

# Workstation LAMP con Ubuntu

**Autore:** Paolo Ferretti <http://www.paoloferretti.it>

*Cosa serve per sviluppare un sito dinamico sul web? La risposta è semplice, un server web, un database e un linguaggio di programmazione. In questo articolo andremo a creare e analizzare una configurazione base per lo sviluppo di applicazioni web con Ubuntu, installando Apache, MySQL, PHP e per finire phpMyAdmin.*

## LAMP

L'acronimo LAMP significa Linux Apache MySQL PHP. Ultimamente, con l'aumento di popolarità di Python, l'ultima P assume anche il significato di Python.

Inizieremo con l'installazione e configurazione del server web Apache, per poi passare a MySQL e PHP. Infine, andremo a testare il nostro ambiente di sviluppo con una famosa applicazione web per la gestione di MySQL: phpMyAdmin. Il sistema utilizzato per scrivere l'articolo è Ubuntu Linux 8.04.1 LTS.

## Installazione e configurazione di Apache

Su Ubuntu l'installazione di Apache è semplice come quella di qualsiasi altro software, con la differenza che dobbiamo usare la linea di comando, dato che Apache non compare nella lista delle normali applicazioni di Ubuntu Desktop. Per installare il server web, lanciate il comando:

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

Dato che Apache 2 è un software molto articolato, sicuramente il sistema di gestione pacchetti vi chiederà di installare ulteriore software a completamento.

Per verificare che l'installazione sia andata a buon fine, aprite il vostro browser web e digitate l'indirizzo `http://127.0.0.1`. Dovreste trovarvi davanti una pagina così:

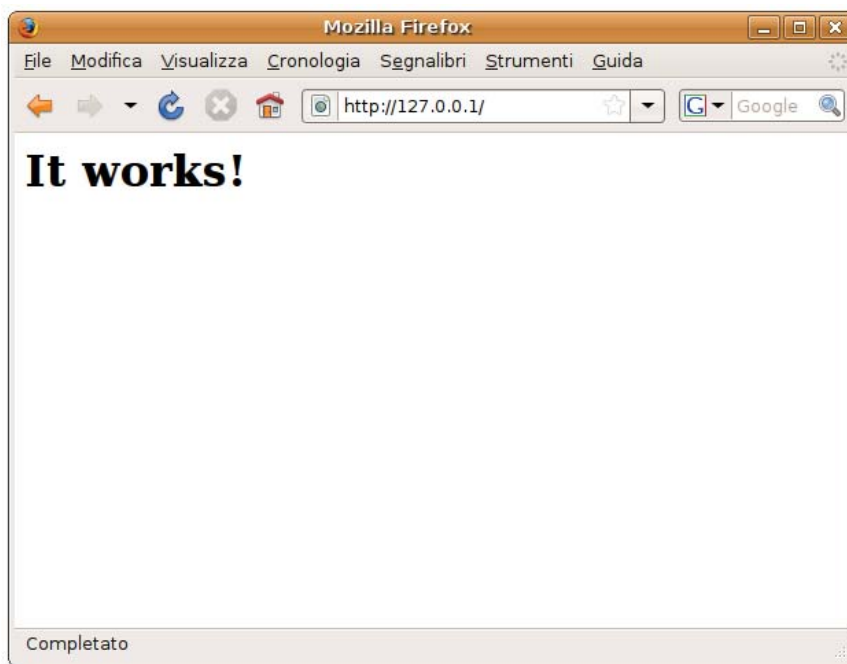


Figura 1: Apache è avviato e funzionante

Purtroppo la configurazione predefinita di Apache va bene solo per visualizzare questa pagina. Per avere un ambiente su misura, bisogna mettere mano ai file di configurazione.

**NOTA**

*Prima di procedere con i passi successivi, si consiglia di leggere l'articolo che parla della console dei comandi.*

Portatevi nella cartella `/etc/apache`, la quale contiene tutti i file di configurazione di Apache. Normalmente il file di configurazione principale è `httpd.conf`. Su Ubuntu, invece, il sistema di configurazione è un po' diverso dal solito: il file principale è `apache.conf`, in più ci sono molti file accessori che vengono inclusi in quello principale tramite una apposita direttiva. In questo modo, ogni volta che si vuole aggiungere o togliere qualcosa nel file di configurazione, non si va a modificare direttamente `apache.conf`, ma si aggiungono o tolgono i file accessori. È un modo molto efficace di gestire la configurazione, dato che questa frammentazione rende più facile trovare eventuali errori di sintassi.

