

# Instalasi LAMP Stack (Apache - MySQL/MariaDB - PHP ) di Ubuntu

## Step 1 – Update system

First, let's update your system to ensure that you have the latest versions of software packages and security updates, while also avoiding any potential errors that may arise from outdated versions.

```
sudo apt update && sudo apt upgrade -y
```

## Step 2 – Install Apache ( Web server)

Run the following command to install the Web server with Apache:

```
sudo apt install apache2 -y
```

After the installation, start and enable the Apache service to run with the system every time it restarts.

```
sudo systemctl start apache2
```

```
sudo systemctl enable apache2
```

Check the status of the Apache service and ensure that it is running:

```
sudo systemctl status apache2
```

If successful, you will see its status as **running** as shown in the example below:

install-lamp-stack-on-ubuntu-23

By default, the Apache service listens on ports **80** and **443** (Secure). We need to allow these two ports through the UFW firewall. (Skip this step if you are not using the UFW firewall.)

```
sudo ufw enable
```

```
sudo ufw allow 80
```

```
sudo ufw allow 443
```

```
sudo ufw reload
```

```
sudo ufw status
```

### Output:

install-lamp-stack-on-ubuntu-23

Finally, access the server address. If you see a similar page as shown below, it means that Apache has been successfully installed.

install-lamp-stack-on-ubuntu-23

# Step 2 – Install MySQL/MariaDB Database Server

## Install MySQL/MariaDB

In this article, we will install **MariaDB** as the database server. MariaDB is an open-source database management system that is fully compatible with MySQL. To install it, run the following command:

```
sudo apt install mariadb-server -y
```

After the installation, start and enable the MariaDB server to automatically start with the system reboot using the following command:

```
sudo systemctl start mariadb
```

```
sudo systemctl enable mariadb
```

Check the status of the MariaDB server and ensure that it is running on the system, use the following command:

```
sudo systemctl status mariadb
```

[install lamp stack on ubuntu-23](#)

## Secure MySQL/MariaDB

Next, let's configure the security settings for your MySQL/MariaDB to ensure that it is properly set up and secure after the installation. Use the following command to initiate the security configuration:

```
sudo mysql_secure_installation
```

To configure the security settings for MySQL/MariaDB, please provide answers to the following questions:

```
sudo mysql_secure_installation
```

“ NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current password for the root user. If you've just installed MariaDB, and haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):  
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB

SERVICES IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current password for the root user. If you've just installed MariaDB, and haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):  
OK, successfully used password, moving on...

Setting the root password or using the unix\_socket ensures that nobody can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix\_socket authentication [Y/n] N #Type N and hit Enter  
... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] #Keep default and hit Enter  
New password: # Enter password for root and hit Enter  
Re-enter new password: # Re-password for root and hit Enter  
Password updated successfully!  
Reloading privilege tables..  
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone to log into MariaDB without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

Remove anonymous users? [Y/n] #Keep default and hit Enter  
... Success!

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] #Keep default and hit Enter  
... Success!

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? [Y/n]    **#Keep default and hit Enter**

- Dropping test database...

... Success!

- Removing privileges on test database...

... Success!

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? [Y/n]    **#Keep default and hit Enter**

... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!

r none):

OK, successfully used password, moving on...

Setting the root password or using the unix\_socket ensures that nobody can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix\_socket authentication [Y/n] N    **#Type N and hit Enter**

... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n]    **#Keep default and hit Enter**

New password:    **# Enter password for root and hit Enter**

Re-enter new password:    **# Re-password for root and hit Enter**

Password updated successfully!

Reloading privilege tables..

... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone to log into MariaDB without having to have a user account created for

them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

Remove anonymous users? [Y/n]    #Keep default and hit Enter  
... Success!

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n]    #Keep default and hit Enter  
... Success!

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? [Y/n]    #Keep default and hit Enter  
- Dropping test database...  
... Success!  
- Removing privileges on test database...  
... Success!

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? [Y/n]    #Keep default and hit Enter  
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!

Now, please try logging into your MySQL/MariaDB using the root account and the password previously set with the following command:

```
sudo mysql -uroot -p
```

```
exit;
```

## Step 3 – Install PHP

Next, we will install PHP and its extensions to make it compatible with Apache and MySQL/MariaDB using the following command:

```
sudo apt install php libapache2-mod-php php-mysql php-mbstring -y
```

After installation, you can check the installed PHP version using the following command:

```
php -v
```

If you see information about the current PHP version, it means PHP has been installed successfully, and the necessary extensions have been activated.

[install lamp stack on ubuntu-23](#)

## Step 5: Test PHP Processing on Web Server

To test the new LAMP installation, create a basic PHP script and place it in the web root directory located at `/var/www/html/`, then check if the script is accessible via an internet browser. The steps below explain the procedure for performing this test.

Create a file in the web root directory by typing the following command:

```
sudo nano /var/www/html/info.php
```

Add new content below:

```
<?php
phpinfo ();
?>
```

After finishing, save and exit the file by pressing `ctrl+o` and `enter` ,and then `ctrl+x` and then `enter`

[install lamp stack on ubuntu-23](#)

And restart Apache service to apply the changes:

```
sudo systemctl restart apache2
```

Now, access the address <http://IP-Server/info.php> .If you see a similar image as shown below, it means PHP is compatible with Apache.

[install lamp stack on ubuntu-23](#)

# Conclusion

That's it! Through this [How to Install and Configure the LAMP stack on Ubuntu 23](#) guide, you have successfully installed the LAMP stack on your server. If you have any comments or contributions, please feel free to leave them in the comments section below.

- [Instalasi Snipe IT Manajemen Asset TIK pada Ubuntu 22](#)

---

Revision #7

Created 17 August 2024 18:12:27 by Admin Diskominfo

Updated 27 August 2024 17:15:08 by Admin Diskominfo