

Ficha de Trabalho N.º 7 – Instalação do LAMP no Ubuntu

Introdução

O conjunto “LAMP” é um grupo de softwares open source que é tipicamente instalado em conjunto para permitir a um servidor hospedar sites dinâmicos e aplicações web. Este termo é atualmente um acrónimo que representa o sistema operacional **L**inux, com o servidor web **A**pache. A informação do site é armazenada em uma base de dados **M**ySQL, e o conteúdo dinâmico é processado pelo **P**HP.

Neste guia, vamos instalar a plataforma LAMP em um servidor Ubuntu

Passo 1 — Instalação do Apache e Atualização do Firewall

O servidor web Apache está entre os servidores web mais populares do mundo. É bem documentado, e tem sido amplamente utilizado em grande parte da história da web, o que faz dele uma ótima escolha padrão para hospedar um site.


Instale o Apache utilizando o gestor de pacotes do Ubuntu, apt:

- `sudo apt update`
- `sudo apt install apache2`

Como estamos utilizando um comando `sudo`, essas operações são executadas com privilégios de root. Ele irá pedir a senha do utilizador comum para verificar suas intenções.

Pressione `Y` e clique no `Enter` para continuar, e a instalação prosseguirá.

Ao entrar no browser e digitando o endereço <http://localhost> (endereço que identifica a própria máquina) deverá visualizar a página web padrão do 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.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in** [/usr/share/doc/apache2/README.Debian.gz](#). Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

Se estiver a visualizar esta página, então o seu servidor web está corretamente instalado e acessível.

Passo 2 — Instalação do MySQL

Agora que temos o nosso servidor web pronto e funcional, passemos à instalação do MySQL. O MySQL é um sistema de gestão de bases de dados. Basicamente, ele irá organizar e fornecer acesso às bases de dados onde os sites desenvolvidos podem armazenar informação.

Novamente, utilize o `apt` para obter e instalar este software:

- `sudo apt install mysql-server`

Novamente, será mostrada uma lista dos pacotes que serão instalados, juntamente com a quantidade de espaço em disco que irão ocupar. Digite `s` para continuar.

Quando a instalação estiver concluída, execute um script de segurança simples que vem pré-instalado com o MySQL e que irá remover alguns padrões perigosos e bloquear o acesso ao seu sistema de base de dados. Inicie o script interativo executando:

- `sudo mysql_secure_installation`
-

Será questionado se quer configurar o `VALIDATE PASSWORD PLUGIN`.

Nota: A habilitação dessa funcionalidade é algo que deve ser avaliado. Se habilitado, senhas que não seguem o critério especificado serão rejeitadas pelo MySQL com um erro. Isso irá causar problemas se você utilizar uma senha fraca juntamente com software que configura automaticamente as credenciais de usuário do MySQL, tais como os pacotes do Ubuntu para o phpMyAdmin. É seguro deixar a validação desativada, mas você deve sempre utilizar senhas fortes e exclusivas para as credenciais do banco de dados.

Responda `N` para **Não**. Obviamente que em um cenário real deveria avançar com o procedimento.

Neste ponto, o sistema de gestão de base de dados está agora configurado e podemos seguir em frente para a instalação do PHP, o componente final da plataforma LAMP.

Passo 3 — Instalação do PHP

O PHP é o componente da nossa configuração que irá processar código para mostrar o conteúdo dinâmico de um site. Ele pode executar script, conectar às bases de dados MySQL para obter informações, e entregar o conteúdo processado para que o nosso servidor web possa mostrar esses dados.

Uma vez mais, aproveite o sistema `apt` para instalar o PHP. Adicionalmente, inclua alguns pacotes auxiliares para que o código PHP possa ser executado sob o servidor Apache e comunicar com a sua base de dados MySQL:

- `sudo apt install php libapache2-mod-php php-mysql`

Este comando irá instalar o PHP. Vamos de seguida testar o seu funcionamento.

Na maioria dos casos, devemos modificar a forma com a qual o Apache responde com arquivos quando uma pasta é requisitada. Atualmente, se um utilizador requisita uma pasta do servidor, o Apache irá olhar em primeiro lugar para um ficheiro chamado `index.html`. Como queremos informar ao nosso servidor web para dar preferência aos arquivos PHP, então faremos o Apache olhar para um arquivo `index.php` em primeiro lugar.

