Apache MySQL PHP PHPAdmin Install

Installing Apache 2
To only install the apache2 webserver, use any method to install
apache2

It requires a restart for it to work

sudo /etc/init.d/apache2 restart

Checking Apache 2 installation

With your web browser, go to the URI http://localhost : if you read "It works!", which is the content of the file **/var/www/index.html** , this proves Apache works.

Troubleshooting Apache

If you get this error:

apache2: Could not determine the server's fully qualified domain name, using 127.0.0.1 for ServerName

then use a text editor such as "**sudo nano**" at the command line or "**gksudo gedit**" on the desktop
to create a new file,

sudo nano /etc/apache2/conf.d/fqdn
or
gksu "gedit /etc/apache2/conf.d/fqdn"

then add

ServerName localhost

to the file and save. This can all be done in a single command with the following:
echo "ServerName localhost" | sudo tee /etc/apache2/conf.d/fqdn

Virtual Hosts
Apache2 has the concept of sites, which are separate configuration files that Apache2 will read.
These are available in **/etc/apache2/sites-available**. By default, there is one site available called default this is what you will see when you browse to **http://localhost** or **http://127.0.0.1**.

You can have many different site configurations available, and activate only those that you need. As an example, we want the default site to be /home/user/public_html/. To do this, we must create a new site and then enable it in Apache2.

To create a new site:
Copy the default website as a starting point. **sudo cp /etc/apache2/sites-available/default/etc/apache2/sites-available/mysite**

Edit the new configuration file in a text editor "sudo nano" on the command line or **"gksudo gedit**", for

example: **gksudo gedit /etc/apache2/sites-available/mysite**
Change the DocumentRoot to point to the new location. For example, **/home/user/public_html/**

Change the Directory directive, replace to
/home/user/public_html/>

You can also set separate logs for each site. To do this, change the ErrorLog and CustomLog directives. This is optional, but handy if you have many sites

Save the file

Now, we must deactivate the old site, and activate our new one. Ubuntu provides two small utilities that take care of this: a2ensite (apache2enable site) and a2dissite (apache2disable site).
sudo a2dissite default && sudo a2ensite mysite

Finally, we restart Apache2:
sudo /etc/init.d/apache2 restart

If you have not created **/home/user/public_html/**, you will receive an warning message To test the new site, create a file in **/home/user/public_html/**:
echo '**Hello! It is working!**' > **/home/user/public_html/index.html**
Finally, browse to **http://localhost/**

Installing PHP 5
To only install PHP5. use any method to install the package
libapache2-mod-php5

Enable this module by doing
sudo a2enmod php5
which creates a symbolic link **/etc/apache2/mods-enabled/php5** pointing to **/etc/apache2/modsavailble/php5 .**
Except if you use deprecated PHP code beginning only by "inadvisable), open, as root, the file **/etc/php5/apache2/php.ini** , look for the line **"short_open_tag = On",** change it to **"short_open_tag = Off**" (not including the quotation marks) and add a line of comment (beginning by a semi-colon) giving the reason, the author and the date of this change.

This way, if youlater want some XML or XHTML file to be served as PHP, the "of being seen as a PHP code mistake.

Relaunch Apache 2 by
sudo service apache2 restart

Checking PHP 5 installation
In **/var/www** , create a text file called "**test.php**", grant the world (or, at least, Ubuntu user "apache")permission to read it, write in it the only line: "" (without the quotation marks) then, with your web browser, go to the URI "**http://localhost/test.php**": if you can see a description of PHP5 configuration, it proves PHP 5 works with Apache.

Troubleshooting PHP 5
Does your browser ask if you want to download the php file instead of displaying it? If Apache is not actually parsing the php after you restarted it, install

libapache2-mod-php5.

It is installed when you install the php5 package, but may have been removed inadvertently by packages which need to run a different version of php.
If **sudo a2enmod php5 returns "$ This module does not exist!", you should purge (not just remove) the libapache2-mod-php5** package and reinstall it.

Be sure to clear your browser's cache before testing your site again. To do this in Firefox 4: Edit ? **Preferences … Privacy ? History: clear your recent history ? Details : choose "Everything" in "Time range to clean" and check only "cache", then click on "Clear now".**

Remember that, for Apache to be called, the URI in your web browser must begin with "**http://**". If it begins with "**file://**", then the file is read directly by the browser, without Apache, so you get (X)HTML and CSS, but no PHP.

If you didn't configure any host alias or virtual host, then a local URI begins with **"http://localhost", "http://127.0.0.1" or http://**" followed by your IP number.
If the problem persists, check your PHP file authorisations (it should be readable at least by Ubuntu user "apache"), and check if the PHP code is correct. For instance, copy your PHP file, replace your whole PHP file content by "" (without the quotation marks): if you get the PHP test page in your web browser, then the problem is in your PHP code, not in Apache or PHP configuration nor in file permissions.

If this doesn't work, then it is a problem of file authorisation, Apache or PHP configuration, cache not emptied, or Apache not running or not restarted. Use the display of that test file in your web browser to see the list of files influencing PHP behavior.

Installing MYSQL with PHP 5
Use any method to install
mysql-server libapache2-mod-auth-mysql php5-mysql
After installing PHP
You may need to increase the memory limit that PHP imposes on a script. Edit the **/etc/php5/apache2/php.ini** file and increase the **memory_limit** value.

