

Tutorial: Nginx com PHP 7 e MySQL no Ubuntu 16.04 LTS

Date : 21 de Abril de 2017

Quando necessitamos de criar sites/conteúdos Web, temos de possuir um ambiente de desenvolvimento que tenha suporte para as tecnologias nas quais vamos desenvolver os nossos sites.

No Pplware já apresentamos alguns servidores/plataformas Web mas o nosso tutorial de hoje tem como objetivo ensinar a instalar o Nginx com suporte para PHP 7 e MySQL 5.7. Vamos a isso?



O Nginx é um servidor HTTP (Web) gratuito, open-source e com alta performance. O Nginx foi desenvolvido pelo programador russo Igor Sysoev em 2005 e tinha como principal objetivo ser um servidor estável, simples de configurar e que necessitasse de poucos recursos ao nível de hardware.

Como instalar o Nginx + PHP7 e MySQL5.7 no Ubuntu

Vamos considerar que a máquina onde vamos instalar o servidor web tem o IP: **192.168.1.90**

Para começar vamos abrir o terminal e uma vez que vamos necessitar de privilégios de root

(várias vezes) vamos executar o seguinte comando:

```
su
```

Passo 1 – Instalar o MySQL 5.7

```
apt-get -y install mysql-server mysql-client
```

Durante a instalação devem indicar uma password de acesso ao MySQL.

Devem ainda garantir ainda a melhor segurança do serviço. Para tal executem o seguinte comando:

```
/usr/bin/mysql_secure_installation
```

Durante a configuração, devem seguir os seguintes passos:

- Enter current password for root (enter for none): **# DAR ENTER #**
- Set root password? [Y/n] **# DAR ENTER #**
- New password: **# Redefinir Password #**

- Re-enter new password: **# Redefinir Password #**
- Remove anonymous users? [Y/n] **Y**
- Disallow root login remotely? [Y/n] **Y**
- Remove test database and access to it? [Y/n] **Y**
- Reload privilege tables now? [Y/n] **Y**

Passo 2 – Instalar o Nginx

Primeiro devem desinstalar por completo o apache2

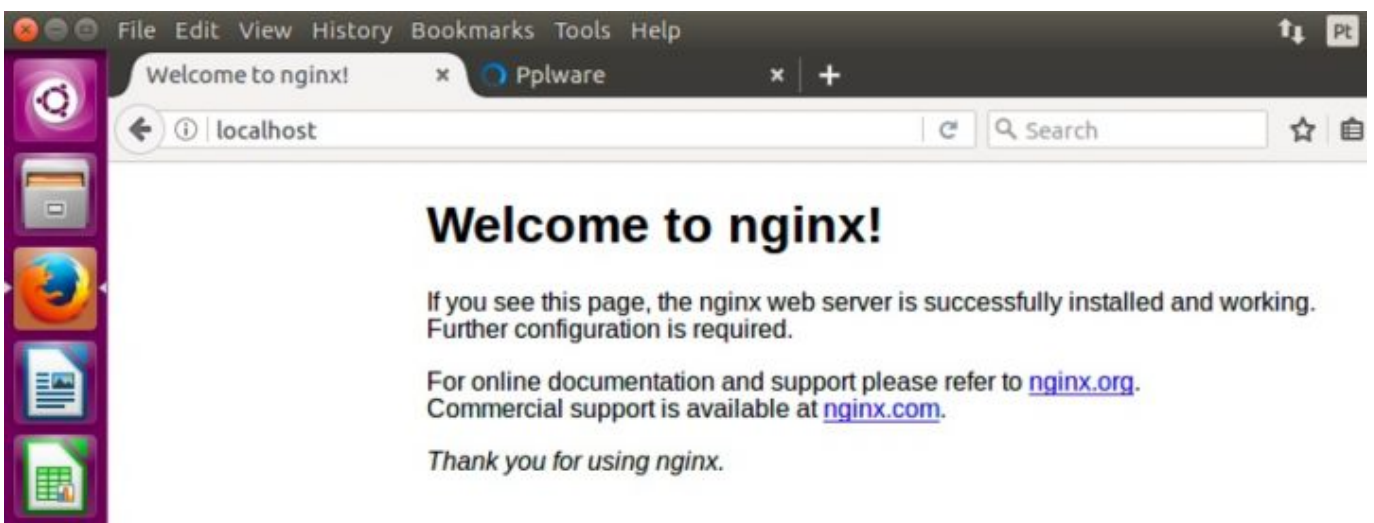
```
service apache2 stop update-rc.d -f apache2 remove apt-get remove apache2
```

