You are here

Step-by-step Installation Guide for Ubuntu

Note: You are currently viewing documentation for Moodle 3.1. Up-to-date documentation for the latest stable version of Moodle is probably available here: ***Step-by-step Installation Guide for Ubuntu.***

Note: This document is about installing Moodle 3.1 in an Ubuntu 16.04 server with PHP 7.0. The Moodle 3.0 version of this document covers Moodle 3.0 with Ubuntu 14.04 and PHP 5.

Contents

Step 1: Install Ubuntu 16.04 LTS

Why we prefer Ubuntu server over Ubuntu desktop

- Most IT professionals prefer to use a Command Line Interface (CLI) server, because it is safer and less prone to hacking.
- Amateur users might find it easier to use a graphical (desktop) interface.
- If you will only be using your Moodle server for local, experimental purposes, you might prefer to install the desktop (64 bits preferred) version of Ubuntu.
- If you install a CLI only server and later regret it, you can easily add a graphical desktop:
  - Even though it is not recommended by most experts, you could install a Graphical User Interface (desktop) by issuing the command 'sudo tasksel' and choosing (with the Space Bar) to install 'Ubuntu desktop'.

Why we prefer (or don't prefer) Ubuntu 16.04 over Ubuntu 14.04

- Both packages are LTS (Long Term Service) releases.
- Ubuntu Server 16.04LTS has some differences from 14.04LTS, mainly PHP 7.0.
- Moodle 3.01 onwards can use PHP 7.0
- This document has been updated to reflect those changes.
- But beware that if you are using external authentication or enrollment plug-ins, please note that at the moment of Moodle 3.0.1 release (December 2015) neither MSSQL nor SQLSRV are available under php7. So, anybody using SQL*Server as primary database or remote auth/enrol plugin should not move to PHP 7 at all ! Also, depending of your configuration, some extensions (memcached can be installed, redis, mongodb, xmlrpc...) maybe missing or work in progress for your distribution, triple check the exact availability for your OS.

Procedure

Server Edition amd64 preferred

http://www.ubuntu.com/download

- Ubuntu Server 16.04LTS amd64 has all the required packages.

- Ubuntu Server 16.04LTS has some minor differences from 14.04LTS, mainly PHP 7.0. This document has been updated to reflect those changes.
- You can use either VI (lightweight editor) or VIM (heavyweight editor), however, if you wish to use VIM you will need to install it

```
sudo apt-get install vim
```

- VI or VIM Commands

To edit a file press "Insert" Key
To finish editing press "Esc" Key
To write the file press ":w"
To Exit the editor press ":q"
You can also write and quit ":wq"

- In Ubuntu, the standard user, the account you created during the install, does not have rights to install/write to many of the directories. In the below tutorial we will be using the term "sudo" which stands for "super user do" before most of the commands.

Step 2: Install Apache/MySQL/PHP

Note: Moodle 3.0.1 introduced support for PHP 7.0 and we will be using PHP 7.0 in this tutorial

Open up Terminal and install the following;

- :22 in the Video

```
sudo apt-get update
```

```
sudo apt-get install apache2 mysql-client mysql-server php7.0 libapache2-mod-php7.0
```

'**It will prompt you to set the root password for mysql - please, please my dear friends, WRITE IT DOWN and spare yourself some grief, you will need it in step 6.**

Note: Even though it is not recommended by most experts, you could install a Graphical User Interface (desktop) by issuing the command 'sudo tasksel' and choosing (with the Space Bar) to install 'Ubuntu desktop'

Step 3: Install Additional Software

- 1:10 in the video

```
sudo apt-get install graphviz aspell php7.0-pspell php7.0-curl php7.0-gd php7.0-intl php7.0-mysql php7.0-
```

Restart Apache so that the modules are loaded correctly

```
sudo service apache2 restart
```

We will be using Git to install/update the Moodle Core Application

```
sudo apt-get install git-core
```

Step 4: Download Moodle

- 1:55 in the video

Setup your local repository and download Moodle, We will use /opt for this installation.

