepts network connections, so the MySQL client is sufficient.
- **Flexibility:** Multiple containers and databases can be reached simultaneously from the Host.

#### 💡 Alternative Method:

- Direct access in the container via `pct exec 120 -- mysql ...` or `pct enter 120`.
- Advantage: directly in the container.
- Disadvantage: additional steps, less centralized.

Log in to the MariaDB container **from the Proxmox Host**:

```
mysql -u pdal -p -h 192.168.137.120
```

```
root@proxmox:~# mysql -u pdal -p -h 192.168.137.120
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 124
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.
```

Prerequisite: we have installed the mariadb-client-core (on the Proxmox Host)

```
root@proxmox:~# apt install mariadb-client-core
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libmariadb3 mariadb-common mysql-common
The following NEW packages will be installed:
  libmariadb3 mariadb-client-core mariadb-common mysql-common
0 upgraded, 4 newly installed, 0 to remove and 1 not upgraded.
Need to get 1,101 kB of archives.
After this operation, 15.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.de.debian.org/debian bookworm/main amd64 mysql-com
Get:2 http://ftp.de.debian.org/debian bookworm/main amd64 mariadb-c
Get:3 http://ftp.de.debian.org/debian bookworm/main amd64 libmariadb
```

At the MySQL prompt:

```
CREATE DATABASE phpmyadmin;
EXIT;
```

## Step 6.4: Execute SQL File and Create Tables

### **Explanation:**

In this step, we execute the previously copied SQL file `create_tables.sql` in the MariaDB container. This creates the necessary tables for phpMyAdmin.

#### 1. Establish connection to MariaDB (from Proxmox Host):

```
bash mysql -h 192.168.137.120 -u pdal -p phpmyadmin < /tmp/create_tables.sql
```

- `-h 192.168.137.120` → Address of the MariaDB container
- `-u pdal` → User with privileges to create the phpMyAdmin tables
- `-p` → Password prompt for the user
- `phpmyadmin` → Database where the tables will be created

#### 2. Execute the SQL file:

- All tables and structures defined in `create_tables.sql` are automatically created in the `phpmyadmin` database.
- This activates the extended features of phpMyAdmin such as Bookmarks, PDF Export, and Relation features.

#### 3. No manual input necessary:

- The file contains all `CREATE TABLE` commands, so no SQL commands need to be entered individually.

### **In Summary:**

- With this step, we initialize the phpMyAdmin-specific tables in the database so that the web interface can be used correctly and completely.

```
root@proxmox:~# mysql -h 192.168.137.120 -u pdal -p phpmyadmin < /tmp/create_tables.sql
Enter password:
root@proxmox:~#
```

## Step 6.5: Grant Privileges to User Pdal for Configuration Storage

Note: The user `pdal` already exists and has full privileges, so creating a new user is omitted.

## Step 6.6: Adjust phpMyAdmin Configuration

Open the file on the Apache container:

```
sudo nano /etc/phpmyadmin/config.inc.php
```

