How to Install phpMyAdmin with Nginx (LEMP) on Ubuntu 20.04 LTS

This tutorial will be showing you how to install phpMyAdmin with Nginx, MariaDB and PHP7.4 (LEMP) on Ubuntu 20.04. phpMyAdmin is a free and open-source web-based database management tool written in PHP. It provides a graphical web interface for users to manage MySQL or MariaDB database.

phpMyAdmin allows administrators to:

- browse through databases and tables;
- create, copy, rename, alter and drop databases;
- create, copy, rename, alter and drop tables;
- perform table maintenance;
- add, edit and drop fields;
- execute any SQL-statement, even multiple queries;
- create, alter and drop indexes;
- load text files into tables;
- create and read dumps of tables or databases;
- export data to SQL, CSV, XML, Word, Excel, PDF and LaTeX formats;
- administer multiple servers;
- manage MySQL users and privileges;
- check server settings and runtime information with configuration hints;
- check referential integrity in MyISAM tables;
- create complex queries using Query-by-example (QBE), automatically
- connecting required tables;
- create PDF graphics of database layout;
- search globally in a database or a subset of it;
- transform stored data into any format using a set of predefined functions, such as displaying BLOB-data as image or download-link;
- manage InnoDB tables and foreign keys;

Prerequisites

To follow this tutorial, you need to have an Ubuntu 20.04 OS running on your local computer or on a remote server.

If you are looking for a VPS (Virtual Private Server), then you can click this special link to get $100 free credit on DigitalOcean. (For new users only). If you are already a DigitalOcean user, then you can click this special link to get $50 free credit on Vultr (for new users only).

It is assumed that you have already installed LEMP stack on Ubuntu 20.04. If not, please check out the following tutorial.

- How to Install LEMP stack (Nginx, MariaDB, PHP7.4) on Ubuntu 20.04

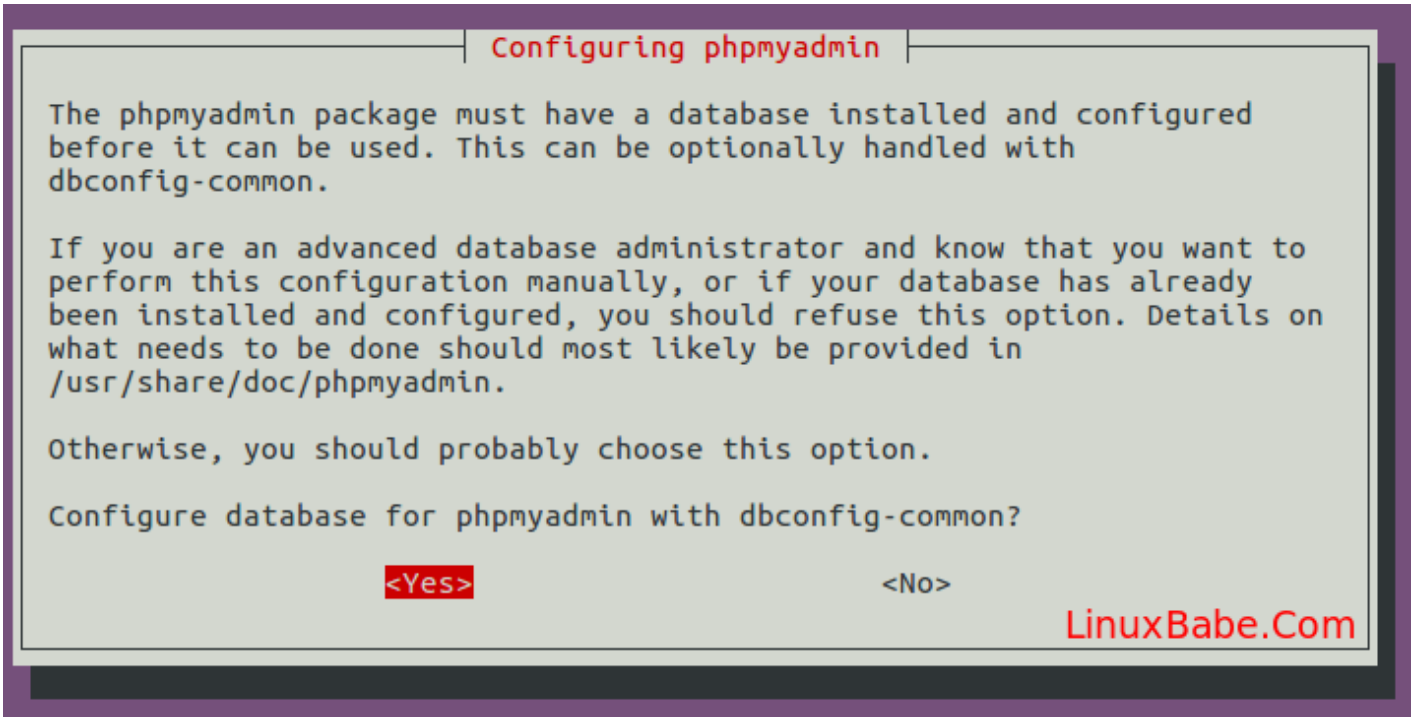
With that out of the way, let's get started with installing phpMyAdmin.

