Table of Contents

[**Mode Control Web Server Setup** 1](#_Toc508125807)

[1. Installing MySQL or MariaDB Database 1](#_Toc508125808)

[1.1 Install MySQL 5.7 1](#_Toc508125809)

[1.2 Install MariaDB 10 2](#_Toc508125810)

[2. Install Apache Web Server 3](#_Toc508125811)

[3. Install PHP 7 3](#_Toc508125812)

[4. Test PHP and get details about your PHP installation 4](#_Toc508125813)

[5. Get MySQL / MariaDB support in PHP 5](#_Toc508125814)

[6. Import database 6](#_Toc508125815)

[6. Upload - Config source code: 7](#_Toc508125816)

# **Mode Control Web Server Setup**

## 1. Installing MySQL or MariaDB Database

There are currently two widely used MySQL database systems available, the classic "MySQL" server which is developed by Oracle and is available in version 5.7 now and the MySQL fork named MariaDB which is developed by the original MySQL developer Monty Widenius.

I will show you how to install both alternatives below. Just follow either chapter 1.1 or 1.2 but not both. I will use MySQL 5.7 for the virtual machine image that can be downloaded from Howtoforge.

### 1.1 Install MySQL 5.7

To install MySQL 5.7, execute this command:

*apt-get -y install mysql-server mysql-client*

The packages mysql-server and mysql-client are so called 'meta-packages', they install always the latest MySQL version that is available from Ubuntu. Th latest version is currently MySQL 5.7.

We have set the root password for MySQL already during installation, but I would like to remove the anonymous user and test database for security reasons. Run the mysql\_secure\_installation command below to achieve that.

*mysql\_secure\_installation*

You will be asked these questions:

*Securing the MySQL server deployment.*

*Enter password for user root:****<-- Enter the MySQL root password***

*VALIDATE PASSWORD PLUGIN can be used to test passwords  
and improve security. It checks the strength of password  
and allows the users to set only those passwords which are  
secure enough. Would you like to setup VALIDATE PASSWORD plugin?*

*Press y|Y for Yes, any other key for No:  
Using existing password for root.  
Change the password for root ? ((Press y|Y for Yes, any other key for No) :****<-- Choose 'y' here if you like to enable the password validation, I don't need that function, so I choose 'n' here.***

*... skipping.  
By default, a MySQL installation has an anonymous user,  
allowing anyone to log into MySQL 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? (Press y|Y for Yes, any other key for No) :****<-- y*** *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? (Press y|Y for Yes, any other key for No) :****<-- y*** *Success.*

*By default, MySQL 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? (Press y|Y for Yes, any other key for No) :****<-- y*** *- 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? (Press y|Y for Yes, any other key for No) :****<-- y*** *Success.*

*All done!*

The MySQL setup has been secured now.

### 1.2 Install MariaDB 10

Run the following command to install MariaDB-server and client:

*apt-get -y install mariadb-server mariadb-client*

Now we set a root password for MariaDB.

*mysql\_secure\_installation*

You will be asked these questions:

*Enter current password for root (enter for none):****<-- press enter*** *Set root password? [Y/n]****<-- y*** *New password:****<-- Enter the new MariaDB root password here*** *Re-enter new password:****<-- Repeat the password*** *Remove anonymous users? [Y/n]****<-- y*** *Disallow root login remotely? [Y/n]****<-- y*** *Reload privilege tables now? [Y/n]****<-- y***

Test the login to MariaDB with the "mysql command"

*mysql -u root -p*

and enter the MariaDB root password that you've set above.

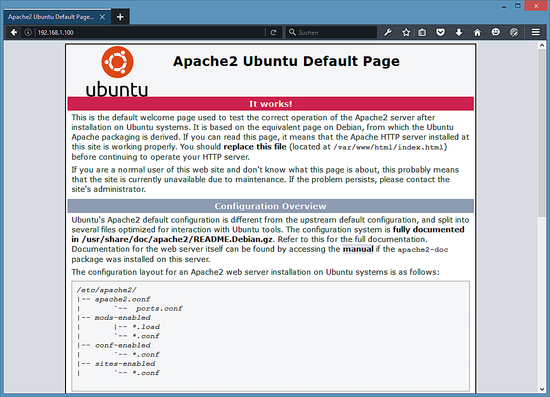
To leave the MariaDB shell, enter the command "quit" and press enter.

## 2. Install Apache Web Server

Apache 2 is available as an Ubuntu package, therefore we can install it like this:

*apt-get -y install apache2*

Now direct your browser to *http://192.168.1.100*, and you should see the Apache2 default page (*It works!*):

[](https://www.howtoforge.com/images/install-apache-with-php-and-mysql-on-ubuntu-16-04-lamp/big/apache_ubuntu_default_page.png)

The document root of the apache default vhost is */var/www/html* on Ubuntu and the main configuration file is */etc/apache2/apache2.conf*. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz.