Instalar o Nginx no Ubuntu é muito simples. Para tal basta inserir no terminal o seguinte comando:

```
apt-get install nginx
```

Para iniciar o nginx devem executar o seguinte comando:

```
service nginx start
```



Nota: No Ubuntu, a directoria por omissão para colocar os sites é /var/www/html.

Passo 3 – Instalar o PHP7

Para que o PHP7 funcione correctamente no Ubuntu é necessário instalar o [PHP-FPM](#) (PHP-FPM (FastCGI Process Manager) que é uma alternativa ao PHP FastCGI. Para instalar o php7-fpm basta executar o seguinte comando:

```
apt-get -y install php7.0-fpm
```

Configuração

Vamos agora proceder a algumas configurações a nível de vhosts. Para isso vamos abrir o vhost por omissão que se encontra em /etc/nginx/sites-available/default e definir as seguintes alterações:

```
[...]
server {
listen 80 default_server;
listen [::]:80 default_server;

# SSL configuration
#
# listen 443 ssl default_server;
# listen [::]:443 ssl default_server;
#
# Note: You should disable gzip for SSL traffic.
# See: https://bugs.debian.org/773332
#
# Read up on ssl_ciphers to ensure a secure configuration.
# See: https://bugs.debian.org/765782
#
# Self signed certs generated by the ssl-cert package
# Don't use them in a production server!
#
# include snippets/snakeoil.conf;

root /var/www/html;

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;
```

```
location / {
# First attempt to serve request as file, then
# as directory, then fall back to displaying a 404.
try_files $uri $uri/ =404;
}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
#
location ~ \.php$ {
include snippets/fastcgi-php.conf;

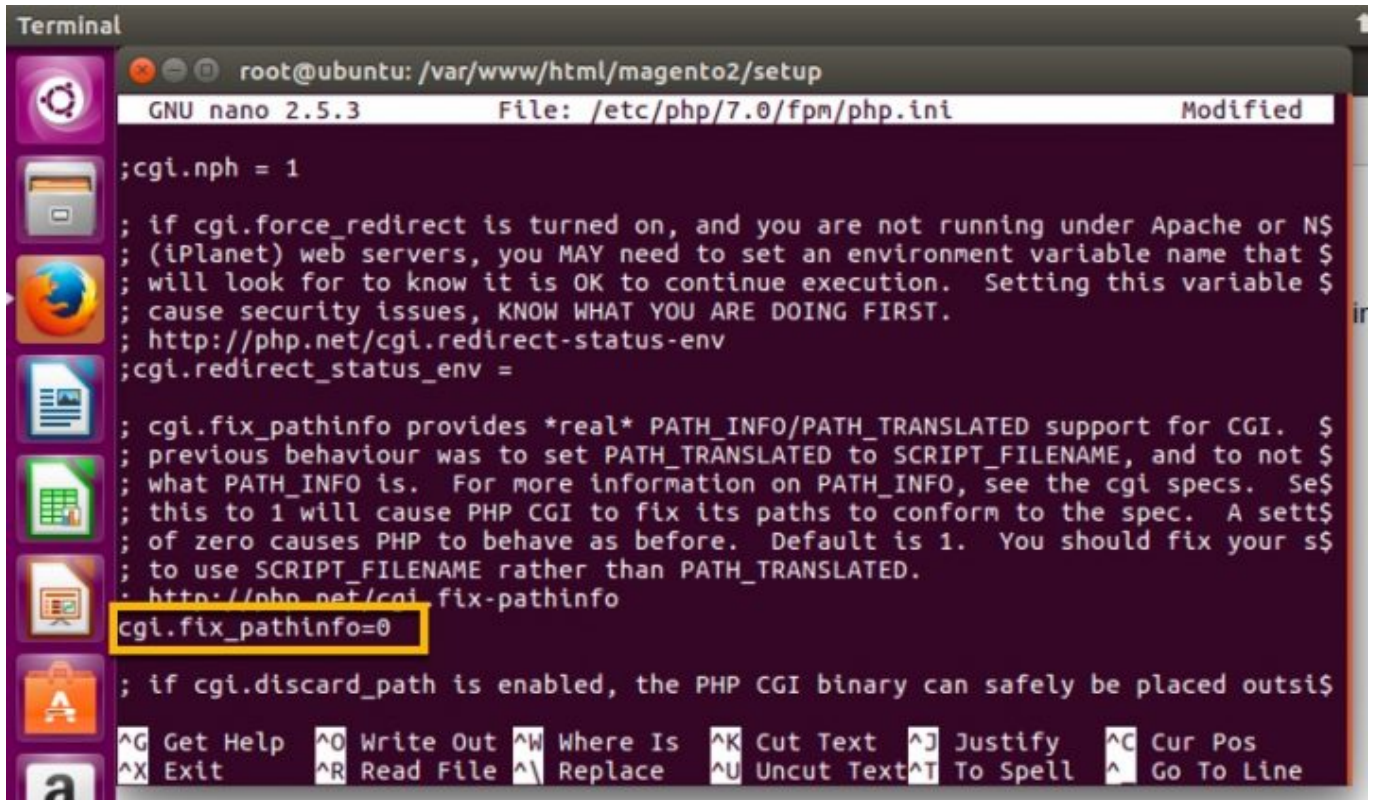
# With php7.0-cgi alone:
# fastcgi_pass 127.0.0.1:9000;
# With php7.0-fpm:
fastcgi_pass unix:/run/php/php7.0-fpm.sock;
}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
location ~ /\.ht {
deny all;
}
}
[...]
```

