


CONTENTS

- Prerequisites
- Step 1 — Installing Apache and Adjusting the Firewall
- Step 2 — Installing MySQL
- Step 3 — Installing PHP
- Step 4 — Testing PHP Processing on your Web Server
- Conclusion

RELATED

How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu
[View](#) 

How To Install Rails, Apache, and MySQL on Ubuntu with Passenger
[View](#) 

// Tutorial //



How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 16.04

Published on April 21, 2016 · Updated on November 30, 2021

- Apache
- MySQL
- LAMP Stack
- PHP
- Ubuntu 16.04



By [Brennen Bearnes](#)
Developer and author at DigitalOcean.

 English 



Not using Ubuntu 16.04?

Choose a different version or distribution.

Ubuntu 16.04 ▼

Introduction

The *LAMP* stack is a set of open source software that is typically installed together to enable a server to host dynamic websites and web apps. This term is an acronym which represents the **L**inux operating system, with the **A**pache web server. The site data is stored in a **M**ySQL database, and dynamic content is processed by **P**HP.

In this guide, we'll get a LAMP stack installed on an Ubuntu 16.04 server. Ubuntu will fulfill our first requirement: a Linux operating system.

Prerequisites

Before you begin with this guide, you should have a separate, non-**root** user account with `sudo` privileges set up on your server. You can learn how to do this by completing steps 1–4 in our [initial server setup for Ubuntu 16.04](#) tutorial.

Step 1 – Installing Apache and Adjusting the Firewall

The Apache web server is among the most popular web servers in the world. It's well-documented, and has been in wide use for much of the history of the web, which makes it a great default choice for hosting a website.

We can install Apache easily using Ubuntu's package manager, `apt`. A package manager allows us to install most software pain-free from a repository maintained by Ubuntu. You can learn more about [how to use apt](#) here.

For our purposes, we can get started by typing these commands:

```
$ sudo apt-get update
$ sudo apt-get install apache2
```

Copy

Since we are using a `sudo` command, these operations get executed with **root** privileges. It will ask you for your regular user's password to verify your intentions.

Once you've entered your password, `apt` will tell you which packages it plans to install and how much extra disk space they'll take up. Press `y` and hit `ENTER` to continue, and the installation will proceed.

Setting Global ServerName to Suppress Syntax Warnings

Next, we will add a single line to the `/etc/apache2/apache2.conf` file to suppress a warning message. While harmless, if you do not set `ServerName` globally, you will receive the following warning when checking your Apache configuration for syntax errors:

```
$ sudo apache2ctl configtest
```

Copy

Output

```
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using
Syntax OK
```

Open up the main configuration file with your text edit:

```
$ sudo nano /etc/apache2/apache2.conf
```

Copy

Inside, at the bottom of the file, add a `ServerName` directive, pointing to your primary domain name. If you do not have a domain name associated with your server, you can use your server's public IP address:

Note: If you don't know your server's IP address, skip down to the section on [how to find your server's public IP address](#) to find it.

```
/etc/apache2/apache2.conf
```

```
. . .  
ServerName server_domain_or_IP
```

Save and close the file when you are finished.

Next, check for syntax errors by typing:

```
$ sudo apache2ctl configtest
```

Copy

Since we added the global `ServerName` directive, all you should see is:

```
Output  
Syntax OK
```

Restart Apache to implement your changes:

```
$ sudo systemctl restart apache2
```

Copy

You can now begin adjusting the firewall.

Adjusting the Firewall to Allow Web Traffic

Next, assuming that you have followed the initial server setup instructions to enable the UFW firewall, make sure that your firewall allows HTTP and HTTPS traffic. You can make sure that UFW has an application profile for Apache like so:

```
$ sudo ufw app list
```

Copy

```
Output  
Available applications:  
  Apache  
  Apache Full  
  Apache Secure  
  OpenSSH
```

If you look at the `Apache Full` profile, it should show that it enables traffic to ports 80 and 443:

```
$ sudo ufw app info "Apache Full"
```