## 3. Install PHP 7

We can install PHP 7 and the Apache PHP module as follows:

*apt-get -y install php7.0 libapache2-mod-php7.0*

Then restart Apache:

*systemctl restart apache2*

## 4. Test PHP and get details about your PHP installation

The document root of the default web site is */var/www/html*. We will now create a small PHP file (*info.php*) in that directory and call it in a browser. The file will display lots of useful details about our PHP installation, such as the installed PHP version.

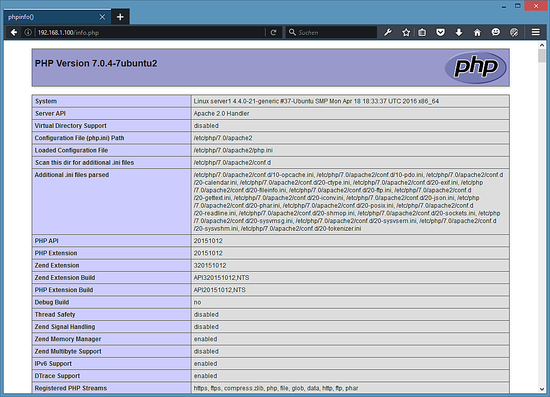
*nano /var/www/html/info.php*

<?php  
phpinfo();  
?>

Then change the owner of the info.php file to the www-data user and group.

*chown www-data:www-data /var/www/html/info.php*

Now we call that file in a browser (e.g. *http://192.168.1.100/info.php*):

[](https://www.howtoforge.com/images/install-apache-with-php-and-mysql-on-ubuntu-16-04-lamp/big/ubuntu_1604_php7_info.png)

As you see, PHP 7.0 is working, and it's working through the *Apache 2.0 Handler*, as shown in the *Server API* line. If you scroll further down, you will see all modules that are already enabled in PHP5. MySQL is not listed there which means we don't have MySQL / MariaDB support in PHP yet.

## 5. Get MySQL / MariaDB support in PHP

To get MySQL support in PHP, we can install the *php7.0-mysql* package. It's a good idea to install some other PHP modules as well as you might need them for your applications. You can search for available PHP modules like this:

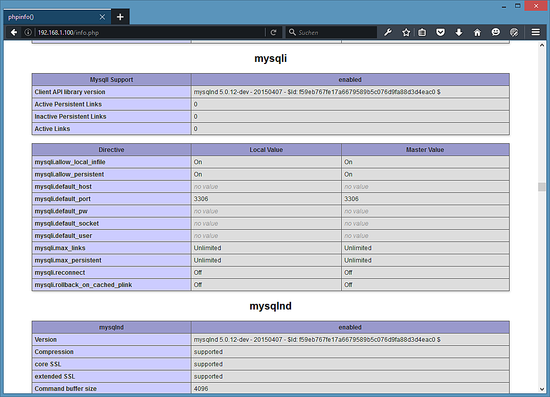
*apt-cache search php7.0*

Pick the ones you need and install them like this:

*apt-get -y install php7.0-mysql php7.0-curl php7.0-gd php7.0-intl php-pear php-imagick php7.0-imap php7.0-mcrypt php-memcache  php7.0-pspell php7.0-recode php7.0-sqlite3 php7.0-tidy php7.0-xmlrpc php7.0-xsl php7.0-mbstring php-gettext*

Now restart Apache2:

*systemctl restart apache2*

[](https://www.howtoforge.com/images/install-apache-with-php-and-mysql-on-ubuntu-16-04-lamp/big/php_mysql_installed.png)

PHP 7 has now MySQL / MariaDB support as shown in phpinfo() above.

## 6. Import database

To import an existing dump file into MySQL or MariaDB, you will have to create the new database. This is where the contents of the dump file will be imported.

First, log in to the database as **root** or another user with sufficient privileges to create new databases.

*mysql -u root -p*

This will bring you into the MySQL shell prompt. Next, create a new database called db\_machine\_web\_controller.

*CREATE DATABASE db\_machine\_web\_controller;*

You'll see this output confirming it was created.

Output

Query OK, 1 row affected (0.00 sec)

Now exit the MySQL shell by pressing CTRL+D. On the normal command line, you can import the dump file (in database folder) with the following command:

mysql -u username -p *db\_machine\_web\_controller* < db\_machine\_web\_controller.sql

* username is the username you can log in to the database with
* db\_machine\_web\_controller is the name of the freshly created database
* db\_machine\_web\_controller.sql is the data dump file to be imported, located in the current directory ( you need upload and move it)

## 6. Upload - Config source code

Copy the source code in to the web root folder /var/www/html and set permitsion with the following command:

sudo chmod 777 workon/

Open the config file in workon/config/config.php:

sudo nano workon/config/config.php

Edit the username and password you set up above:

<?php

$servername = "localhost";

$username = "root";

$password = "jungle2017";

$database = "db\_machine\_web\_controller";

?>