Portatevi nella sotto-cartella **sites-available**, che contiene tutti i virtual hosts disponibili nella vostra configurazione di Apache. L'unico file presente è **default**, contenente la configurazione predefinita del virtual host principale di Apache. Copiate questo file per creare un nuovo virtual host. Nel nostro caso lo abbiamo copiato col nome **vh\_prova**.

**NOTA**

*ovviamente, dato che siamo all'interno della cartella **/etc**, per poter fare modifiche alla struttura del filesystem e ai file stessi, dobbiamo avere i diritti di amministrazione.*

Aperto il file appena creato, vi trovate davanti al file di configurazione per il vostro nuovo host virtuale:

```
<VirtualHost *>
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www/

  <Directory />
    Options FollowSymLinks
    AllowOverride None
  </Directory>

  <Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
  </Directory>

  ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/

  <Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
  </Directory>
```

```
ErrorLog /var/log/apache2/error.log

# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn
CustomLog /var/log/apache2/access.log combined
ServerSignature On

Alias /doc/ "/usr/share/doc/"
<Directory "/usr/share/doc/">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
    Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
</VirtualHost>
```

Ai fini del nostro articolo, sfortiremo questo file per renderlo più semplice e maneggevole, senza le complicazioni tipiche di una configurazione da server. In fondo il nostro obiettivo è avere una workstation funzionante, non un server perfettamente configurato. Il nostro nuovo file, debitamente sfortito, risulterà essere:

```
NameVirtualHost *
<VirtualHost *>
    ServerAdmin vostra_email@prova.it
    ServerName vh_prova.backintime

    DocumentRoot /home/paolo/public/vh_prova
    <Directory />
        Options FollowSymLinks
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
</VirtualHost>
```

Le due righe più importanti di questo file sono:

```
ServerName vh_prova.backintime
DocumentRoot /home/paolo/public/vh_prova
```

La prima è auto-documentante: indica il nome, o indirizzo, del nostro host virtuale. Dalla nostra configurazione di esempio, si evince che il nostro host risponderà all'indirizzo:

*http://vh\_prova.backintime.*

Come scegliere l'indirizzo? Di solito consiglio di usare **nome\_virtual\_host.nome\_computer**; nel nostro caso, come si intuisce, il nome del computer è backintime.

La seconda riga, invece, indica la nostra document root, cioè la cartella radice del nostro host virtuale, quella che conterrà tutti i file della futura applicazione web che svilupperemo. Nel nostro caso, ho deciso che i files risiederanno nella sottocartella **vh\_prova** della mia cartella **public**. Qualsiasi cartella va bene come document root, a patto che possa essere accessibile dal server web Apache.

Il significato delle altre opzioni di configurazione esula dallo scopo dell'articolo, quindi rimando alla documentazione ufficiale di Apache disponibile all'indirizzo <http://httpd.apache.org/docs/2.2/>.

Dopo aver sistemato la configurazione del nostro host virtuale, dovete creare un link simbolico nella cartella **sites-enabled**:

```
$ sudo ln -s /etc/apache2/sites-available/vh_prova /etc/apache2/sites-enabled/
```

Creando questo link simbolico, avete reso attivo il vostro nuovo host: infatti questa cartella contiene i link simbolici alle configurazioni degli host virtuali che devono essere attivati sulla vostra installazione di Apache.

Adesso dovete rendere accessibile l'indirizzo del nostro virtual host: infatti digitando l'indirizzo: *http://vh\_prova.backintime*, il vostro computer andrà a cercare la risoluzione del nome nel vostro server DNS, che nella maggior parte dei casi risponderà con un errore. Per effettuare un by-pass del DNS, dovete modificare il file `/etc/hosts`, che contiene le informazioni sugli host conosciuti. Aggiungete la riga:

```
127.0.0.1          vh_prova.backintime
```

e salvate il file. Ovviamente, se avete scelto un nome diverso da **vh\_prova.backintime**, mettete la vostra scelta.