Step 1: Download and Install phpMyAdmin on Ubuntu 20.04

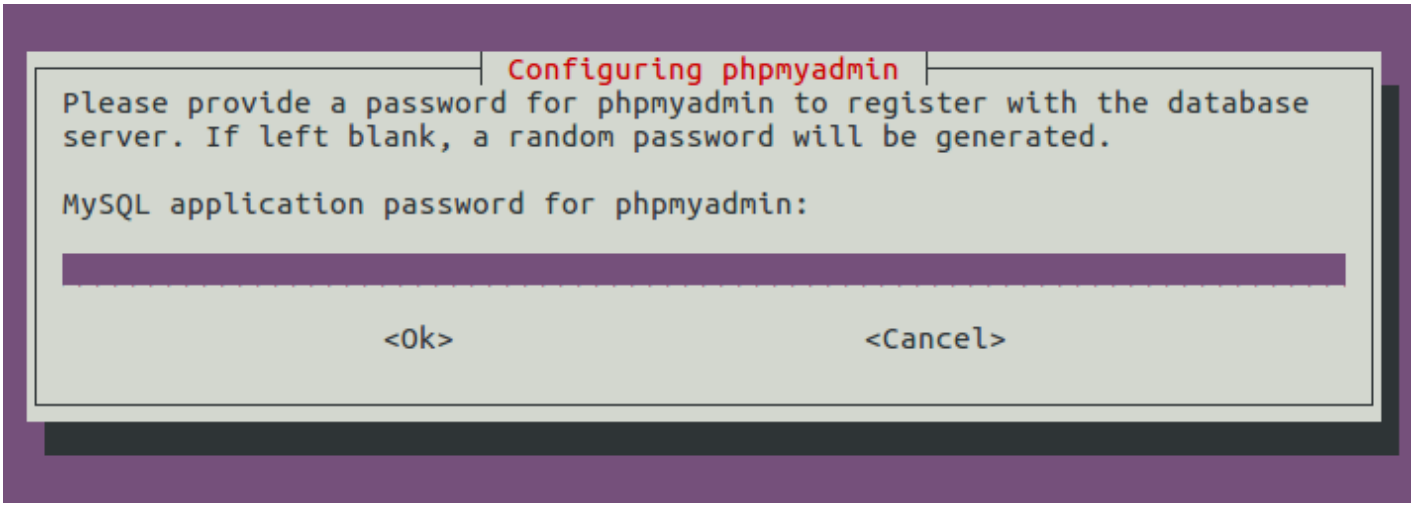
phpMyAdmin is included in Ubuntu 20.04 software repository, so we can easily install it with the following command.

```
sudo apt update
sudo apt install phpmyadmin
```