Para fazer isto, execute o seguinte comando para abrir o arquivo `dir.conf` em um editor de texto com privilégios de root:

- `sudo nano /etc/apache2/mods-enabled/dir.conf`

Ele terá esta aparência:

```
/etc/apache2/mods-enabled/dir.conf
```

```
/etc/apache2/mods-enabled/dir.conf
```

```
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml
index.htm
</IfModule>
```

Mova o arquivo de índice PHP (em destaque acima) para a primeira posição depois da especificação `DirectoryIndex`, como apresentado de seguida:

```
/etc/apache2/mods-enabled/dir.conf
```

```
/etc/apache2/mods-enabled/dir.conf
```

```
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml
index.htm
</IfModule>
```

Após terminar, guarde e feche o ficheiro usando a combinação de teclas `CTRL-X`. Confirme a gravação digitando `Y` e em seguida pressione `ENTER` para confirmar a localização do ficheiro guardado.

Após isso, reinicie o servidor web Apache de forma que alterações sejam reconhecidas. O reinício do Apache pode ser realizado com o seguinte comando:

- `sudo systemctl restart apache2`

Também pode ser verificado o status do serviço `apache2` utilizando `systemctl`:

- `sudo systemctl status apache2`

Sample Output

```
• apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Tue 2018-04-23 14:28:43 EDT; 45s ago
     Docs: man:systemd-sysv-generator(8)
   Process: 13581 ExecStop=/etc/init.d/apache2 stop (code=exited,
status=0/SUCCESS)
   Process: 13605 ExecStart=/etc/init.d/apache2 start (code=exited,
status=0/SUCCESS)
```

```
Tasks: 6 (limit: 512)
CGroup: /system.slice/apache2.service
├─13623 /usr/sbin/apache2 -k start
├─13626 /usr/sbin/apache2 -k start
├─13627 /usr/sbin/apache2 -k start
├─13628 /usr/sbin/apache2 -k start
├─13629 /usr/sbin/apache2 -k start
└─13630 /usr/sbin/apache2 -k start
```

Para melhorar a funcionalidade do PHP, temos a opção de instalar alguns módulos adicionais. Para ver as opções disponíveis para os módulos e bibliotecas PHP, direcione os resultados do comando `apt search` para o comando `less`, um paginador que lhe permite percorrer a saída de outros comandos:

- `apt search php- | less`

Use as teclas de seta para navegar para cima e para baixo e `q` para sair.

O resultado são todos os componentes opcionais que podem ser instalados. Será dada uma breve descrição de cada um:

```
bandwidthd-pgsql/bionic 2.0.1+cvs20090917-10ubuntu1 amd64
  Tracks usage of TCP/IP and builds html files with graphs
```

```
bluefish/bionic 2.2.10-1 amd64
  advanced Gtk+ text editor for web and software development
```

```
cacti/bionic 1.1.38+ds1-1 all
  web interface for graphing of monitoring systems
```

```
ganglia-webfrontend/bionic 3.6.1-3 all
  cluster monitoring toolkit - web front-end
```

```
golang-github-unknwon-cae-dev/bionic 0.0~git20160715.0.c6aac99-4 all
  PHP-like Compression and Archive Extensions in Go
```

```
haserl/bionic 0.9.35-2 amd64
  CGI scripting program for embedded environments
```

```
kdevelop-php-docs/bionic 5.2.1-1ubuntu2 all
transitional package for kdevelop-php
```

```
kdevelop-php-docs-l10n/bionic 5.2.1-1ubuntu2 all
transitional package for kdevelop-php-l10n
```

...

Para aprender mais sobre o que cada módulo faz, pode pesquisar na internet para maiores informações. Alternativamente, veja a descrição longa do pacote digitando:

- `apt show nome_do_pacote`
-

Haverá uma grande quantidade de saída, com um campo chamado `Description` que terá uma explicação mais longa da funcionalidade que o módulo oferece.

Por exemplo, para encontrar o que o módulo `php-cli` faz, você pode digitar isto:

- `apt show php-cli`

Juntamente com várias outras informações, irá encontrar algo parecido com isto:

Output

...

Description: command-line interpreter for the PHP scripting language
(default)

This package provides the `/usr/bin/php` command interpreter, useful for testing PHP scripts from a shell or performing general shell scripting tasks.

.

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

.

This package is a dependency package, which depends on Ubuntu's default PHP version (currently 7.2).

...

Se, após pesquisar, decidirmos que gostaríamos de instalar um pacote, isso pode ser feito com o comando `apt install` assim como fizemos para nossos outros softwares.