Depois de realizadas as configurações devemos recarregá-las executando para isso o comando:

```
service nginx reload
```

Devem ainda aceder ao ficheiro `/etc/php/7.0/fpm/php.ini` e definir **`cgi.fix_pathinfo=0`**;



```
Terminal
root@ubuntu: /var/www/html/magento2/setup
GNU nano 2.5.3 File: /etc/php/7.0/fpm/php.ini Modified
;cgi.nph = 1
; if cgi.force_redirect is turned on, and you are not running under Apache or NS
; (iPlanet) web servers, you MAY need to set an environment variable name that $
; will look for to know it is OK to continue execution. Setting this variable $
; cause security issues, KNOW WHAT YOU ARE DOING FIRST.
; http://php.net/cgi.redirect-status-env
;cgi.redirect_status_env =
; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for CGI. $
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not $
; what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. Se$
; this to 1 will cause PHP CGI to fix its paths to conform to the spec. A sett$
; of zero causes PHP to behave as before. Default is 1. You should fix your s$
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; http://php.net/cgi.fix_pathinfo
cgi.fix_pathinfo=0
; if cgi.discard_path is enabled, the PHP CGI binary can safely be placed outsi$
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Depois da alterações fazemos reload ao PHP-FPM

```
service php7.0-fpm reload
```

Para testar vamos criar um ficheiro info.php

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

e colocar o seguinte código:

Em seguida abrimos o browser e inserimos o endereço <http://localhost/info.php>. O resultado deverá ser idêntico ao que aparece na imagem seguinte:

| | |
|--|---|
| PHP Version 7.0.15-0ubuntu0.16.04.4 | |
| System | Linux ubuntu 4.8.0-36-generic #36~16.04.1-Ubuntu SMP Sun Feb 5 09:39:57 UTC 2017 x86_64 |
| Server API | FPM/FastCGI |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php/7.0/fpm |
| Loaded Configuration File | /etc/php/7.0/fpm/php.ini |
| Scan this dir for additional .ini files | /etc/php/7.0/fpm/conf.d |
