

# Installazione e configurazione di un server LAMP locale su Ubuntu 11.10

## 1 Introduzione

In questa breve guida vedremo come installare e configurare su Ubuntu 11.10 un server web composto da:

- Apache (web server)
- MySQL (database)
- PHP (linguaggio di scripting)

Lo scopo della guida è quello di realizzare un server per lo sviluppo ed il testing in locale di siti web sviluppati in PHP.

ATTENZIONE: tutte le configurazioni sono pensate per un utilizzo come sistema di sviluppo! NON sono quindi adatte per un server da utilizzare effettivamente per fornire servizi web!

Tutti i comandi verranno eseguiti tramite il terminale, è quindi necessario conoscerne almeno le basi.

Ok, cominciamo aprendo proprio il terminale!

## 2 Installazione di Apache

Installiamo il server web Apache 2 con il comando

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

Verrà richiesta l'installazione di alcuni software aggiuntivi. Alla richiesta "Continuare?" confermiamo con il tasto Invio.

Verifichiamo se l'installazione è andata a buon fine aprendo un browser e digitando nella barra degli indirizzi

```
http://localhost/
```

Dovrebbe comparire una pagina con il messaggio "It works!".

## 3 Configurazione di Apache

Vediamo ora come configurare i parametri più comuni di Apache.

### 3.1 Configurazione dell'utente

Per prima cosa andiamo a modificare l'utente che esegue il server. Benchè questo non sia assolutamente necessario, rende più semplice il testing delle pagine PHP. Andremo inoltre a limitare l'accesso al server dal solo computer locale per ovvie ragioni di sicurezza, trattandosi, come già accennato, di un server per lo sviluppo.

Apriamo il file `/etc/apache2/envvars` con un editor di testo (qui uso l'editor grafico *Gedit*, ma si può sostituire con qualunque altro editor, come *Nano* o *Vim*).

```
sudo gedit /etc/apache2/envvars
```

(come tutti i comandi eseguiti con `sudo` verrà richiesta la password dell'utente per continuare)

Cerchiamo le righe che iniziano con `export APACHE_RUN_USER` e `export APACHE_RUN_GROUP` e sostituiamo il valore dopo il simbolo di uguaglianza con il nostro username. Ad esempio, se il nome utente è `“pippo”` modificheremo le due righe così:

```
export APACHE_RUN_USER=pippo
export APACHE_RUN_GROUP=pippo
```

Salviamo e chiudiamo il file, lasciandone inalterato il resto.

### 3.2 Limitazione dell'accesso al solo computer locale

Apriamo `/etc/apache2/ports.conf`:

```
sudo gedit /etc/apache2/ports.conf
```

e sostituiamo

```
Listen 80
```

con

```
Listen 127.0.0.1:80
```

### 3.3 Document Root

Andiamo ora a cambiare la document root del server. Questa directory contiene i file che vengono serviti da Apache agli utenti del web. Di default viene usata `/var/www`, ma per comodità sposteremo questa cartella nella nostra home directory. Creiamo la cartella `www` all'interno della nostra home:

```
mkdir /home/pippo/www
```

(sostituendo ovviamente `pippo` con il nostro nome utente) e apriamo un altro file di configurazione:

```
sudo gedit /etc/apache2/sites-enabled/000-default
```

In questo file cerchiamo la riga

```
DocumentRoot /var/www
```

e sostituiamola con:

```
DocumentRoot /home/pippo/www
```

(sostituendo ovviamente `pippo` con il nostro nome utente).

Poi cerchiamo `<Directory /var/www/>` e sostituiamo con:

```
<Directory /home/pippo/www/>
```

(cambiando sempre `pippo` con il nostro nome utente). Salviamo e chiudiamo il file.

### 3.4 Attivazione delle modifiche

Possiamo ora riavviare Apache per rendere effettive le modifiche effettuate:

```
sudo /etc/init.d/apache2 restart
```

Riapriamo con un browser la pagina `http://localhost/`. Ora si dovrebbe vedere il messaggio “Index of /”.

## 4 Installazione di PHP e MySQL

Installiamo il tutto con il comando

```
sudo apt-get install php5 libapache2-mod-php5 php5-gd  
php5-imagick php5-mcrypt php5-mysql php5-sqlite mysql-server
```

Come al solito confermiamo con Invio l’installazione dei pacchetti aggiuntivi richiesti. Durante l’installazione ci verrà inoltre richiesta una password per l’utente `root` di MySQL. Terminiamo l’operazione riavviando nuovamente Apache:

```
sudo /etc/init.d/apache2 restart
```

## 5 Installazione di PHPMyAdmin

Questo software serve per gestire tramite una interfaccia grafica il database MySQL. Installiamolo con:

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

Durante l’installazione verrà richiesto il webserver su cui installare il programma. Scegliamo `apache2` premendo Spazio e poi Invio. Rispondiamo inoltre “Sì” alla richiesta di “Configurare il database di `phpmyadmin` con `dbconfig-common`”.

PHPMyAdmin è ora accessibile puntando il browser all’indirizzo

```
http://localhost/phpmyadmin/
```

I parametri di accesso sono:

- Nome utente: `root`
- Password: *la password scelta per MySQL*

## 6 Conclusione

Ora è tutto pronto. Possiamo cominciare quindi ad inserire i file PHP del nostro sito web all’interno della cartella `www` nella nostra home directory. Per fare una prova si può creare il file `info.php` con il seguente contenuto:

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

Puntando il browser all'indirizzo `http://localhost/info.php` vedremo la pagina appena creata!

## 7 Altre informazioni utili

### 7.1 Avviare e fermare i server

Per avviare e fermare Apache e MySQL si possono usare i seguenti comandi. Per l'avvio:

```
sudo /etc/init.d/apache2 start
sudo start mysql
```

Per l'arresto:

```
sudo /etc/init.d/apache2 stop
sudo stop mysql
```

## 8 Evitare l'avvio automatico dei server all'avvio del computer

Dato che probabilmente non useremo Apache e MySQL ogni volta che avviamo il computer, possiamo impedirne l'avvio automatico in questo modo. Per disabilitare l'avvio automatico di Apache usiamo il comando:

```
sudo update-rc.d -f apache2 remove
```

Per MySQL, invece, apriamo il file `/etc/init/mysql.conf`:

```
sudo gedit /etc/init/mysql.conf
```

ed inseriamo il carattere “#” all'inizio delle righe

```
start on (net-device-up
          and local-filesystems
          and runlevel [2345])
```

che diventeranno quindi:

```
#start on (net-device-up
#          and local-filesystems
#          and runlevel [2345])
```