La configurazione del virtual host è finita. Adesso non resta che scrivere una pagina `index.html` di prova per testarne il funzionamento. Per i più pigri, ecco un semplice documento HTML per un veloce copia-incolla:

```
<html>
<head>
  <title>Home page di prova</title>
</head>
<body>
  <h1>Home page di prova</h1>
  <p>Questa &egrave; la prima home page di prova</p>
</body>
</html>
```

Salvatelo nella document root che avete indicato in precedenza. L'ultima operazione da fare è riavviare Apache per rendere attive le modifiche:

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

Se il riavvio va a buon fine, dovrete trovarvi davanti a una schermata così:

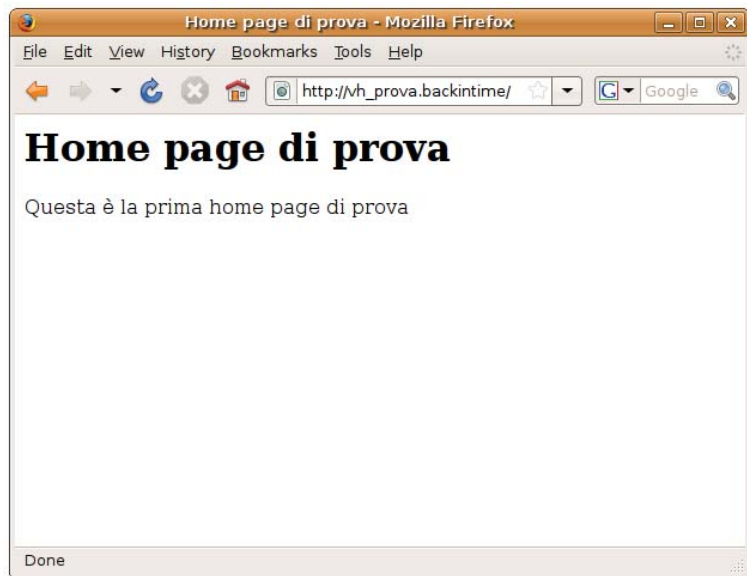


Figura 2: VirtualHost che funziona

Nel caso in cui il riavvio di Apache non abbia successo, controllate la sintassi del vostro file di configurazione: nella maggior parte dei casi il problema è quello.

## Installazione e configurazione di MySQL

MySQL, almeno sotto l'aspetto di installazione e configurazione, è decisamente più semplice di Apache. Per installare MySQL server, aprite la console e scrivete:

```
$ sudo apt-get install mysql-server-5.0
```

In questo articolo ci basiamo sulla versione 5, ma nei repository di Ubuntu è disponibile anche la 4.1. Consiglio anche l'installazione degli strumenti di configurazione:

```
$ sudo apt-get install mysql-admin mysql-query-browser
```

Questi strumenti sostituiscono la riga di comando per molte operazioni, rendendo più comoda la gestione del nostro server MySQL.

La prima cosa da fare, dopo l'installazione, è creare un utente non amministratore: non è saggio utilizzare privilegi di amministrazione per i normali accessi al database. Aprite MySQL Administrator, lo trovate in **Applicazioni** -> **Programmazione**.

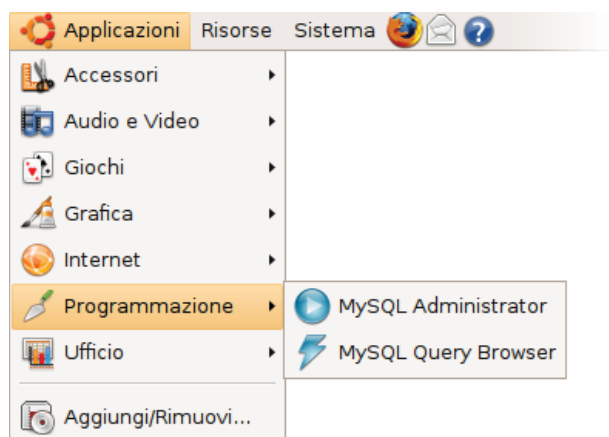


Figura 3: il menu Programmazione con i collegamenti alle utility di amministrazione per MySQL Server

Nella finestra che compare, immettete **localhost** nel campo **Server HostName**, il vostro nome utente nel campo **username** e la relativa **password** nel campo rimanente. La porta deve rimanere 3306, a meno che non abbiate specificato un'impostazione diversa in fase di installazione. MySQL Administrator permette di gestire molti aspetti della configurazione di MySQL: noi ci occuperemo delle sezioni **User Administration** e **Catalogs**. Portatevi nella prima e cliccate sul pulsante **Add User** situato nella parte bassa della finestra. Per ora limitiamoci a definire nome utente e password. Quando avete finito, premete il pulsante **Apply Changes**.