| Additional .ini files parsed | /etc/php/7.0/fpm/conf.d/10-mysqld.ini, /etc/php/7.0/fpm/conf.d/10-opcache.ini, /etc/php/7.0/fpm/conf.d/10-pdo.ini, /etc/php/7.0/fpm/conf.d/15-xml.ini, /etc/php/7.0/fpm/conf.d/20-calendar.ini, /etc/php/7.0/fpm/conf.d/20-ctype.ini, /etc/php/7.0/fpm/conf.d/20-curl.ini, /etc/php/7.0/fpm/conf.d/20-dom.ini, /etc/php/7.0/fpm/conf.d/20-exif.ini, /etc/php/7.0/fpm/conf.d/20-fileinfo.ini, /etc/php/7.0/fpm/conf.d/20-ftp.ini, /etc/php/7.0/fpm/conf.d/20-gd.ini, /etc/php/7.0/fpm/conf.d/20-gettext.ini, /etc/php/7.0/fpm/conf.d/20-iconv.ini, /etc/php/7.0/fpm/conf.d/20-intl.ini, /etc/php/7.0/fpm/conf.d/20-json.ini, /etc/php/7.0/fpm/conf.d/20-mbstring.ini, /etc/php/7.0/fpm/conf.d/20-mcrypt.ini, /etc/php/7.0/fpm/conf.d/20-mysqli.ini, /etc/php/7.0/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.0/fpm/conf.d/20-phar.ini, /etc/php/7.0/fpm/conf.d/20-posix.ini, /etc/php/7.0/fpm/conf.d/20-readline.ini, /etc/php/7.0/fpm/conf.d/20-shmop.ini, /etc/php/7.0/fpm/conf.d/20-simplexml.ini, /etc/php/7.0/fpm/conf.d/20-soap.ini, /etc/php/7.0/fpm/conf.d/20-sockets.ini, /etc/php/7.0/fpm/conf.d/20-sysmsg.ini, /etc/php/7.0/fpm/conf.d/20-syssem.ini, /etc/php/7.0/fpm/conf.d/20-sysvshm.ini, /etc/php/7.0/fpm/conf.d/20-tokenizer.ini, /etc/php/7.0/fpm/conf.d/20-wddx.ini, /etc/php/7.0/fpm/conf.d/20-xmlreader.ini, /etc/php/7.0/fpm/conf.d/20-xmlwriter.ini, /etc/php/7.0/fpm/conf.d/20-xsl.ini, /etc/php/7.0/fpm/conf.d/20-zip.ini |
| PHP API | 20151012 |
| PHP Extension | 20151012 |
| Zend Extension | 320151012 |
| Zend Extension Build | API320151012.NTS |
| PHP Extension Build | API20151012.NTS |
| Debug Build | no |
| Thread Safety | disabled |
| Zend Signal Handling | disabled |

Passo 4 – Configurar o PHP para suporte com MySQL

Para que o PHP tenha suporte para MySQL devem executar o seguinte comando:

```
apt-get -y install php7.0-mysql php7.0-curl php7.0-gd php7.0-intl php-pear php-imagick php7.0-imap php7.0-mcrypt php-memcache php7.0-pspell php7.0-recode php7.0-sqlite3 php7.0-tidy php7.0-xmlrpc php7.0-xsl php7.0-mbstring php-gettext
```

Para finalizar voltem a fazer reload ao PHP-FPM

```
service php7.0-fpm reload
```

Acedendo a <http://localhost/info.php> já devem ver que agora o servidor nginx já tem suporte para MySQL. E está feito. Como tiveram a oportunidade de ver a instalação/configuração do

servidor Nginx requer algum trabalho (assim como outros servidores Web), mas no final o trabalho vale bem a pena já que o Nginx demonstrou uma boa performance e estabilidade. Ficamos a aguardar o vosso feedback.