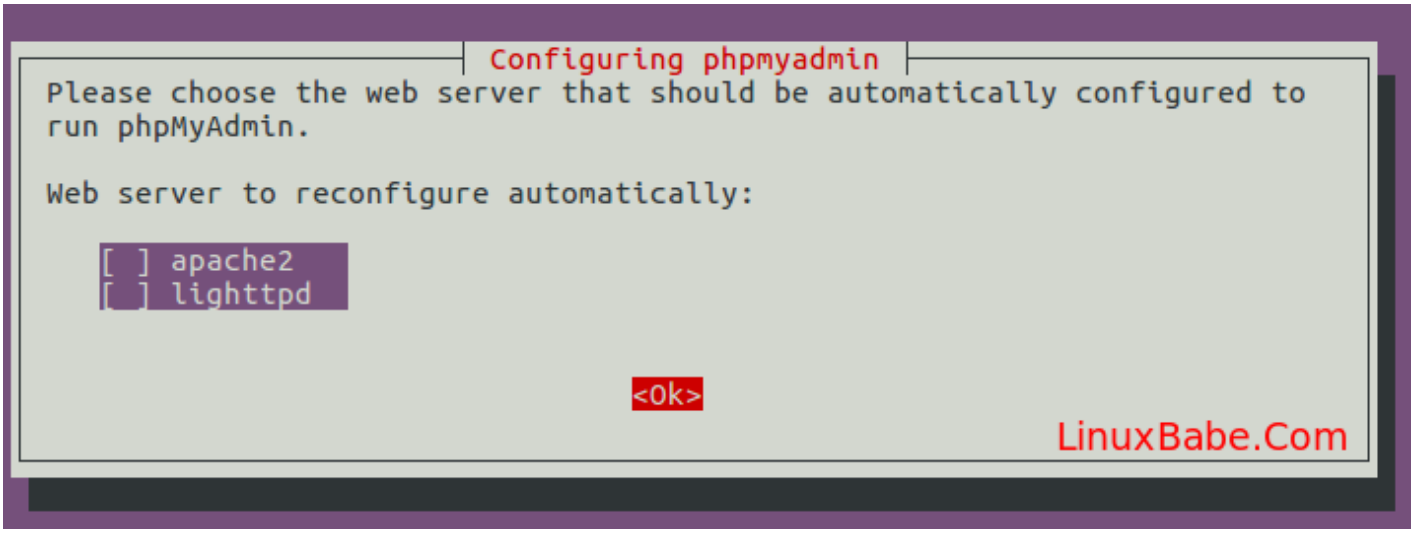
The above command will install all necessary dependencies including PHP7 extensions. During the installation, it will ask if you want to use `dbconfig-common` to configure the database. Press Tab key to select Yes.



This will also create a new database user named `phpmyadmin`. Give this user a password.



Next, it will prompt you to select a web server to configure. Nginx isn't on the list, so press the Tab key and hit OK to skip this step.



Once done, a new database named `phpmyadmin` is created and the database user `phpmyadmin` has necessary privileges to manage this database. If you are curious as I am, you can log into MariaDB and check what privileges `phpmyadmin` user has been granted.

You can use the following command to log into MariaDB server.

```
sudo mysql -u root
```

Then check the privileges.

```
show grants for phpmyadmin@localhost;
```

Output:

```
MariaDB [(none)]> show grants for phpmyadmin@localhost;
+----------------------------------------------------------------------------------------------------------------+
| Grants for phpmyadmin@localhost                                                                                |
+----------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'phpmyadmin'@'localhost' IDENTIFIED BY PASSWORD '*[redacted]' |
| GRANT ALL PRIVILEGES ON `phpmyadmin`.* TO 'phpmyadmin'@'localhost'                                             |
+----------------------------------------------------------------------------------------------------------------+
2 rows in set (0.00 sec)
```