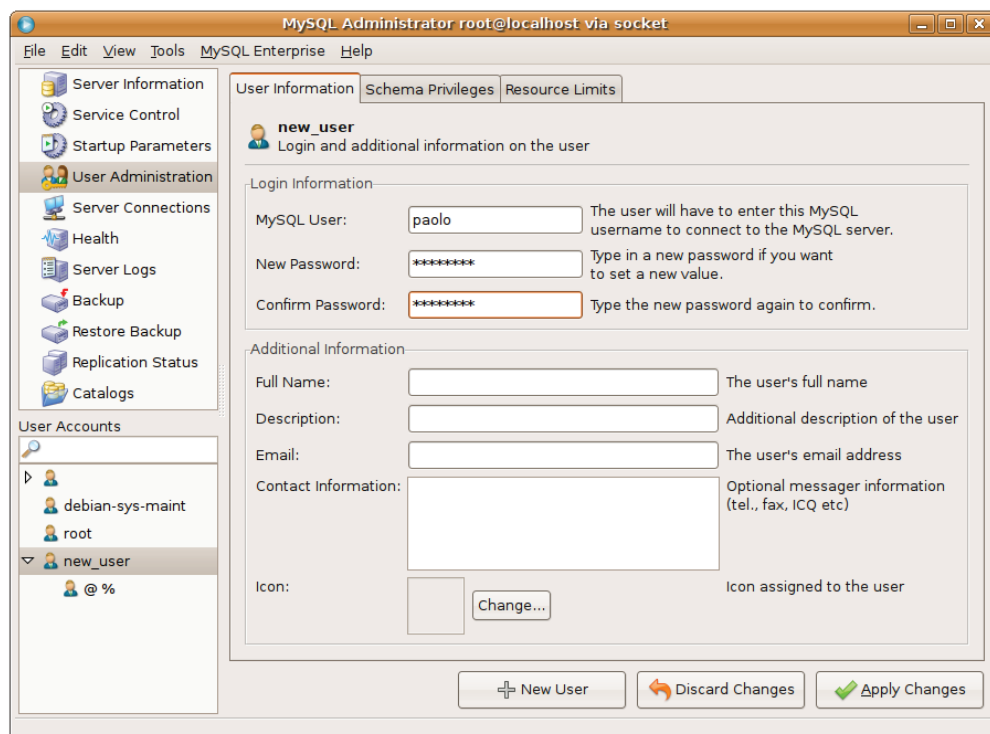


Figura 4: aggiunta utente senza privilegi di amministrazione

Completata questa operazione, cliccate col tasto destro sul nome dell'utente appena creato, presente nella parte in basso a sinistra della finestra. Insieme all'utente nuovo, dovrebbero esserci anche gli utenti root e debian-sys-maint. Facendo clic col tasto destro, compare un menu popup: selezionate la voce **Add Host**. Nella finestra che compare selezionate **Localhost** e poi confermate la scelta, sia cliccando su **Ok**, sia applicando le modifiche come al passo precedente. Così avete dato accesso al vostro nuovo utente, specificando di accettare la connessione dal nostro computer locale.

Adesso spostatevi in **Catalogs**, cliccate col tasto destro nell'area in basso a sinistra della finestra e selezionate **Create Schema**. Il nome usato all'interno di questo articolo sarà **prova**. Dopo aver creato il database, tornate al pannello di amministrazione del vostro nuovo utente e selezionate la tab **Schema Privileges**. Qui potete dare all'utente selezionato i privilegi sui vari database. In questo caso, date tutti i privilegi sul database di prova appena creato.

### NOTA

*Normalmente sui server di produzione non si hanno tutti i privilegi sui propri database. Se conoscete già i privilegi che avrete in produzione, usate quelli. In caso contrario, per iniziare, la soluzione più semplice è quella consigliata nel paragrafo sopra.*

Se tutto è andato bene, avete un database pronto per essere utilizzato nella vostra futura applicazione web.

## Installazione PHP

L'ultimo passo, a livello di installazione del software, consiste nell'installare l'interprete PHP. Si può optare per almeno due alternative: installazione manuale o attraverso i pacchetti disponibili nei repository di Ubuntu. La prima opzione è indicata per chi ha esigenze particolari e magari vuole anche compilarsi PHP dai sorgenti. Noi useremo i pacchetti forniti da Ubuntu: comodi, veloci da installare e auto-configuranti. L'ultima caratteristica è sicuramente la più interessante. Infatti, optando per l'installazione manuale, si deve passare per forza attraverso la configurazione di PHP: operazione non banale se non si è consapevoli di quello che si sta facendo.

