Step 1: Update Server Environment

Before we move forward, Run the following command to update the *apt* repository:

```
➔ sudo apt-get update
```

Step 2: Install Apache

Run the following command to install apache:

```
➔ sudo apt-get install apache2
```

After installation is complete please verify the installation:

```
➔ apache2 -v
```

It will also give you additional commands to start, stop & restart your apache server anytime you want:

```
➔ sudo service apache2 start
➔ sudo service apache2 stop
➔ sudo service apache2 restart
```

Additional Configurations:

1. Enable few apache modules:

```
➔ sudo a2enmod rewrite
➔ sudo phpenmod mbstring
```

To make the above change effective, we need to restart apache:

```
➔ sudo service apache2 restart
```

2. We need to update apache site config file so **.htaccess** works:

➪ To open the config file:

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

➪ This will open the apache default config file in view mode. We need to update that file. So **click " I "** to enable the insert / edit mode.

➪ Make sure your indent the file properly. And leave one line gap after DocumentRoot **/var/www/html** and insert below code:

```
<Directory /var/www/html>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    allow from all
</Directory>
```

➪ Now, to save the above changes we need to continue with **" Esc "** → **" : "** → **" w "** → **" q "**

➪ Finally we need to make the above change effective, we need to restart apache:

```
➔ sudo service apache2 restart
```

Step 3: Install MySQL

Again, Run the following command to update the apt repository:

```
➔ sudo apt-get update
```

Run the command below on the terminal window to install MySQL server:

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

After installation is complete please verify the installation:

```
➔ mysql -V
```

MySQl server ships with default settings which are not secure. We need to configure it to make our server less vulnerable to attacks. Luckily there is a single configuration command that will make things easier for us. Just run the command below:

```
➔ sudo mysql_secure_installation
```

★ Would you like to setup VALIDATE PASSWORD plugin?

N

★ Please set the password for root here:
### Set password and Confirm password.

★ Remove anonymous users?
### y/Y - for yes.

★ Disallow root login remotely?
### y/Y - for yes.

★ Remove test database and access to it?
### y/Y - for yes.

★ Reload privilege tables now?
### y/Y - for yes.

Additional Configurations to Secure MYSQL Access - Adjusting User Authentication and Privileges:

➪ To do this, open up the MySQL prompt from your terminal:

```
➔ sudo mysql
```

➪ Next, check which authentication method each of your MySQL user accounts use with the following command:

```
➔ Mysql > SELECT user,authentication_string,plugin,host FROM mysql.user;
```

That will display all the existing mysql users in table format:

IMG

➪ In this example, you can see that the root user does in fact authenticate using the auth_socket plugin. To configure the root account to authenticate with a password, run the following ALTER USER command. Be sure to change the **password** to a **strong password** of your choosing, and **Note:** *that this command will change the root password you set previously* :

```
➔ Mysql > ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password
  BY 'password';
```

➪ Next, to check if password is set:

```
➔ Mysql > SELECT user,authentication_string,plugin,host FROM mysql.user;
```

That will display all the existing mysql users in table format:

IMG -2

➪ Next, to flush privilege table:

```
➔ Mysql > FLUSH PRIVILEGES;
```

➪ Once it is done, The following will run your MySQL client with regular user privileges, and you will only gain administrator privileges within the database by authenticating:

```
➔ Mysql > CREATE USER 'user'@'localhost' IDENTIFIED BY 'password';
➔ Mysql > GRANT ALL PRIVILEGES ON *.* TO 'user'@'localhost' WITH GRANT
  OPTION;
➔ Mysql > FLUSH PRIVILEGES;
```

➪ Next, to check if new user and password is set:

```
➔ Mysql > SELECT user,authentication_string,plugin,host FROM mysql.user;
```

➪ Next, exit from the mysql terminal and reload mysql service once:

```
➔ sudo service mysql restart
```

Step 4: Install PHP

Again, Run the following command to update the apt repository:

```
➔ sudo apt-get update
```

Enter the command below to install PHP:

```
➔ sudo apt-get install php libapache2-mod-php
```

Next, install necessary php modules:

```
➔ sudo apt-get install php-gd php-mysql php-dom php-cli php-json php-common
  php-mbstring php-opcache php-readline php-curl php-gettext php-zip
```

Restart apache to activate the changes:

```
➔ sudo service apache2 restart
```

Test Settings:

➪ Create info.php to test php:

```
➔ sudo vi /var/www/html/info.php
```

⇨ **Click " I "** to enable the insert / edit mode and insert below code:

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

⇨ Now, to save the above changes we need to continue with **" Esc "** → **" : "** → **" w "** → **" q "**. You new can run your browser and check the following url : http://your_site_ip/info.php

Step 5: Install phpMyAdmin

Funny! but, Run the following command to update the apt repository:

```
➔ sudo apt-get update
```

Next we need to install zip:

```
➔ sudo apt-get install zip
```

I personally do not prefer default phpMyAdmin installation so I will do that with zip archive from phpMyAdmin domain. Link: https://www.phpmyadmin.net/downloads/

Next, run the following series of commands to upload it to the server:

```
➔ cd /var/www
➔ sudo wget https://files.phpmyadmin.net/phpMyAdmin/4.9.0.1/phpMyAdmin-4.9.0.1-english.zip
➔ sudo unzip phpMyAdmin-4.9.0.1-english.zip
➔ sudo rm phpMyAdmin-4.9.0.1-english.zip
➔ sudo mv phpMyAdmin-4.9.0.1-english phpMyAdmin
➔ ls -l
```

After **LS -L** you can see there is two directory **/html** and **/phpMyAdmin**. But, yet the phpMyAdmin is not accessible with URL because it is not hosted in /html. We need to set the Alias to set access.

```
➔ sudo vi /etc/apache2/mods-enabled/alias.conf
```

⇨ This will open the alias.conf config file in view mode. We need to update that file. So **click " I "** to enable the insert / edit mode.

⇨ Make sure your indent the file properly. And leave one line gap after **place below alias there after icons alias**:

```
        Alias /phpMyAdmin "/var/www/phpMyAdmin/"

        <Directory "/var/www/phpMyAdmin/">
                Options Indexes FollowSymLinks MultiViews
                AllowOverride All
                Order allow,deny
                allow from all
        </Directory>
```

➪ Now, to save the above changes we need to continue with **“ Esc ”** → **“ : ”** → **“ w ”** → **“ q ”**.

➪ Now, to add an additional layer of security I prefer to add external security layer to access the phpMyAdmin using **.htaccess** and **.htpasswd**:

.htaccess

```
AuthType Basic
AuthName "Password Protected Area"
AuthUserFile /var/www/phpMyAdmin/.htpasswd
Require valid-user
```

.htpasswd

```
innofied:$apr1$BA6dpmNC$CqpUOQVO7RiHNbMiixW6N0
```

FTP LOGIN CREDENTIAL IS: innofied / inno#123

Restart apache to activate the changes:

```
    ➔ sudo service apache2 restart
```

Step 6: Change default apache user

First we need to make change on apache environment variables:

```
    ➔ sudo vi /etc/apache2/envvars
```

➪ This will open the envvars config file in view mode. We need to update that file. So **click “ I ”** to enable the insert / edit mode.

➪ Make sure your indent the file properly. Now update APACHE_RUN_USER and APACHE_RUN_GROUP to ubuntu.

➪ Now, to save the above changes we need to continue with **“ Esc ”** → **“ : ”** → **“ w ”** → **“ q ”**.