- Git is what is called a "version control system". By using git it will much easier down the road to update the moodle core application. Within Step 5 there is a little more detail on why we put the moodle core application code in the /opt directory.

```
cd /opt
```

Download the Moodle Code and Index

```
sudo git clone git://git.moodle.org/moodle.git
```

Change directory into the downloaded Moodle folder

```
cd moodle
```

Retrieve a list of each branch available

```
sudo git branch -a
```

Tell git which branch to track or use

```
sudo git branch --track MOODLE_31_STABLE origin/MOODLE_31_STABLE
```

Finally, Check out the Moodle version specified

```
sudo git checkout MOODLE_31_STABLE
```

Step 5: Copy local repository to /var/www/html/

- 3:05 in the video

```
sudo cp -R /opt/moodle /var/www/html/
```

```
sudo mkdir /var/moodledata
```

```
sudo chown -R www-data /var/moodledata
```

```
sudo chmod -R 777 /var/moodledata
```

```
sudo chmod -R 0755 /var/www/html/moodle
```

- Explanation:

Since we setup a local repository in the previous step, you will copy it to your webroot after any updates and making changes. Having your local repository outside of the webroot, like we have in /opt, you will be able to prepare and stage your upgrades in a more efficient manner. For example, you want to make some changes or add some plug-ins, you would download the plugin and copy it to your local moodle repository. After you have added the plug-in and any other changes you might have made you will need to edit the file located in /opt/moodle/.git/info/exclude. Within that file you want to tell git which files/folders to exclude when it pulls down the updates when you run your next "sudo git pull". An example entry would be the certificate mod located in /opt/moodle/mod/certificate so within the exclude file you want to add "/mod/certificate" below the last comments. You would add additional entries, 1 per line, for each plug-in or file you might have changed. If I were to change the favicon.ico file you would just add "favicon.ico" to the exclude file. Now when you run "sudo git pull" to update moodle to the latest version it will ignore those files and directories and just update the core moodle code. Before copying to your webroot to upgrade you want to make sure and download and copy over the latest versions of the plug-ins you might have added.

Step 6: Setup MySQL Server

- 4:45 in the video

First we need to change the default storage engine to innodb and change the default file format to Barracuda, this is a new setting compared to previous versions. You also need to set innodb_file_per_table in order for Barracuda to work properly. Ref: https://dev.mysql.com/doc/refman/5.6/en/innodb-compression-usage.html

- You should not need to make innodb the default storage engine anymore, the latest version of Moodle will select it automatically during install. It is always a good idea to make it default anyway. You do however need to set the default file format!

- If you chose to use VIM instead please substitute vi for vim

```
sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
```

Scroll down to the [mysqld] section and under Basic Settings add the following line under the last statement. if you want to add you have to press the "insert" button on your keyboard. this is usually above the "delete" button. this allows you to add some text.

```
default_storage_engine = innodb
```

```
innodb_file_per_table = 1
```

```
innodb_file_format = Barracuda
```

In order to save my.cnf using the editor, press the Esc (Escape) key, type the following in sequence which will save :w then close the editor :q

```
:w
```

```
:q
```

Restart MySQL Server for changes to take affect

```
sudo service mysql restart
```

Now we need to create the Moodle database and the Moodle MySQL User with the correct permissions

Use the password you created in step 1

```
mysql -u root -p
```

mysql>

```
CREATE DATABASE moodle DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

Where it says "moodledude" and "passwordformoodledude" you should change to the username and password of your choosing.

mysql>

```
create user 'moodledude'@'localhost' IDENTIFIED BY 'passwordformoodledude';
```

mysql>

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,CREATE TEMPORARY TABLES,DROP,INDEX,ALTER ON moodle.* TO moodledu
```

mysql>

```
quit;
```

Note - If you care using MySQL 5.6+ and when you issue the create user and get an error about the password hash you need to adjust the password to use the hash value

You can get this by following the below

mysql>

```
SELECT password('passwordformoodledude');
```

This will print the hash of the password like *AD51BAFB2GD003D3480BCED0DH81AB0BG1712535, you will want to use this in the IDENTIFIED BY ' part

Step 7: Complete Setup

- 6:40 in the video
- Note - If you are not comfortable using terminal to create the config.php file that needs to be created when going through the installer, you should temporarily make the webroot writable by doing the following:

```
sudo chmod -R 777 /var/www/html/moodle
```