As you can see, user `phpmyadmin` has all privileges on database `phpmyadmin`. Now you can exit by executing:

```
exit;
```

Step 2: Create Nginx Server Block for phpMyAdmin

To be able to access the phpMyAdmin web interface, we need to create a Nginx server block by running the following command.

```
sudo nano /etc/nginx/conf.d/phpmyadmin.conf
```

We will configure it so that we can access phpMyAdmin via a sub-domain. Paste the following text into the file. Replace `pma.example.com` with your actual sub-domain and don't forget to create DNS A record for it.

```
server {
  listen 80;
  listen [::]:80;
  server_name pma.example.com;
  root /usr/share/phpmyadmin/;
  index index.php index.html index.htm index.nginx-debian.html;

  access_log /var/log/nginx/phpmyadmin_access.log;
  error_log /var/log/nginx/phpmyadmin_error.log;

  location / {
    try_files $uri $uri/ /index.php;
  }

  location ~ ^/(doc|sql|setup)/ {
    deny all;
  }

  location ~ \.php$ {
    fastcgi_pass unix:/run/php/php7.4-fpm.sock;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
    include snippets/fastcgi-php.conf;
  }
```

```
    location ~ /\.ht {
        deny all;
    }
}
```

Your phpMyAdmin files are in `/usr/share/phpmyadmin/` directory. Save and close the file. Then test Nginx configurations.