After installing MySQL
Set mysql bind address
Before you can access the database from other computers in your network, you have to change its bind address. Note that this can be a security problem, because your database can be accessed by other computers than your own. Skip this step if the applications which require mysql are running on the same machine. type:
nano /etc/mysql/my.cnf
and change the line: **bind-address = localhost**
to your own internal ip address e.g. 192.168.1.20
bind-address = 192.168.1.20

If your ip address is dynamic you can also comment out the bind-address line and it will default to your current ip.
If you try to connect without changing the bind-address you will recieve a "***Can not connect to mysql error 10061".***

Set mysql root password
Before accessing the database by console you need to type:
mysql -u root
At the mysql console type:
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('yourpassword');

A successful mysql command will show:
Query OK, 0 rows affected (0.00 sec)
Mysql commands can span several lines. Do not forget to end your mysql command with a semicolon.
Note: If you have already set a password for the mysql root, you will need to use:
mysql -u root -p
(Did you forget the mysql-root password? See MysqlPasswordReset.)

Create a mysql database
mysql> CREATE DATABASE database1;

Create a mysql user
For creating a new user with all privileges (use only for troubleshooting), at mysql prompt type:
mysql> GRANT ALL PRIVILEGES ON *.* TO 'yourusername'@'localhost' IDENTIFIED BY 'yourpassword' WITH GRANT OPTION;

For creating a new user with fewer privileges (should work for most web applications) which can only use the database named "**database1**", at mysql prompt type:
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES ON database1.* TO 'yourusername'@'localhost' IDENTIFIED BY 'yourpassword';

yourusername and yourpassword can be anything you like. database1 is the name of the database the user gets access to. localhost is the location which gets access to your database. You can change it to '**%**'
(or to hostnames or ip addresses) to allow connections from every location (or only from specific locations) to the database.

Note, that this can be a security problem and should only be used for testing purposes!

To exit the mysql prompt type:
mysql> q
Since the mysql root password is now set, if you need to use mysql again (as the mysql root), you will need to use:
mysql -u root -p
and then enter the password at the prompt.
Backup-Settings
Please, let's say something in which directories mysql stores the database information and how to configure a backup

Alternatively
There is more than just one way to set the mysql root password and create a database. For example mysqladmin can be used:
mysqladmin -u root -p password yourpassword
and
mysqladmin -u root -p create database1

mysqladmin is a command-line tool provided by the default LAMP install.
Phpmyadmin and mysql-admin
All mysql tasks including setting the root password and creating databases can be done via a graphical interface using phpmyadmin or mysql-admin.
To install one or both of them, first enable the universe repository
I am using Ubuntu server (command line)

I am using a desktop
Use any method to install phpmyadmin

Troubleshooting Phpmyadmin & mysql-admin
If you get blowfish_secret error: Choose and set a phrase for cryptography in the file **/etc/phpmyadmin/blowfish_secret.inc.php** and copy the line (not the php tags) into the file **/etc/phpmyadmin/config.inc.php** or you will receive an error.
If you get a 404 error upon visiting **http://localhost/phpmyadmin**:

You will need to configure
apache2.conf to work with Phpmyadmin.
sudo gedit /etc/apache2/apache2.conf

Include the following line at the bottom of the file, save and quit.
Include **/etc/phpmyadmin/apache.conf**

Alternative: install phpMyAdmin from source
See the phpMyAdmin page for instructions on how to install phpmyadmin from source:
Mysql-admin
Mysql-admin runs locally, on the desktop. Use any method to install
mysql-admin

For more information
2.9.3. Securing the Initial MySQL Accounts from the MySQL Reference Manual is worth reading.

Edit Apache Configuration
You may want your current user to be the PHP pages administrator. To do so, edit the

Apacheconfiguration file :
$ gksudo "gedit /etc/apache2/apache2.conf"

Search both the strings starting by "User" and "Group", and change the names by the current username and groupname you are using. Then you'll need to restart Apache.

Configuration options relating specifically to user websites (accessed through localhost/~username) are in /etc/apache2/mods-enabled/userdir.conf.

Run, Stop, Test, And Restart Apache

Use the following command to run Apache :

$ sudo /usr/sbin/apache2ctl start

To stop it, use :
$ sudo /usr/sbin/apache2ctl stop

To test configuration changes, use :
$ sudo /usr/sbin/apache2ctl configtest

Finally, to restart it, run :
$ sudo /usr/sbin/apache2ctl restart

Alternatively, you can use a graphical interface by installing Rapache or the simpler localhost-indicator.

Using Apache
You can access apache by typing 127.0.0.1 or http://localhost (by default it will be listening on port 80)in your browser address bar. By default the directory for apache server pages is /**var**/**www** .

It needsroot access in order to put files in. A way to do it is just starting the file browser as root in a terminal:
$ sudo nautilus
or
if you want to make /**var**/**www** your own. (Use only for non-production web servers – this is not the most secure way to do things.)
$ sudo chown -R $USER:$USER /var/www
Status

To check the status of your PHP installation:
$ gksudo "gedit /var/www/testphp.php"
and insert the following line

View this page on a web browser at http://yourserveripaddress/testphp.php or http://localhost/testphp.php

Securing Apache
If you just want to run your Apache install as a development server and want to prevent it from listening for incoming connection attempts, this is easy to do.
$ gksudo "gedit /etc/apache2/ports.conf"
$ password:

Change ports.conf so that it contains:

Listen 127.0.0.1:80

Save this file, and restart Apache (see above). Now Apache will serve only to your home domain, **http://127.0.0.1 or http://localhost.**

Password-Protect a Directory
There are 2 ways to password-protect a specific directory. The recommended way involves editing /**etc**/**apache2**/**apache2.conf** . (To do this, you need root access). The other way involves editing a **.htaccess file** in the directory to be protected. (To do this, you need access to that directory). Password-Protect a Directory With .htaccess