After you have ran the installer and you have moodle setup, you NEED to revert permissions so that it is no longer writable using the below command.

```
sudo chmod -R 0755 /var/www/html/moodle
```

Open your browser and go to http://IP.ADDRESS.OF.SERVER/moodle

Follow the prompts:

Change the path for moodledata

/var/moodledata

Database Type

Choose: mysqli

Database Settings

Host server: localhost

Database: moodle

User: moodledude (the user you created when setting up the database)

Password: passwordformoodledude (the password for the user you created)

Tables Prefix: mdl_

Environment Checks

This will indicate if any elements required to run moodle haven't been installed.

Next next next...

follow prompts and confirm installation

Create a Site Administrator Account

Create your moodle user account which will have site administrator permissions.

The password you select has to meet certain security requirements.

Installation Complete

Congrats! You can now start using Moodle!

Don't Forget

If you made the webroot writable, revert permissions

```
sudo chmod -R 0755 /var/www/html/moodle
```

System Paths After Install

- 10:05 in the video

After installing Moodle you should set the system paths, this will provide better performance VS not setting them. Each entry in Moodle will have it's explanation.

Navigate, on the moodle webpage, to Site Administration > Server > System Paths

Input the following;

Path to Du: /usr/bin/du

Path to Apsell: /usr/bin/aspell

Path to dot: /usr/bin/dot

Save Changes

- Optional if you do not already have an AntiVirus Solution

We also installed ClamAV in Step 3 so we need to set the path in Moodle

1st Create the Quarantine Directory

```
sudo mkdir /var/quarantine
```

Change Ownership

```
sudo chown -R www-data /var/quarantine
```

Navigate to Site Administration > Security > Anti-Virus

Check "Use ClamAV on uploaded files"

ClamAV Path : /usr/bin/clamscan

Quarantine Directory : /var/quarantine

Save Changes

Suggestions: Enable Zend OpCache/Change Document Root

- 11:00 in the video
- Since we have installed Ubuntu Server 14.04LTS, we can use the built-in PHP OPcache, https://docs.moodle.org/26/en/OPcache

Within the link above, https://docs.moodle.org/26/en/OPcache add the recommended settings to your 05-opcache.ini file. Again, substitute vi with vim and remember to use the correct key squences from the introduction.

```
sudo vi /etc/php5/apache2/conf.d/05-opcache.ini
```

NOTE: In Ubuntu 16.04 opcache.ini is located in:

```
/etc/php/7.0/mods-available/opcache.ini
```

Restart Apache for changes to take affect.

```
sudo service apache2 restart
```

That's it for the Zend OpCache!

You can also install a GUI to view the status of your Zend OpCache, not recommended on production servers.

```
cd /var/www/html/moodle/
```

Download the PHP Script to your Moodle directory, you should also add this file to /opt/moodle/.git/info/exclude file so it does not get removed when upgrading your installation.

```
sudo wget https://github.com/rlerdorf/opcache-status/blob/master/opcache.php
```

Visit http://ip.address.of.server/moodle/opcache.php

If you do not want your end users to type http://yourserver/moodle and just want them to navigate to http://youserver you will need to edit the site configuration for Apache which will tell Apache to use the /var/www/html/moodle as the root directory and not /var/www/html

Open up the Apache sites config and change the document root

```
sudo vi /etc/apache2/sites-available/000-default.conf
```

On the line where DocumentRoot is;

Change From: DocumentRoot /var/www/html

Change To: DocumentRoot /var/www/html/moodle

```
:w
```

```
:q
```

Restart Apache for changes to take affect.

```
sudo service apache2 restart
```

Important note!

If you have already installed Moodle then you should make the below changes.