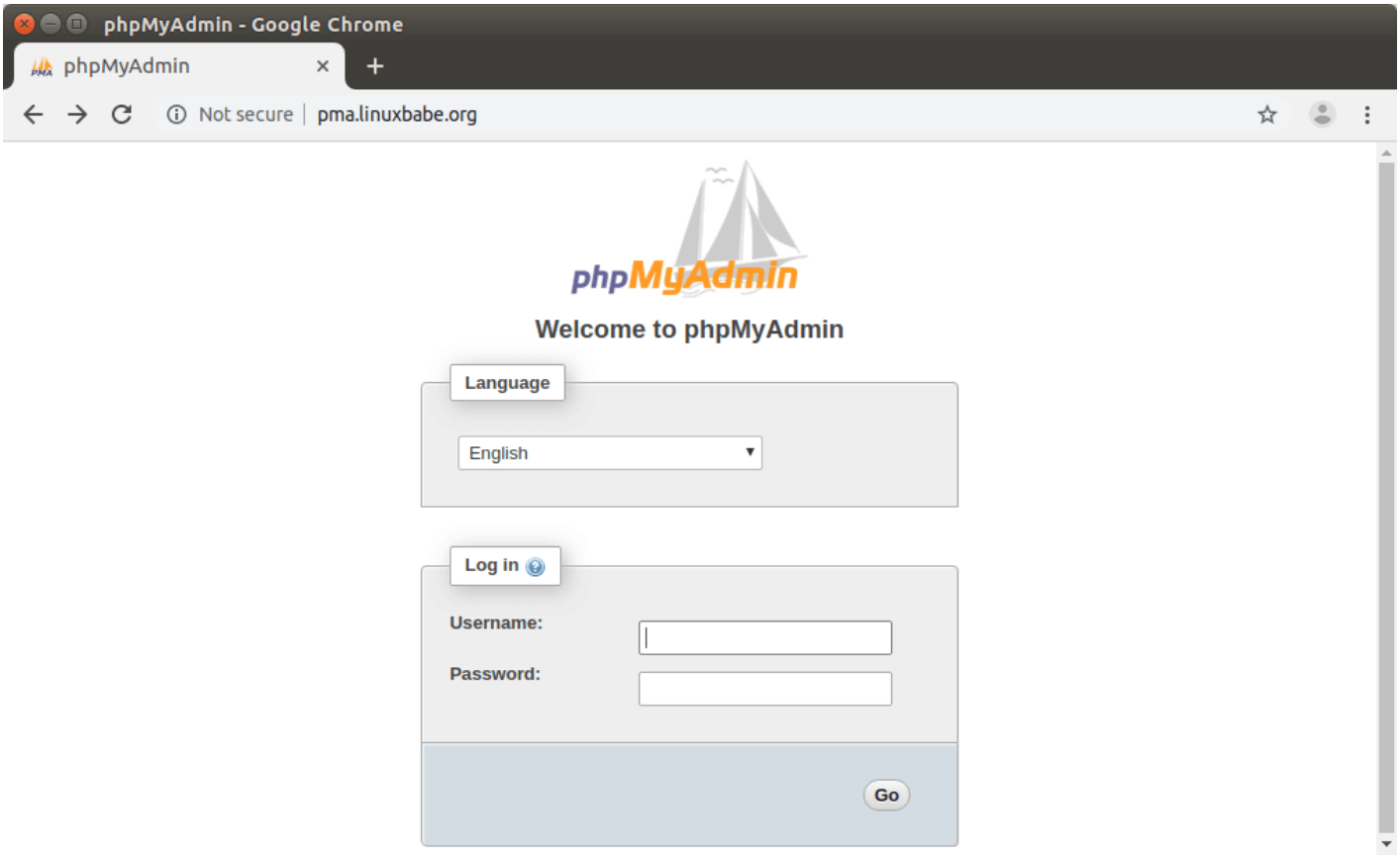
```
sudo nginx -t
```

If the test is successful, reload Nginx for the changes to take effect.

```
sudo systemctl reload nginx
```

Now you should be able to access phpMyAdmin web interface via

```
pma.example.com
```



Step 3: Installing TLS Certificate

To secure the phpMyadmin web interface, we can install a free Let's Encrypt TLS certificate. Install

the Let’s Encrypt client from Ubuntu 20.04 software repository like below:

```
sudo apt install certbot python3-certbot-nginx
```

`Python3-certbot-nginx` is the Nginx plugin for Certbot. Now run the following command to obtain and install TLS certificate.

```
sudo certbot --nginx --agree-tos --redirect --hsts --staple-ocsp -d pma.example.com --email you@example.com
```

Where:

- **–nginx**: Use the Nginx authenticator and installer
- **–agree-tos**: Agree to Let’s Encrypt terms of service
- **–redirect**: Enforce HTTPS by 301 redirect.
- **–hsts**: Add the Strict-Transport-Security header to every HTTP response.
- **–staple-ocsp**: Enables OCSP Stapling.
- **–must-staple**: Adds the OCSP Must Staple extension to the certificate.
- **-d** flag is followed by a list of domain names, separated by a comma. You can add up to 100 domain names.
- **–email**: Email used for registration and recovery contact.