Copy

```
Output  
Profile: Apache Full  
Title: Web Server (HTTP,HTTPS)  
Description: Apache v2 is the next generation of the omnipresent Apache web  
server.  
  
Ports:  
  80,443/tcp
```

Allow incoming traffic for this profile:

```
$ sudo ufw allow in "Apache Full"
```

Copy

You can do a spot check right away to verify that everything went as planned by visiting your server's public IP address in your web browser (see the note under the next heading to find out what your public IP address is if you do not have this information already):

```
http://your_server_IP_address
```

You will see the default Ubuntu 16.04 Apache web page, which is there for informational and testing purposes. It should look something like this:



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in** [/usr/share/doc/apache2/README.Debian.gz](#). Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the [manual](#) if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

If you see this page, then your web server is now correctly installed and accessible through your firewall.

Finding your Server's Public IP Address

If you do not know what your server's public IP address is, there are a number of ways you can find it. Usually, this is the address you use to connect to your server through SSH.

From the command line, you can find this a few ways. First, you can use the `iproute2` tools to get your address by typing this:

```
$ ip addr show eth0 | grep inet | awk '{ print$2; }' | sed 's/\/*$//'
```

[Copy](#)

This will give you two or three lines back. They are all correct addresses, but your computer may only be able to use one of them, so feel free to try each one.

An alternative method is to use the `curl` utility to contact an outside party to tell you how *it* sees your server. You can do this by asking a specific server what your IP address is:

```
$ sudo apt-get install curl
$ curl http://icanhazip.com
```

[Copy](#)

Regardless of the method you use to get your IP address, you can type it into your web browser's address bar to get to your server.

Step 2 – Installing MySQL

Now that we have our web server up and running, it is time to install MySQL. MySQL is a database management system. Basically, it will organize and provide access to databases where our site can store information.

Again, we can use `apt` to acquire and install our software. This time, we'll also install some other “helper” packages that will assist us in getting our components to communicate with each other:

```
$ sudo apt-get install mysql-server
```

[Copy](#)

Note: In this case, you do not have to run `sudo apt-get update` prior to the command. This is because we recently ran it in the commands above to install Apache. The package index on our computer should already be up-to-date.

Again, you will be shown a list of the packages that will be installed, along with the amount of disk space they'll take up. Enter **Y** to continue.

During the installation, your server will ask you to select and confirm a password for the MySQL “root” user. This is an administrative account in MySQL that has increased privileges. Think of it as being similar to the root account for the server itself (the one you are configuring now is a MySQL-specific account, however). Make sure this is a strong, unique password, and do not leave it blank.

When the installation is complete, we want to run a simple security script that will remove some dangerous defaults and lock down access to our database system a little bit. Start the interactive script by running:

```
$ mysql_secure_installation
```

[Copy](#)

You will be asked to enter the password you set for the MySQL root account. Next, you will be asked if you want to configure the `VALIDATE PASSWORD PLUGIN`.

Warning: Enabling this feature is something of a judgment call. If enabled, passwords which don't match the specified criteria will be rejected by MySQL with an error. This will cause issues if you use a weak password in conjunction with software which automatically configures MySQL user credentials, such as the Ubuntu packages for phpMyAdmin. It is safe to leave validation disabled, but you should always use strong, unique passwords for database credentials.

Answer **y** for yes, or anything else to continue without enabling.

```
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:
```

You'll be asked to select a level of password validation. Keep in mind that if you enter **2**, for the strongest level, you will receive errors when attempting to set any password which does not contain numbers, upper and lowercase letters, and special characters, or which is based on common dictionary words.

```
There are three levels of password validation policy:

LOW      Length >= 8
MEDIUM  Length >= 8, numeric, mixed case, and special characters
STRONG  Length >= 8, numeric, mixed case, special characters and dictionary

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 1
```

If you enabled password validation, you'll be shown a password strength for the existing root password, and asked you if you want to change that password. If you are happy with your current password, enter **n** for "no" at the prompt:

```
Using existing password for root.

Estimated strength of the password: 100
Change the password for root ? ((Press y|Y for Yes, any other key for No) : n
```

For the rest of the questions, you should press **Y** and hit the **Enter** key at each prompt. This will remove some anonymous users and the test database, disable remote root logins, and load these new rules so that MySQL immediately respects the changes we have made.