Add the following lines (replace existing lines with the following lines if they differ):

```
$cfg['Servers'][$i]['controluser'] = 'pdal';
$cgi['Servers'][$i]['controlpass'] = 'JadeHS20';
$cgi['Servers'][$i]['pmadb'] = 'phpmyadmin';
```

```
$cfg['Servers'][$i]['bookmarktable'] = 'pma_bookmark';
$cfg['Servers'][$i]['relation'] = 'pma_relation';
$cfg['Servers'][$i]['table_info'] = 'pma_table_info';
$cfg['Servers'][$i]['table_coords'] = 'pma_table_coords';
$cfg['Servers'][$i]['pdf_pages'] = 'pma_pdf_pages';
$cfg['Servers'][$i]['column_info'] = 'pma_column_info';
$cfg['Servers'][$i]['history'] = 'pma_history';
$cfg['Servers'][$i]['designer_coords'] = 'pma_designer_coords';
$cfg['Servers'][$i]['tracking'] = 'pma_tracking';
$cfg['Servers'][$i]['userconfig'] = 'pma_userconfig';
$cfg['Servers'][$i]['recent'] = 'pma_recent';
$cfg['Servers'][$i]['favorite'] = 'pma_favorite';
$cfg['Servers'][$i]['users'] = 'pma_users';
$cfg['Servers'][$i]['usergroups'] = 'pma_usergroups';
$cfg['Servers'][$i]['navigationhiding'] = 'pma_navigationhiding';
$cfg['Servers'][$i]['savedsearches'] = 'pma_savedsearches';
$cfg['Servers'][$i]['central_columns'] = 'pma_central_columns';
$cfg['Servers'][$i]['designer_settings'] = 'pma_designer_settings';
$cfg['Servers'][$i]['export_templates'] = 'pma_export_templates';
```

```
//$cfg['Servers'][$i]['compress'] = false;
/* Optional: User for advanced features */
$cfg['Servers'][$i]['controluser'] = $dbuser;
$cfg['Servers'][$i]['controlpass'] = $dbpass;
/* Optional: User for advanced features */
$cfg['Servers'][$i]['controluser'] = 'Pdal';
$cfg['Servers'][$i]['controlpass'] = 'JadeHS20';
```

```
/* Optional: Advanced phpMyAdmin features */
$cfg['Servers'][$i]['pmadb'] = $dbname;
$cfg['Servers'][$i]['bookmarktable'] = 'pma_bookmark';
$cfg['Servers'][$i]['relation'] = 'pma_relation';
$cfg['Servers'][$i]['table_info'] = 'pma_table_info';
$cfg['Servers'][$i]['table_coords'] = 'pma_table_coords';
$cfg['Servers'][$i]['pdf_pages'] = 'pma_pdf_pages';
$cfg['Servers'][$i]['column_info'] = 'pma_column_info';
$cfg['Servers'][$i]['history'] = 'pma_history';
$cfg['Servers'][$i]['table_uiprefs'] = 'pma_table_uiprefs';
$cfg['Servers'][$i]['tracking'] = 'pma_tracking';
$cfg['Servers'][$i]['userconfig'] = 'pma_userconfig';
$cfg['Servers'][$i]['recent'] = 'pma_recent';
$cfg['Servers'][$i]['favorite'] = 'pma_favorite';
$cfg['Servers'][$i]['users'] = 'pma_users';
$cfg['Servers'][$i]['usergroups'] = 'pma_usergroups';
$cfg['Servers'][$i]['navigationhiding'] = 'pma_navigationhiding';
$cfg['Servers'][$i]['savedsearches'] = 'pma_savedsearches';
$cfg['Servers'][$i]['central_columns'] = 'pma_central_columns';
$cfg['Servers'][$i]['designer_settings'] = 'pma_designer_settings';
$cfg['Servers'][$i]['export_templates'] = 'pma_export_templates';
```

```
/* Optional: Advanced phpMyAdmin features */
$cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';
$cfg['Servers'][$i]['bookmarktable'] = 'pma_bookmark';
$cfg['Servers'][$i]['relation'] = 'pma_relation';
$cfg['Servers'][$i]['table_info'] = 'pma_table_info';
$cfg['Servers'][$i]['table_coords'] = 'pma_table_coords';
$cfg['Servers'][$i]['pdf_pages'] = 'pma_pdf_pages';
$cfg['Servers'][$i]['column_info'] = 'pma_column_info';
$cfg['Servers'][$i]['history'] = 'pma_history';
$cfg['Servers'][$i]['designer_coords'] = 'pma_designer_coords';
$cfg['Servers'][$i]['tracking'] = 'pma_tracking';
$cfg['Servers'][$i]['userconfig'] = 'pma_userconfig';
$cfg['Servers'][$i]['recent'] = 'pma_recent';
$cfg['Servers'][$i]['favorite'] = 'pma_favorite';
$cfg['Servers'][$i]['users'] = 'pma_users';
$cfg['Servers'][$i]['usergroups'] = 'pma_usergroups';
$cfg['Servers'][$i]['navigationhiding'] = 'pma_navigationhiding';
$cfg['Servers'][$i]['savedsearches'] = 'pma_savedsearches';
$cfg['Servers'][$i]['central_columns'] = 'pma_central_columns';
$cfg['Servers'][$i]['designer_settings'] = 'pma_designer_settings';
$cfg['Servers'][$i]['export_templates'] = 'pma_export_templates';
```

## Step 6.7: Restart Apache in the Container

```
systemctl restart apache2
```

```
pdal@apache101:~$ sudo systemctl restart apache2.service
pdal@apache101:~$ █
```

## Result

After setup, the warning message in phpMyAdmin should be gone:

"The phpMyAdmin configuration storage is not completely configured, some extended features have been deactivated."

Additional features like Designer and Relations are now available.

---

## Task (Optional): Set up an Alias for PhpMyAdmin

The document "Apache2 Web Server & User Management in the LXC Container" explains how to set up an alias.

Move the PhpMyAdmin directory from `/var/www/html/phpmyadmin` to `/var/www/phpmyadmin` and set up an **Alias** for PhpMyAdmin. This keeps the HTML directory free for your applications.

## Sources

- "Introduction — phpMyAdmin 6.0.0-dev documentation". Accessed: September 25, 2025. [Online]. Available at: [Introduction](#)
  - "Requirements — phpMyAdmin 6.0.0-dev documentation". Accessed: September 25, 2025. [Online]. Available at: [Requirements](#)
  - "Installation — phpMyAdmin 6.0.0-dev documentation". Accessed: September 25, 2025. [Online]. Available at: [Installation](#)
  - "Configuration — phpMyAdmin 6.0.0-dev documentation". Accessed: September 25, 2025. [Online]. Available at: [Configuration](#)
  - "User Guide — phpMyAdmin 6.0.0-dev documentation". Accessed: September 25, 2025. [Online]. Available at: [User Guide](#)
  - "FAQ - Frequently Asked Questions — phpMyAdmin 6.0.0-dev documentation". Accessed: September 25, 2025. [Online]. Available at: [FAQ](#)
- 

## License

This work is licensed under the **Creative Commons Attribution - ShareAlike 4.0 International License**.

[To the license text on the Creative Commons website](#)