

This document contains instructions which should help with setting up a local stack and Drupal on Linux Mint 17:

Setting up a mono- site install of drupal: steps 1- 23;
Setting up a multi- site install of drupal: steps 24- 34;
Importing a built site into an existing multi- site install: steps 35- 44;
Setting up the Backup And Migrate module: step 45;
Working off line: step 46.

1. First download and install LAMP server by opening a command prompt and typing:

```
sudo apt-get install lamp-server^
```

2. Download and install PhPMysqladmin:

```
sudo apt-get install phpmyadmin
```

Note: during the installation of this package as well as the previous one You will be prompted to enter the database password so don't hesitate to do that if You wish Your database to be protected by one.

3. Download a distribution of Drupal of Your choice and save it somewhere on the hard drive;
4. Since Drupal 7 displays an error message notifying the user that PHP extensions are disabled on Linux Mint 17 with KDE desktop, also run the following command:

```
sudo apt-get install php5-gd php5-db php5-mysql
```

5. By now everything necessary to run and configure a local server should be installed. The first thing to do is creating a directory which will become the root directory of Drupal. For instance:

```
sudo mkdir /var/www/drupal7
```

6. Ensure that the directory can be written into by the current user:

```
sudo chown -R $USER:$USER /var/www/drupal7
```

7. Copy the default configuration file in the sites-available folder as a file with an appropriate name:

```
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/001-drupal7.conf
```

Open the new file in a text editor of Your choice:

```
sudo nano /etc/apache2/sites-available/001-drupal7.conf
```

And then edit it to look like this:

```
<VirtualHost *:80>
    ServerAdmin admin@localhost
    ServerName drupal7
    ServerAlias drupal7
    DocumentRoot /var/www/drupal7
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Save the file to disk.

Note: In this instance the name and the alias of the virtual server are identical, but this doesn't necessarily need to be so.

8. Drupal's Clean URLs feature will require opening the `apache2.conf` file in a text editor:

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

, navigating to the section that relates to `/var/www` directory (or creating a new section relating to a separate directory) and changing the line:

```
AllowOverride None
```

to:

```
AllowOverride All
```

Note: Clean URLs feature will also take enabling the rewrite module, but this will be done later.

9. Open the `hosts` file in a text editor:

```
sudo nano /etc/hosts
```

and add the following line:

```
127.0.0.1    drupal7
```

10. Enable the SSL security protocol for your new site by copying the existing file to an appropriately named file:

```
sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/drupal7-ssl.conf
```

, open the new file in a text editor:

```
sudo nano /etc/apache2/sites-available/drupal7-ssl.conf
```

and make the line pointing to the default directory:

```
DocumentRoot /var/www/html
```

point to the new directory You just created:

```
DocumentRoot /var/www/drupal7
```

and save the file to disk;

11. Enable Your new site:

```
sudo a2ensite drupal7
```

Note: This command creates a symbolic link in the directory /etc/apache2/sites-enabled and, if one is curious enough, one can navigate to that directory and check whether a file named 001-drupal7.conf appeared there. To remove the link, which is to say to disable the new site, type the command which will remove the symbolic link:

```
sudo a2dissite drupal7
```

Creating the symbolic link by using the 'ln -s' command is mentioned in one of the tutorials linked at the bottom of this document used to work but no longer does;

12. Enable the Rewrite module necessary for Drupal's Clean URLs feature:

```
sudo a2enmod rewrite
```

13. Restart the Apache service:

```
sudo service apache2 restart
```

14. It is now time to test the Apache server as well as PHP. Open Your browser and type “localhost” in the address bar. If everything went well You should see the “It works! Yada yada...” message:



ubuntu

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.

15. Create a test.php document in Your new site's root directory:

```
sudo nano /var/www/test.php
```

, copy the following line into it:

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

and save it to disk;

16. Point Your browser to the new document by typing “drupal7/test.php”, or the appropriate address if a different folder and/ or site name was chosen, in it's adress bar. If everything went well the page containing the information regarding the local installation of php should be displayed:



System	Linux leo-MS-7599 3.13.0-37-generic #64-Ubuntu SMP Mon Sep 22 21:28:38 UTC 2014 x86_64
Build Date	Mar 16 2015 20:43:56
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded	/etc/php5/apache2/php.ini

Note: If everything went well, the Apache server and php have been properly configured to use the /var/www/drupal7 directory as the virtual host. Off course, one could use the pre-configured /var/www/html directory as the virtual host in which case steps 5- 16 can be skipped.

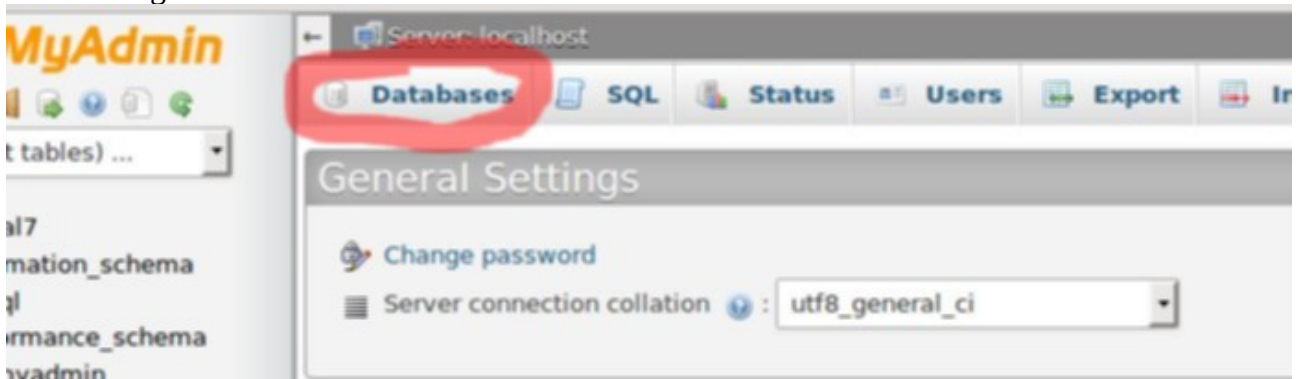
17. In order to create a database Drupal will use, type the following in the command prompt:

```
mysqladmin -u root -p create drupal7
```

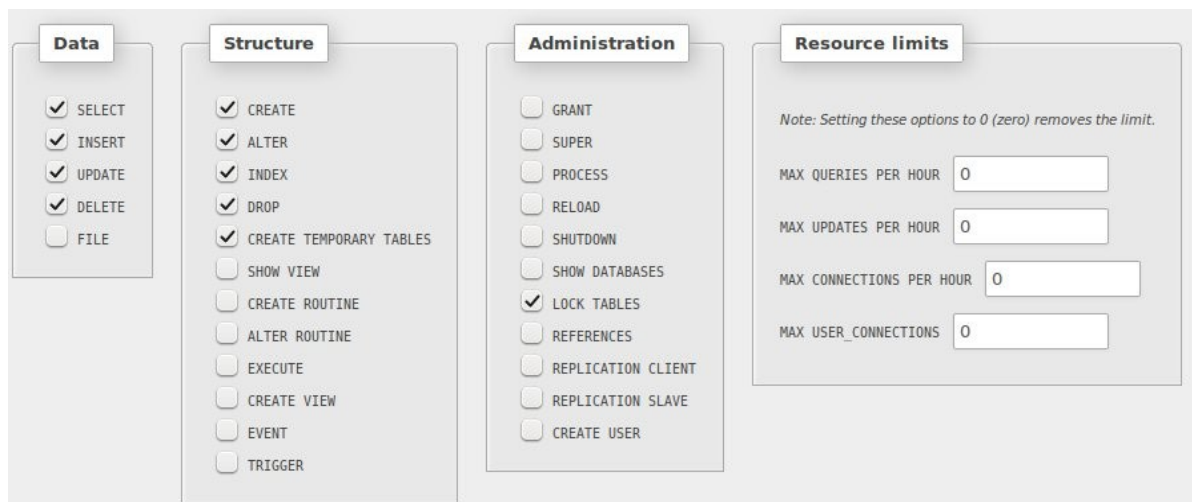
, wherein 'drupal7' is the name of the new database which can, off course, be named differently. Don't forget to enter Your password if You set one;

18. Open your browser and type 'localhost/phpmyadmin' in the address bar. The first time You log into the phpmyadmin applet, You will probably need to log in as the 'root' user, so just

type the password and proceed. Once logged in, click the 'Databases' button marked on the image below:



Click the 'Edit privileges' button to the right of Your new database's name, click 'Add user', enter the new user's name, select 'localhost' for a local install of Drupal and set a password. Scroll down and edit user privileges to reflect the image below:



Click the 'Go' button at the bottom, wait for the process to complete and log out of the phpmyadmin applet or just switch the browser off;

Note: If an error message is perceived at the bottom of the screen when phpmyadmin applet starts notifying the user that the mcrypt extension is missing, it may be worthwhile to enable it by typing the following in the command prompt:

```
sudo php5enmod mcrypt  
sudo service apache2 restart
```

If this doesn't help, then the extension can be installed:

```
sudo apt-get install mcrypt php5-mcrypt; php5enmod mcrypt; service apache2 restart
```

19. Unpack the Drupal install archive and copy the contents of the directory containing the files to the /var/www/drupal7 directory. Dragging and dropping the contents from one open window of a file manager, such as Dolphin, to another will work just fine, but one can use the command prompt as well;

20. If the file settings.php in the drupal/sites/default directory doesn't exist, copy the default.settings.php file as the settings.php file, for instance:

```
cp /var/www/drupal7/sites/default/default.settings.php /var/www/drupal7/sites/defaultsettings.php
```

21. It's time to set the permissions for the drupal/sites/default and drupal/sites/default/files directories as well as for the drupal/sites/settings.php file. The following scheme is suggested for site development, but reading the article that explains securing Drupal websites linked at the bottom of this document would be well advised:

```
sudo chown www-data /var/www/drupal7/sites/default
sudo chmod 770 /var/www/drupal7/sites/default
sudo chown www-data /var/www/drupal7/sites/default/settings.php
sudo chmod 770 /var/www/drupal7/sites/default/settings.php
sudo chown www-data /var/www/drupal7/sites/default/files
sudo chmod 770 /var/www/drupal7/sites/default/files
```

Ownership of relevant directories and files has been assigned to Drupal (chown command), Drupal and the user were given full access to relevant directories and files, while no access at all has been given to the rest of the world. Drupal will take it from there;

22. Open the web browser and type 'drupal7' in the address bar. The 'Install.php' script will run and, by following the on- screen prompts and entering the data as required, the process of installing Drupal for a single site will end. Run 'Cron' and 'Clean URLs' tests in Drupal. Both test should yield positive results;
23. Those content with working on a single website, which will be stored in the /var/www/drupal7/sites/default folder and use the 'drupal7' in the instance presented here, may proceed with creating their website. Those interested in working on several websites with a single install of Drupal can read on;
24. Another Drupal site on a local stack will need another virtual host, another site folder and another database. **The following steps will need to be repeated for every site one may wish to develop under a single install of Drupal.** The first thing to do is to add another virtual host definition to the 001-Drupal7.conf file created. Therefore, open the file in a text editor:

```
sudo nano /etc/apache2/sites-available/001-drupal7.conf
```

and add the following lines:

```
<VirtualHost *:80>
    ServerAdmin admin@localhost
    ServerName zombieattackproductions.net
    ServerAlias zombiesattack
    DocumentRoot /var/www/drupal7
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

ServerName entry contains the server name. In the tutorials linked at the end of this document example.com is used as the server name and example as the server alias, which is shorter and more neutral, but one could also go with a politically incorrect server name and server alias like flavoursofsocialism.org and 88. Save the file;

25. Open the Hosts file in a text editor:

```
sudo nano /etc/hosts
```

and add the following line:

```
127.0.0.1    zombieattackproductions.net
```

26. Restart the server:

```
sudo service apache2 restart
```

27. Create a new site directory in the drupal/sites directory, in the instance presented here:

```
mkdir /var/www/drupal7/sites/zombie
```

28. Copy the contents of the drupal/sites/default folder over to the drupal/sites/new_site folder, new-site standing for zombie in this instance. A file manager will do just fine;

29. Set permissions on the new folder and relevant files and folders within:

```
sudo chown www-data /var/www/drupal7/sites/zombie
sudo chmod 770 /var/www/drupal7/sites/zombie
sudo chown www-data /var/www/drupal7/sites/zombie/settings.php
sudo chmod 770 /var/www/drupal7/sites/zombie/settings.php
sudo chown www-data /var/www/drupal7/sites/zombie/files
sudo chmod 770 /var/www/drupal7/sites/zombie/files
```

30. Create another database for the zombieattackproductions.net website by typing:

```
mysqladmin -u root -p create zombiedb
```

The name of the new database will be zombiedb and this can be checked by logging into the phpmyadmin applet;

31. Edit the new site's settings.ini file in order to point it to the new database. In this instance:

```
sudo nano /var/www/drupal7/sites/zombie/settings.php
```

scroll down to the sections which is not marked as comments and which contains data regarding the database user's username, password etc. and change the line:

```
'database' => 'drupal7',
```

to

```
'database' => 'zombiedb',
```

and save the file;