At this point, your database system is now set up and we can move on.

Step 3 – Installing PHP

PHP is the component of our setup that will process code to display dynamic content. It can run scripts, connect to our MySQL databases to get information, and hand the processed content over to our web server to display.

We can once again leverage the `apt` system to install our components. We're going to include some helper packages as well, so that PHP code can run under the Apache server and talk to our MySQL database:

```
$ sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql
```

This should install PHP without any problems. We'll test this in a moment.

In most cases, we'll want to modify the way that Apache serves files when a directory is requested. Currently, if a user requests a directory from the server, Apache will first look for a file called `index.html`. We want to tell our web server to prefer PHP files, so we'll make Apache look for an `index.php` file first.

To do this, type this command to open the `dir.conf` file in a text editor with root privileges:

```
$ sudo nano /etc/apache2/mods-enabled/dir.conf
```

It will look like this:

```
/etc/apache2/mods-enabled/dir.conf

<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>
```

We want to move the PHP index file highlighted above to the first position after the `DirectoryIndex` specification, like this:

```
/etc/apache2/mods-enabled/dir.conf
```



```
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
```

When you are finished, save and close the file by pressing **Ctrl-X**. You'll have to confirm the save by typing **Y** and then hit **Enter** to confirm the file save location.

After this, we need to restart the Apache web server in order for our changes to be recognized. You can do this by typing this:

```
$ sudo systemctl restart apache2
```

[Copy](#)

We can also check on the status of the `apache2` service using `systemctl`:

```
$ sudo systemctl status apache2
```

[Copy](#)

Sample Output

```
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Wed 2016-04-13 14:28:43 EDT; 45s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 13581 ExecStop=/etc/init.d/apache2 stop (code=exited, status=0/SUCCESS)
  Process: 13605 ExecStart=/etc/init.d/apache2 start (code=exited, status=0/SUCCESS)
    Tasks: 6 (limit: 512)
   CGroup: /system.slice/apache2.service
           └─13623 /usr/sbin/apache2 -k start
             └─13626 /usr/sbin/apache2 -k start
               └─13627 /usr/sbin/apache2 -k start
                 └─13628 /usr/sbin/apache2 -k start
                   └─13629 /usr/sbin/apache2 -k start
                     └─13630 /usr/sbin/apache2 -k start

Apr 13 14:28:42 ubuntu-16-lamp systemd[1]: Stopped LSB: Apache2 web server.
Apr 13 14:28:42 ubuntu-16-lamp systemd[1]: Starting LSB: Apache2 web server...
Apr 13 14:28:42 ubuntu-16-lamp apache2[13605]: * Starting Apache httpd web server apache2
Apr 13 14:28:42 ubuntu-16-lamp apache2[13605]: AH00558: apache2: Could not reliably determine
Apr 13 14:28:43 ubuntu-16-lamp apache2[13605]: *
Apr 13 14:28:43 ubuntu-16-lamp systemd[1]: Started LSB: Apache2 web server.
```

Install PHP Modules

To enhance the functionality of PHP, we can optionally install some additional modules.

To see the available options for PHP modules and libraries, you can pipe the results of `apt-cache search` into `less`, a pager which lets you scroll through the output of other commands:

```
$ apt-cache search php- | less
```

[Copy](#)

Use the arrow keys to scroll up and down, and **q** to quit.

The results are all optional components that you can install. It will give you a short description for each:

```
libnet-libidn-perl - Perl bindings for GNU Libidn
php-all-dev - package depending on all supported PHP development packages
php-cgi - server-side, HTML-embedded scripting language (CGI binary) (default)
php-cli - command-line interpreter for the PHP scripting language (default)
php-common - Common files for PHP packages
php-curl - CURL module for PHP [default]
php-dev - Files for PHP module development (default)
php-gd - GD module for PHP [default]
php-gmp - GMP module for PHP [default]
...
:
```

To get more information about what each module does, you can either search the internet, or you can look at the long description of the package by typing:

```
$ apt-cache show package_name
```

[Copy](#)

There will be a lot of output, with one field called `Description-en` which will have a longer explanation of the functionality that the module provides.