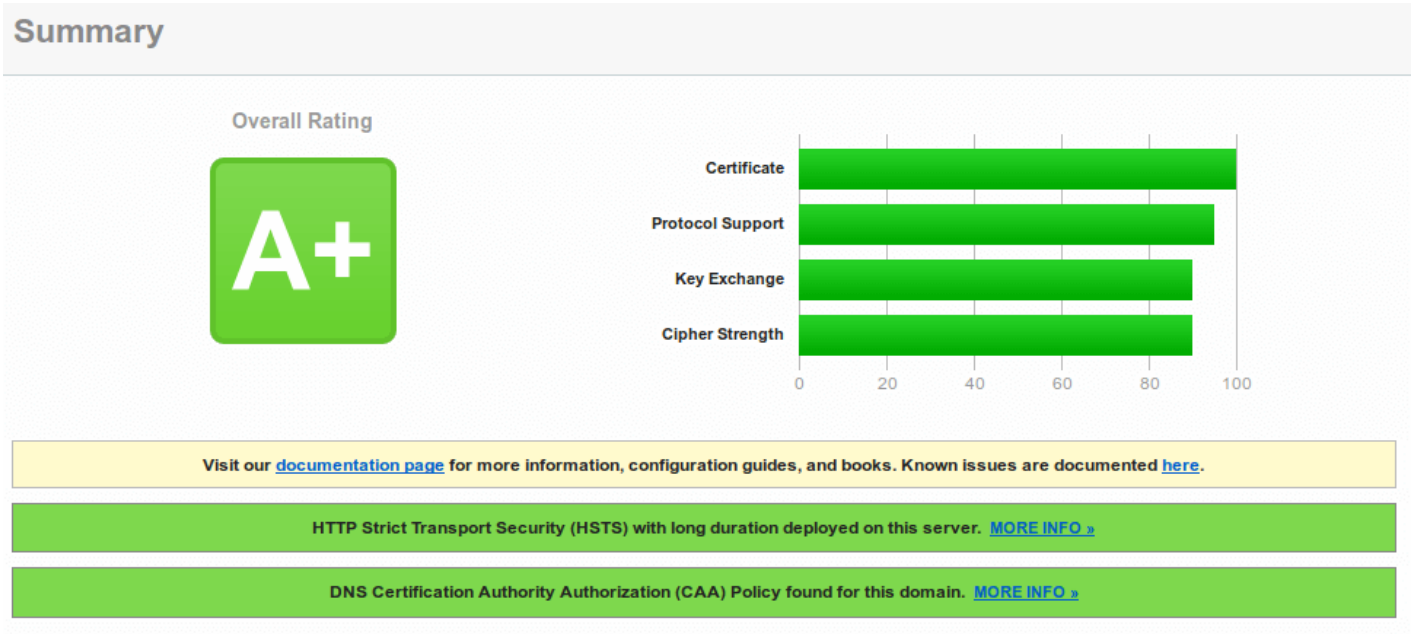
You will be asked if you want to receive emails from EFF(Electronic Frontier Foundation). After choosing Y or N, your TLS certificate will be automatically obtained and configured for you, which is indicated by the message below.

```
IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/pma.linuxbabe.com/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/pma.linuxbabe.com/privkey.pem
   Your cert will expire on 2020-08-10. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot
   again. To non-interactively renew *all* of your certificates, run
   "certbot renew"
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
   Donating to EFF:                    https://eff.org/donate-le
```

Step 4: Test Your TLS Certificate

Go to ssllabs.com to test your TLS certificate and configuration. You should get **A+** because HSTSis enabled.



Step 5: Troubleshoot phpMyAdmin Login Error

If you login with MariaDB root account, you may see the following error.

```
#1698 - Access denied for user 'root'@'localhost'
```

and

```
mysqli_real_connect(): (HY000/1698): Access denied for user 'root'@'localhost'
```

If you login with user `phpmyadmin`, you won't see the above error. However, user `phpmyadmin` can only be used to administer the `phpmyadmin` database. The cause of the error is that by default MariDB root user is authenticated via the unix_socket plugin, instead of using the `mysql_native_password` plugin. To get around this issue, we can create another admin user and grant all privileges to the new admin user.

Log into MariaDB server from the command line.

```
sudo mariadb -u root
```

Create an admin user with password authentication.

```
create user admin@localhost identified by 'your-chosen-password';
```

Grant all privileges on all databases.

```
grant all privileges on *.* to admin@localhost with grant option;
```

Flush privileges and exit;

```
flush privileges;


exit;
```

Now you can log into phpMyAdmin with the admin account and manage all databases.

TLS Certificate Auto-Renewal

To automatically renew Let’s Encrypt certificate, simply edit root user’s crontab file.

```
sudo crontab -e
```

Then add the following line at the bottom.

```
@daily certbot renew --quiet && systemctl reload nginx
```

Reloading Nginx is needed for it to pick up the new certificate to clients.

Revision #1
Created 11 July 2021 18:59:37 by Admin
Updated 11 July 2021 19:00:05 by Admin