

Installation

1. We will not use any of the standard path used in Debian or other distributions. All files related to phpMyAdmin will remain in the custom path (e.g. /home/user/html/phpMyAdmin) where we download the files and which is the easiest way for maintenance and later version upgrades.
2. Navigate to /home/user/html, download the latest stable tarball, extract it, and rename the folder:

```
cd /home/user/html
sudo wget
https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-english.tar.gz
sudo tar -xvzf phpMyAdmin-latest-english.tar.gz
sudo rm phpMyAdmin-latest-english.tar.gz
sudo mv phpMyAdmin-*-english phpmyadmin
```

3. Create the configuration file from the sample template and generate a secure blowfish secret for cookie encryption:

```
sudo cp /home/user/html/phpmyadmin/config.sample.inc.php
/usr/share/phpmyadmin/config.inc.php
```

4. Generate a 32-character random string using OpenSSL:

```
openssl rand -base64 24 > secret
```

5. Open "/home/user/html/phpmyadmin/config.inc.php and paste the generated string into the \$cfg['blowfish_secret'] line:php\$cfg['blowfish_secret'] = 'YOUR_GENERATED_STRING_HERE';</code>

Use code with caution. Create the temporary folder required by phpMyAdmin and assign proper web server ownership: `bashsudo mkdir -p /var/lib/phpmyadmin/tmp sudo chown -R www-data:www-data /var/lib/phpmyadmin /usr/share/phpmyadmin sudo chmod -R 0755 /usr/share/phpmyadmin`

1. Unzip the tarball, then move the folder to a local folder outside of apache's document root (we will create a path alias in phpMyAdmin's `apache.conf` later) and set ownership:

```
$ tar xvf phpMyAdmin-5.2.3-english.tar.gz
sudo mv phpMyAdmin-5.2.3-english/ /home/user/html/phpMyAdmin
$ sudo chown -R www-data:www-data /home/user/html/phpMyAdmin
```

2. To upgrade to a newer version simply download the new version and extract it to the same folder. Files created by the user will remain the same, e.g. `apache.conf`, `config.inc.php`, and `htpasswd.setup`.
- Note: Debian 10 dropped *phpmyadmin* from it's repository. Older and newer Debian versions include it.

Step 2:

Configuration

1. We will not use any of the standard path used in Debian versions up to 9 or in other distributions. All files related to phpMyAdmin will remain in the custom path where we moved the downloaded files which is the easiest way for maintenance and later version upgrades (/home/user/html/phpMyAdmin).

2. Make a new directory for phpMyAdmin to store its temporary files:

```
$ sudo mkdir -m770 /home/user/html/phpMyAdmin/tmp
```

3. Copy config.sample.inc.php to config.inc.php and edit as follows

1. Use the [phpMyAdmin blowfish secret generator](#) to create a new secret passphrase for cookie authentication:

```
$ sudo vim /home/user/html/phpMyAdmin/config.inc.php
$cfg['blowfish_secret'] = 'new 32 byte secret key';
```

2. Add the following custom settings to config.inc.php:

```
$cfg['FirstLevelNavigationItems'] = 150;          // number of
databases in navigation, default: 100
$cfg['MaxNavigationItems'] = 150;                // number of tables in
db navigation, default: 50
$cfg['NavigationWidth'] = 300;                   // width of the
navigation window, default: 240
$cfg['RetainQueryBox'] = true;                   // retain query box,
results of query shown below box, default: false
$cfg['ShowPhpInfo'] = true;                      // show phpinfo link on
home screen, default: false
$cfg['TempDir'] = '/home/user/html/phpMyAdmin/tmp'; // you may
omit this line as the default is ./tmp
```

3. You may check [phpMyAdmin's documentation](#) for other settings to add
 4. Leave the commented out settings in config.inc.php unchanged. The pma settings are better done within phpMyAdmin, where you click "Find out why" in the warning at the bottom of the screen when you first run phpMyAdmin, and then create the database phpmyadmin which will contain those settings.
4. Create /home/user/html/phpMyAdmin/apache.conf:

```
# phpMyAdmin default Apache configuration

Alias /phpmyadmin /home/user/html/phpMyAdmin

# Secure access to phpMyAdmin by restricting access to it's parent, for
example by IP address or domain name, local or external
<Directory /home/user/html>
    <RequireAny>
        Require all denied
        Require ip 127.0.0.1
    </RequireAny>
</Directory>
```

```
<Directory /home/user/html/phpMyAdmin>
    Options SymLinksIfOwnerMatch
    DirectoryIndex index.php

    <IfModule mod_php.c>
        <IfModule mod_mime.c>
            AddType application/x-httpd-php .php
        </IfModule>
        <FilesMatch ".+\.php$">
            SetHandler application/x-httpd-php
        </FilesMatch>

        php_value include_path .
        php_admin_value upload_tmp_dir /home/user/html/phpMyAdmin/tmp
        php_admin_value open_basedir
/home/user/html/phpMyAdmin/:/usr/share/
    </IfModule>

</Directory>

# Authorize for setup
<Directory /home/user/html/phpMyAdmin/setup>
    <IfModule mod_authz_core.c>
        <IfModule mod_authn_file.c>
            AuthType Basic
            AuthName "phpMyAdmin Setup"
            AuthUserFile /home/user/html/phpMyAdmin/htpasswd.setup
        </IfModule>
        Require valid-user
    </IfModule>
</Directory>

# Disallow web access to directories that don't need it
<Directory /home/user/html/phpMyAdmin/templates>
    Require all denied
</Directory>
<Directory /home/user/html/phpMyAdmin/libraries>
    Require all denied
</Directory>
<Directory /home/user/html/phpMyAdmin/setup/lib>
    Require all denied
</Directory>
```

5. You can replace the directives for the setup directory with `Require all denied` as we will not use the setup script. However, if you do want to use the setup script prior to creating `config.inc.php`, first create the password file to access the setup script's directory:

```
$ sudo htpasswd -c /home/user/html/phpMyAdmin/htpasswd.setup user
```

6. Symlink the configuration file for Apache and restart the service:

```
$ sudo ln -s /home/user/html/phpMyAdmin/apache.conf /etc/apache2/conf-
```

```
enabled/phpmyadmin.conf  
$ sudo service apache2 restart
```

7. Create a regular MariaDB user for the purpose of managing databases through phpMyAdmin, if you haven't done that yet when configuring MariaDB. You could create a user that has privileges to all tables within the database, as well as the power to add, change, and remove user privileges, with this command. Whatever privileges you assign to this user, be sure to give it a strong password as well:

```
$ sudo mariadb  
MariaDB [(none)]> CREATE USER 'user'@'localhost' IDENTIFIED BY  
'password';  
MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'user'@'hostname';  
MariaDB [(none)]> FLUSH PRIVILEGES;  
MariaDB [(none)]> exit
```

8. Make sure permissions are set to traverse directories, particularly /home/user/html/ and /home/user/www/

From:

<http://wiki.condrau.com/> - **Bernard's Wiki**

Permanent link:

<http://wiki.condrau.com/deb13:phpmyadmin?rev=1790392556>

Last update: **2026/09/26 10:15**