For example, to find out what the `php-cli` module does, we could type this:

```
$ apt-cache show php-cli
```

[Copy](#)

Along with a large amount of other information, you'll find something that looks like this:

Output

```
...
Description-en: command-line interpreter for the PHP scripting language (default)
This package provides the /usr/bin/php command interpreter, useful for
testing PHP scripts from a shell or performing general shell scripting tasks.
.
PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used
open source general-purpose scripting language that is especially suited
for web development and can be embedded into HTML.
.
This package is a dependency package, which depends on Debian's default
PHP version (currently 7.0).
...
```

If, after researching, you decide you would like to install a package, you can do so by using the `apt-get install` command like we have been doing for our other software.

If we decided that `php-cli` is something that we need, we could type:

```
$ sudo apt-get install php-cli
```

[Copy](#)

If you want to install more than one module, you can do that by listing each one, separated by a space, following the `apt-get install` command, like this:

```
$ sudo apt-get install package1 package2 ...
```

[Copy](#)

At this point, your LAMP stack is installed and configured. We should still test out our PHP though.

Step 4 – Testing PHP Processing on your Web Server

In order to test that our system is configured properly for PHP, we can create a very basic PHP script.

We will call this script `info.php`. In order for Apache to find the file and serve it correctly, it must be saved to a very specific directory, which is called the *web root*.

In Ubuntu 16.04, this directory is located at `/var/www/html/`. We can create the file at that location by typing:

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

[Copy](#)

This will open a blank file. We want to put the following text, which is valid PHP code, inside the file:

info.php

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

When you are finished, save and close the file.

Now we can test whether our web server can correctly display content generated by a PHP script. To try this out, we just have to visit this page in our web browser. You'll need your server's public IP address again.

The address you want to visit will be:

```
http://your_server_IP_address/info.php
```

The page that you come to should look something like this:

PHP Version 7.0.4-7ubuntu1	
System	Linux ubuntu-16-lamp 4.4.0-12-generic #28-Ubuntu SMP Wed Mar 9 00:33:55 UTC 2016 x86_64
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-mcrypt.ini, /etc/php/7.0/apache2/conf.d/20-mysqli.ini, /etc/php/7.0/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS
PHP Extension Build	API20151012,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
Intl Support	enabled

This page basically gives you information about your server from the perspective of PHP. It is useful for debugging and to ensure that your settings are being applied correctly.

If this was successful, then your PHP is working as expected.

You probably want to remove this file after this test because it could actually give information about your server to unauthorized users. To do this, you can type this:

```
$ sudo rm /var/www/html/info.php
```

Copy

You can always recreate this page if you need to access the information again later.

Conclusion

Now that you have a LAMP stack installed, you have many choices for what to do next. Basically, you've installed a platform that will allow you to install most kinds of websites and web software on your server.

As an immediate next step, you should ensure that connections to your web server are secured, by serving them via HTTPS. The easiest option here is to [use Let's Encrypt](#) to secure your site with a free TLS/SSL certificate.

Some other popular options are:

- [Install Wordpress](#) the most popular content management system on the internet.
- [Set Up PHPMyAdmin](#) to help manage your MySQL databases from web browser.
- [Learn more about MySQL](#) to manage your databases.

- [Learn how to use SFTP](#) to transfer files to and from your server.

Spin up a Linux virtual machine with Apache, MySQL, and PHP pre-configured and attached in one simple click with DigitalOcean. Let us spin up a LAMP stack server for you in seconds, so you can focus on building a great application.

[Learn more here](#) →

Want to learn more? Join the DigitalOcean Community!

Join our DigitalOcean community of over a million developers for free! Get help and share knowledge in our Questions & Answers section, find tutorials and tools that will help you grow as a developer and scale your project or business, and subscribe to topics of interest.

[Sign up now](#) →

About the authors



[Brennen Bearnes](#) Author

Developer and author at DigitalOcean.

Still looking for an answer?

[Ask a question](#)

[Search for more help](#)

Was this helpful?