32. It's now necessary to tell Drupal about the multi site scheme. By default, Drupal isn't configured for this, but has the capability built in. The first thing to do is to copy the drupal/sites/example.sites.php as drupal/sites/sites.php. In the instance presented here:

```
cp /var/www/drupal7/sites/example.sites.php /var/www/drupal7/sites/sites.php
```

and then open the new file in a text editor:

```
sudo nano /var/www/drupal7/sites/sites.php
```

add the following to the end of the file:

```
$sites = array(  
  'drupal7' => 'default',  
  'zombieattackproductions.net' => 'zombie',  
);
```

The entries on the left- hand side of the '=' in the code above, 'drupal7' and 'zombieattackproductions.net' in this instance, contain virtual host names which will be used for resolving addresses of local websites when a browser is pointed at them. The entries on the right hand side of the '=' contain the names of corresponding directories in drupal/sites folder;

33. Visit the new website by typing 'zombieattackproductions.net' into the address bar of a web browser, follow the on- screen prompts and enter the data as required and let the installation script finish;
34. Go back to Your browser and run Cron on the new site. Everything should be fine and the work on zombieattackproductions.net web site can begin. Now two sites are present and can be worked on under a single installation of Drupal. Steps 24- 34 will need to be repeated for every website one wishes to add under a single install of Drupal;

Note: Contributed modules and themes should be installed into the drupal/sites/all/modules and drupal/sites/all/themes folders, respectively. The drupal/sites/all folder may also contain a 'libraries' folder, depending on the modules installed;

35. For practical reasons such as cooperation with other people, changing computers, testing the portability of a website created on a different OS etc, one may feel the need to import an existing website into a local Drupal install on Linux. Those interested in doing that should read on. **The first thing one needs to bear in mind is that each Drupal website needs one database and one site folder in the drupal/sites folder.** Due to an ill configured backup_migrate module, exporting the website from a Windows machine to a Linux machine was done in the way that will be explained and which will, hopefully, work every time. First, one needs to prepare the built website for export on the machine on which it has been built. To do this, archive the drupal/sites folder on that machine and then launch the phpmyadmin

applet in a web browser and export the relevant database. To avoid the problems having to do with the size of the database, export it archived. Let's assume that the site folder on the machine on which it was built, within the drupal/sites folder, was named **vampiresinhavana** and that the relevant database was named **vampirooo** for future reference. Only the database relevant to the site that's being migrated needs to be exported, the database 'vampirooo' in this case. Copy the archived sites folder as well as the exported database over to the Linux machine, for instance to the ~/temp folder. Unpack the sites folder archive into the same folder;

36. It's time to create the virtual host for the imported website. Let's assume that the decision was made to keep both the default drupal7 website, the website zombieattackproductions.net that was added and to import vampiresinhavana.com website on top of all that. So let's open the 001-drupal7.conf file in our little text editor:

```
sudo nano /etc/apache2/sites-available/001-drupal7.conf
```

and add the necessary lines:

```
<VirtualHost *:80>
    ServerAdmin admin@localhost
    ServerName vampiresinhavana.com
    ServerAlias vampires
    DocumentRoot /var/www/drupal7
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

then save the file;

37. Open the Hosts file in a text editor:

```
sudo nano /etc/hosts
```

and add the following line:

```
127.0.0.1    vampiresinhavana.com
```

38. Restart the server by typing the following line in the command prompt:

```
sudo service apache2 restart
```

39. Open the unpacked sites folder copied from the other machine in a file manager. It should contain folders named 'all', 'default' and 'vampiresinhavana' folders as well as some files. Or, it could happen that the installation of Drupal on the machine on which the site was built was configured to work with a single website, in which case the site files will be in the sites/default folder. Should the case be the former, copy only the 'all' and the 'vampiresinhavana' folders over to the drupal/sites folder in Linux (/var/www/drupal7/sites/ folder in this particular instance). If the case is the latter, rename the 'default' folder in the unpacked ~/temp/sites folder as 'vampiresinhavana' and then copy both folders to the drupal/sites directory. Only these two folders are copied because the goal is to import a built

web site into a configured install of Drupal, while avoiding the changes to configuration of the existing install on the Linux machine;

40. Set permissions on the new folder and relevant files and folders within:

```
sudo chown www-data /var/www/drupal7/sites/vampiresinhavana
sudo chmod 770 /var/www/drupal7/sites/vampiresinhavana
sudo chown www-data /var/www/drupal7/sites/vampiresinhavana/settings.php
sudo chmod 770 /var/www/drupal7/sites/vampiresinhavana/settings.php
sudo chown www-data /var/www/drupal7/sites/vampiresinhavana/files
sudo chmod 770 /var/www/drupal7/sites/vampiresinhavana/files
```

41. Open a web browser, type 'localhost/phpmyadmin' in the address bar, click on the 'Import' button, click 'Browse', navigate to the database file stored in the ~/temp folder, select it and click 'Go'.

Server: localhost

Databases SQL Status Users Export **Import** Set

Importing into the current server

File to Import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: **Browse...** No file selected. (Max: 2,048KiB)

Character set of the file: utf-8

Partial Import:

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit

Number of rows to skip, starting from the first row: 0

Format:

SQL

Format-Specific Options:

SQL compatibility mode: NONE

☒ Do not use AUTO_INCREMENT for zero values

Go

Wait for the process to complete. Log out of the phpmyadmin applet;

Note: Unless the name of the database that's being imported matches exactly the name of a database that already exists on the Linux machine, things should work smoothly. If there's a match, one could attempt deleting the existing database and then importing, or continue working with a mono- site install of drupal.

42. Now copy the settings.php file located in the drupal/sites/default folder to the imported website's folder. In the instance presented here:

```
sudo cp /var/www/drupal7/sites/default/settings.php /var/www/drupal7/sites/vampiresinhavana/settings.php
```

and open the file in a text editor:

```
sudo nano /var/www/drupal7/sites/vampiresinhavana/settings.php
```

scroll down to the relevant section and edit it as follows:

```
$databases = array (  
  'default' =>  
    array (  
      'default' =>  
        array (  
          'database' => 'vampirooo',  
          'username' => 'your_username',  
          'password' => 'your_password',  
          'host' => 'localhost',  
          'port' => '80',  
          'driver' => 'mysql',  
          'prefix' => "",  
        ),  
      ),  
    ),  
);
```

Now, your_username and your_password entries should be replaced with the username and password of the database user created in step 18, while the database field contains the name of the imported database. Port is set to 80 because the apache server is configured to listen on port 80, but nothing prevents extravagant people from configuring everything for another port. Save the file;

43. Open the sites.php file located in the drupal/sites folder in a text editor:

```
sudo nano /var/www/drupal7/sites/sites.php
```

and add the following line to the existing database array definition:

```
'vampiresinhavana.com'=>'vampiresinhavana',
```

so that the complete definition reads:

```
$sites = array(  
  'drupal7' => 'default',  
  'zombieattackproductions.net' => 'zombie',  
  'vampiresinhavana.com'=>'vampiresinhavana',  
);
```

and save the file. Now we have Drupal pointing at three virtual hosts containing three different sites;

44. Configuring Drupal and the Apache server for use with an imported site is now complete. The site can be visited by entering 'vampiresinhavana.com' into a web browser address bar. If the directory structure hasn't been altered while importing, which is to say if the directory name of the website is the same on Linux install as on the install on which the site was built (drupal/sites/vampiresinhavana in both cases) the site should work fine. If the folder was renamed, the internal links to the content (links to whatever is in the drupal/sites/vampiresinhavana/files folder) will have to be rebuilt;

45. Backup And Migrate module is a very useful tool for backing one's site up. Maybe even for migrating websites, but the author of this document was unlucky enough not to configure it properly at start. To make it work on a Linux install the following was found effective:

- Make a directory which will be used as a private path for Drupal, for instance:

```
mkdir ~/drupalprivate
```

- Assign the ownership of the directory to Drupal:

```
sudo chown www-data ~/drupalprivate
```

- Set the privileges on the directory as follows:

```
sudo chmod 750 ~/drupalprivate
```

- Visit the media/file system setting page in drupal and enter the following into the field in which the private file system path is set:

```
~/drupalprivate
```

and click 'Save configuration';

- Navigate to the backup_migrate module page in Drupal, select the 'Backup' option, select 'Advanced backup', select .zip as the compression method, check the 'Save these settings' tick mark and save the profile as default.

46. Since Internet connections don't work for everyone all the time, it may be advantageous to be able to access one's locally hosted sites while off line. Setting up a routing table properly will probably achieve this, but it has been determined that it's possible to make this happen by using the ifconfig command in the command prompt:

```
sudo ifconfig eth0 192.168.1.2
```

,192.168.1.2 being the IP number of the user's computer on a local network;

47. Links to the web- pages and a Youtube video which were very helpful are listed below. A big thank You to the authors.

<https://www.drupal.org/node/244924>

<http://community.linuxmint.com/tutorial/view/486>

<http://www.davidam.com/docu/installingdrupal.html>

<https://www.digitalocean.com/community/tutorials/how-to-configure-the-apache-web-server-on-an-ubuntu-or-debian-vps>

<https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-14-04-lts>

<https://www.drupal.org/node/1114158>

<https://www.youtube.com/watch?v=WBFlgt7kuI>