Se decidir que o `php-cli` é algo necessário a instalar, é só digitar:

- `sudo apt install php-cli`

Se quisermos instalar mais de um módulo, podemos fazer isso listando cada um, separado por um espaço, seguindo o comando `apt install`, como no exemplo abaixo:

- `sudo apt install pacote1 pacote2 ...`

Neste ponto a plataforma LAMP está instalada e configurada. Antes de mais, seria útil testar proactivamente a configuração do PHP,

Step 4 — Testando o Processamento PHP no Servidor Web

A fim de testar se o sistema está corretamente configurado para o PHP, vamos criar um script PHP bem básico denominado `info.php`. Para que o Apache possa encontrar o arquivo e servi-lo corretamente, ele deve ser guardado em um diretório muito específico, o qual é chamado de “web root”.

No Ubuntu, este diretório está localizado em `/var/www/html`. Crie o arquivo neste local digitando:

- `sudo nano /var/www/html/info.php`

Isto vai abrir um ficheiro em branco. Coloque o seguinte texto, que é um código PHP válido, dentro do arquivo:

info.php

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

Quando tiver concluído, salve e feche o ficheiro.


Agora podemos testar se o servidor web exibir corretamente o conteúdo gerado por esse script PHP.

Para testar isso, visite a página no seguinte link:

<http://localhost/info.php>


A página que deve ver deve se parecer com isto:

PHP Version 7.2.3-1ubuntu1



| | |
|---|---|
| System | Linux LAMP-1804-test 4.15.0-15-generic #16-Ubuntu SMP Wed Apr 4 13:58:14 UTC 2018 x86_64 |
| Build Date | Mar 14 2018 22:03:58 |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php/7.2/apache2 |
| Loaded Configuration File | /etc/php/7.2/apache2/php.ini |
| Scan this dir for additional .ini files | /etc/php/7.2/apache2/conf.d |
| Additional .ini files parsed | /etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-openssl.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-curl.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gd.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-intl.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini, /etc/php/7.2/apache2/conf.d/20-xmlrpc.ini |
| PHP API | 20170718 |
| PHP Extension | 20170718 |
| Zend Extension | 320170718 |
| Zend Extension Build | API320170718.NTS |
| PHP Extension Build | API20170718.NTS |
| Debug Build | no |
| Thread Safety | disabled |
| Zend Signal Handling | enabled |
| Zend Memory Manager | enabled |
| Zend Multibyte Support | disabled |
| IPv6 Support | enabled |
| DTrace Support | available, disabled |
| Registered PHP Streams | https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar |
| Registered Stream Socket Transports | tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2 |
| Registered Stream Filters | zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.* |

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
with Zend OPcache v7.2.3-1ubuntu1, Copyright (c) 1999-2018, by Zend Technologies



Configuration

apache2handler

| | |
|----------------------|--|
| Apache Version | Apache/2.4.29 (Ubuntu) |
| Apache API Version | 20120211 |
| Server Administrator | webmaster@localhost |
| Hostname:Port | 162.243.26.126:80 |
| User/Group | www-data(33)/33 |
| Max Requests | Per Child: 0 - Keep Alive: on - Max Per Connection: 100 |
| Timeouts | Connection: 300 - Keep-Alive: 5 |
| Virtual Server | Yes |
| Server Root | /etc/apache2 |
| Loaded Modules | core mod_so mod_watchdog http_core mod_log_config mod_logio mod_version mod_unixd mod_access_compat mod_alias mod_auth_basic mod_auth_core mod_authn_file mod_authz_core mod_authz_host mod_authz_user mod_autoindex mod_deflate mod_dir mod_env mod_filter mod_mime prefork mod_negotiation mod_php7 mod_reqtimeout mod_setenvif mod_status |

Esta página fornece informações básicas sobre o servidor a partir da perspectiva do PHP.

Cofinanciado por:

Página 9 de 10

É útil para depuração e para assegurar que as configurações estão sendo corretamente aplicadas.

Se consegue ver esta página em seu navegador, então o PHP está funcionando como esperado.

Para remover este arquivo depois do teste, pois ele pode fornecer informações sobre o servidor para utilizadores não autorizados, execute o seguinte comando:

- `sudo rm /var/www/html/info.php`

IMPLEMENTAÇÃO DE NOVO EXEMPLO

Crie um ficheiro **exemplo2.php**:

- `sudo nano /var/www/html/exemplo2.php`

```
<!DOCTYPE html>
<html>
<body>

<?php
echo "Olá Mundo. Sim, sou eu mesmo!";
?>

</body>
</html>
```

Teste o acesso à sua página acedendo ao link: <http://localhost/exemplo2.php>

Conclusão

Agora que você tem uma pilha LAMP instalada, você terá muitas opções para o que fazer em seguida. Basicamente, com esta atividade foi instalada uma plataforma que irá permitir criar e configurar vários tipos de sites e aplicações web no servidor