[Yes](#)

[No](#)



Comments

10 Comments

B *I* U H_1 H_2 H_3 “”



Leave a comment ...



This textbox defaults to using **Markdown** to format your answer.

You can type **!ref** in this text area to quickly search our full set of tutorials, documentation & marketplace offerings and insert the link!

[Sign In or Sign Up to Comment](#)

[SamiF](#) • August 28, 2018



This tutorial used to work 100% and it was a great reference guide until a recent change to MySQL defaults. As of 5.7, now root is set to auth_socket authentication by default, so this will let you login from SSH/Terminal but not through any app/script/website following the tutorial.

Look at the tutorial for installing Ubuntu on 18.04 for instructions on how to make the changes to MySQL to allow root to login inside your PhpMyAdmin/Adminer/Scripts/etc:

<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04>

[Reply](#)

[araminimichael](#) • August 21, 2018



Absolument excellent. J'ai suivi le tutoriel à a Z sans soucis. Tous fonctionnait jusqu'à ce que je teste letsencrypt. Mais pour toute cette partie sur cette page aucun problème ça fonctionne nickel sur orangepi zéro (armbian Ubuntu 16.04 Xenical) du premier coups sans prise de tête. Vraiment excellent. Clairement expliqué pas à pas. Un tuto de grande qualité. Merci

[Reply](#)

[curteneyarellano](#) • November 28, 2017



What if I don't want to secure the mysql that I'd created but I want to change my password? What code should I input?

[Show replies](#) ▾ [Reply](#)

[fourwadsmart](#) • August 3, 2017



Great tutorial. One question, can I rename the info file to something rare instead of removing it? I just would like to keep it.

[Show replies](#) ▾ [Reply](#)

[leaguedraven6](#) • March 21, 2017



This is the best guide on the internet ! I just switched linux full time, was looking for lamp installation on net. I liked the process, and that I could see what I configure, and why ...

Everything worked for the first time, no errors, thanks a lot :)

[Reply](#)

[omissoni](#) • August 3, 2016



For very lazy sysadmins like me:

sudo tasksel

then choose LAMP-Server. Drink a coffee and enjoy

[Show replies](#) ▾ [Reply](#)

[EatingGrayAngler](#) • July 30, 2022



mysql_secure_installation gets stuck for me... <https://share.getcloudapp.com/L1uXw1RJ> ... always this error: ... Failed! Error: SET PASSWORD has no significance for user 'root'@'localhost' as the authentication method used doesn't store authentication data in the MySQL server. Please consider using ALTER USER instead if you want to change authentication parameters.

Guess I gotta leave and restart

[Reply](#)

[Sebastien-Remy](#) • May 17, 2022



I use this to set root@localhost password :

```
$sudo mysql
mysql>ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by 'mynewpa
mysql>exit
```

[Reply](#)

[Krystian Broniszewski](#) • February 3, 2021



Everything is working, but your article in my opinion is to long. All article maybe can be faster and light, but everything is working

[Reply](#)

[gitdemon](#) • September 16, 2020



I'm following this tutorial in 11/2020 and thought I'd share some updates:

1. MySQL Instead of `mysql_secure_installation`, I ran `sudo mysql_secure_installation`. I then proceeded to configure the password that is now know as **VALIDATE PASSWORD COMPONENT**. This is when you use add your pw.

2. PHP To install PHP, use this `sudo apt-get install php libapache2-mod-php php-mysql`.
The `php-mcrypt` has been deprecated since PHP7.2

[Reply](#)

Load More Comments



This work is licensed under a Creative Commons Attribution-NonCommercial- ShareAlike 4.0 International License.

Try DigitalOcean for free

Click below to sign up and get **\$200 of credit** to try our products over 60 days!