Per installare PHP, scrivete sulla console:

```
$ sudo apt-get install php5
```

È d'obbligo installare anche qualche estensione, ci sono pacchetti precompilati per le estensioni più comuni. Per ora limitiamoci al supporto per MySQL, senza il quale non potete accedere al vostro database:

```
$ sudo apt-get install php5-mysql
```

L'installazione di PHP è finita qui. I pacchetti installati si sono auto-configurati e Apache dovrebbe riconoscere i documenti con estensione php. Per verificarlo, cambiate l'estensione da html a php al primo documento HTML che avete utilizzato come indice di prova, poi modificalo così:

```
<html>
<head>
  <title>Home page di prova</title>
</head>
<body>
  <h1>Home page di prova</h1>
  <p>Questa è la prima home page di prova</p>
  <p><?php echo "Anche i documenti PHP funzionano"; ?></p>
</body>
</html>
```

Il risultato nel browser dovrebbe quindi essere:

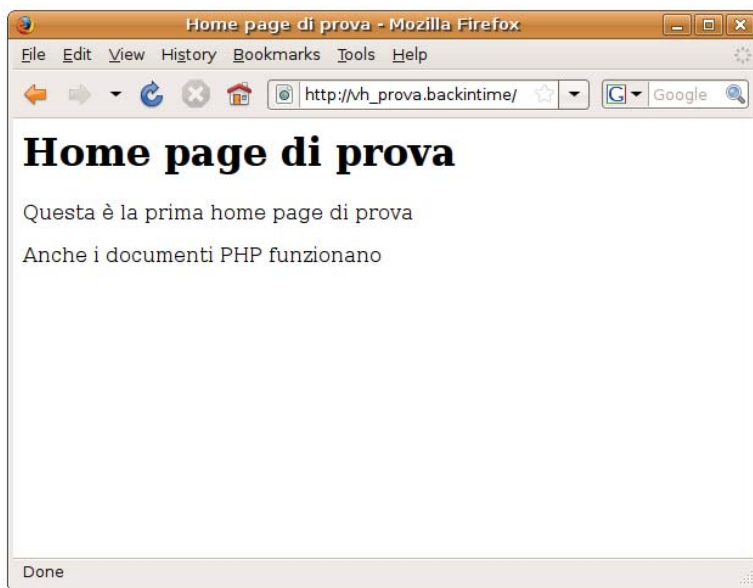


Figura 5: i documenti PHP funzionano

## Installare phpMyAdmin

Scaricate dal sito di phpMyAdmin, disponibile all'indirizzo: <http://www.phpmyadmin.net/>, l'ultima versione stabile, nel nostro caso è la 3.0.1.1. Decomprimate il file nella root directory del vostro virtual host e poi create una copia del file **config.sample.inc.php**. Successivamente rinominate la copia in **config.inc.php** e aprite il file per modificarlo. Generalmente non si devono fare modifiche a questo file, va bene così com'è per gli utilizzi più comuni. Se volete l'autenticazione tramite cookie, immettete un valore per la variabile `blowfish_secret`.

Caricate la home page del vostro virtual host: dovrete vedere la pagina di login di phpMyAdmin. Fate il login col vostro utente e provate a fare delle modifiche al database di prova per verificare il corretto funzionamento della configurazione.

Se avete problemi con qualche passo dell'articolo, contattatemi pure. I miei contatti sono appena sotto il titolo dell'articolo.

## Conclusioni

In questo articolo della sezione Linux & Co. di Come fare a... abbiamo visto come installare il server web Apache, MySQL e PHP su un sistema Linux Ubuntu 8.0.4.

Continuate a seguire i nostri articoli su Linux e il mondo open source collegandovi all'indirizzo: <http://www.comefarea.it/linuxandco/>, per tenervi sempre aggiornati abbonatevi alla nostra newsletter all'indirizzo: <http://www.comefarea.it/newsletter/> o, se utilizzate Windows Live Messenger, abbonatevi al nostro Windows Live Alerts disponibile sulla pagina <http://www.comefarea.it/abbonamenti.php>.