Editing config.php for moodle

In the installation instructions, one of the suggested settings for 'webroot' is 'localhost'. This is fine if all you want to do is some local testing of your new Moodle installation. If, however, you want to view your new installation from another machine on the same local area network, or view your site on the Internet, you will have to change this setting:

For local testing, 'localhost' is fine for the webroot ($CFG->wwwroot in config.php). If you want to test your site from other machines on the same local area network (LAN), then you will have to use the private ip address of the serving machine, (e.g. 192.168.1.2/moodle) or the network name of the serving computer (e.g. network_name_of_serving_machine/moodle) as the web root. Depending on your LAN setup, it may be better to use the network name of the computer rather than its (private) ip address, because the ip address can and will change from time to time. If you don't want to use the network name, then you will have to speak to your network administrator and have them assign a permanent ip address to the serving machine. Finally, if you want to test your new installation across the internet, you will have to use either a domain name or a permanent (public) ip address/moodle as your web root. To handle both types of access, see masquerading.

Edit config.php for Moodle

cd /var/www/html/moodle sudo vim config.php

Hit the "insert" button on your keyboard, make then changes you need to make. Then press "escape" and type the following in to quit and to save changes (excluding quotation marks): ":wq"

Under $CFG->wwwroot change to http://ip.address.of.server instead of http://ip.address.of.server/moodle

Youtube Video

https://www.youtube.com/watch?v=H5vAzBrRxzI

Hosting several Moodle branches in one Ubuntu server

- This is very useful for the language pack maintainers to test translations in several Moodle branches.
- It is also very useful for developers to test their plugins in different Moodle branches.
- Just create a folder for each instance inside the web folder and that would be enough.
- To access the sites you only need to add the folder to localhost URL: http://localhost/moodle31
- You can have an instance for each version from 1.9 to 3.1

- You do need a separate data folder for each instance and a separate database (You can use phpmyadmin to set your database, but that's not necessary), add each instance in its own folder, and carry on as above. You can also host another service (eg, Mahara) in it's separate folder.

Example 1

- So, one example folder tree on one Linux laptop (an actual server would be more) may look something like:

```
var
```

--www

```
   ----maharadata
   ----moodlecleandata
   ----moodlestabledata
   ----moodlemasterdata
   ----moodletestingdata
   ----uswmoodledata
   ----html
       ------mahara
       ------moodleclean
       ------moodlestable
       ------moodlemaster
       ------moodletesting
       ------uswmoodle
```

Example 2

- Have several sandboxed Moodles on a single (CentOS X) server all of different versions .. only the ones that are supported for security fixes and above - 2.7,2.8,2.9,3.0, and now a 3.1. Pretty much 'stock' Moodles with only occasional addons, etc. for testing.
- All have their separate code and data directories as well as their separate DB's.

- Hint: install and maintain them all with git ... even if you don't prefer/like command line, that is by far the most efficient way to update and/or upgrade a site.

```
/var/www/html/moodle27/version.php:$release  = '2.7.14 (Build: 20160509)'
/var/www/html/moodle28/version.php:$release  = '2.8.12 (Build: 20160509)'
/var/www/html/moodle29/version.php:$release  = '2.9.6+ (Build: 20160520)'
/var/www/html/moodle30/version.php:$release  = '3.0.4+ (Build: 20160603)'
/var/www/html/moodle31/version.php:$release  = '3.1+ (Build: 20160603)'
```

- The git -b command locks a site into the version provided with the rest of the git command ... for example, installing the 3.1, which is a long term support version, installed with git -b option. Don't plan on upgrading nor testing upgrades with that one.

```
git clone -b MOODLE_31_STABLE git://git.moodle.org/moodle.git moodle31
```

- All the other moodles I have on that server have been installed via git

```
git clone git://git.moodle.org/moodle.git [nameofdir]
```

- then from nameofdir

```
git branch --track MOODLE_2#_STABLE origin/MOODLE_2#_STABLE
git checkout MOODLE_2#_STABLE
```

- 2# is the version number.
- That allows one to march that moodle upwards ... higher branch(es). So one can test an upgrade (as opposed to an 'update').
- This second method 'gits' more code and backups will range in the 5+ Meg range due to all the older version git stuff The 3.1 much less (restricted to 3.1 branch):
- 545M ./moodle296-code-20160604145012.tar
- 193M ./moodle31+-code-2016060883737.tar

Retrieved from "https://docs.moodle.org/31/en/index.php?title=Step-by-step_Installation_Guide_for_Ubuntu&oldid=125453"

Category: Installation

- This page was last modified on 27 September 2016, at 12:21.
- Content is available under GNU General Public License unless otherwise noted.