Sign up →

Popular Topics

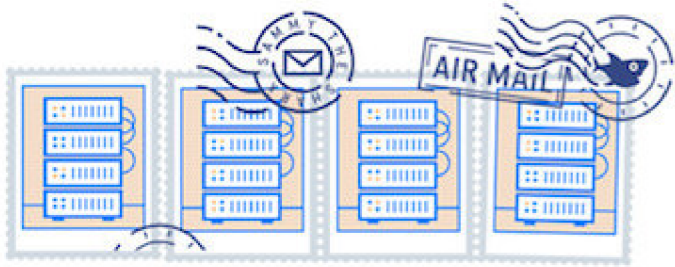
- Ubuntu
- Linux Basics
- JavaScript
- React
- Python
- Security
- MySQL
- Docker
- Kubernetes
- Browse all topic tags

[Free Managed Hosting →](#)

[All tutorials →](#)

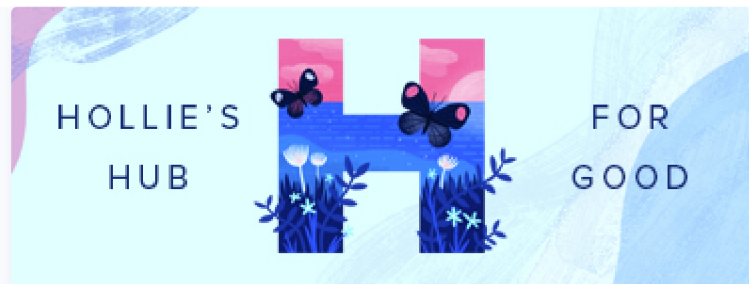
Questions

- Q&A Forum
- Ask a question
- DigitalOcean Support



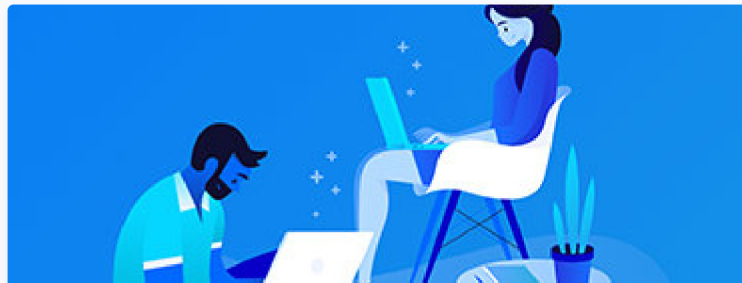
GET OUR BIWEEKLY NEWSLETTER

Sign up for Infrastructure as a Newsletter.



HOLLIE'S HUB FOR GOOD

Working on improving health and education, reducing inequality, and spurring economic growth? We'd like to help.



BECOME A CONTRIBUTOR

You get paid; we donate to tech nonprofits.

[Featured on Community](#) [Kubernetes Course](#) [Learn Python 3](#) [Machine Learning in Python](#) [Getting started with Go](#) [Intro to Kubernetes](#)

[DigitalOcean Products](#) [Virtual Machines](#) [Managed Databases](#) [Managed Kubernetes](#) [Block Storage](#) [Object Storage](#) [Marketplace](#) [VPC](#) [Load Balancers](#)

Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you're running one virtual machine or ten thousand.

[Learn More](#)



Company

[About](#)
[Leadership](#)
[Blog](#)
[Careers](#)
[Customers](#)

Products

[Products Overview](#)
[Droplets](#)
[Kubernetes](#)
[App Platform](#)
[Functions](#)

Community

[Tutorials](#)
[Q&A](#)
[CSS-Tricks](#)
[Write for DOnations](#)
[Currents Research](#)

Solutions

[Website Hosting](#)
[VPS Hosting](#)
[Web & Mobile Apps](#)
[Game Development](#)
[Streaming](#)

Contact

[Support](#)
[Sales](#)
[Report Abuse](#)
[System Status](#)
[Share your ideas](#)

Partners	Cloudways	Hatch Startup Program	VPN
Channel Partners	Managed Databases	deploy by DigitalOcean	SaaS Platforms
Referral Program	Spaces	Shop Swag	Cloud Hosting for Blockchain
Affiliate Program	Marketplace	Research Program	Startup Resources
Press	Load Balancers	Open Source	
Legal	Block Storage	Code of Conduct	
Security	Tools & Integrations	Newsletter Signup	
Investor Relations	API	Meetups	
DO Impact	Pricing		
	Documentation		
	Release Notes		
	Uptime		

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000

