

## Настройка веб-сервера в связке с MySQL и PHP7 на Ubuntu 16.04

*Якимов Антон Сергеевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

*Баженов Руслан Иванович*

*Приамурский государственный университет имени Шолом-Алейхема*

*Кандидат педагогических наук, доцент, зав. кафедрой информационных систем, математики и методик обучения*

### **Аннотация**

В данной статье рассматривается процесс установки MySQL и PHP7 компонентов на операционной системе Ubuntu 16.04, а также настройка PHP с Nginx сервером.

**Ключевые слова:** веб-сервер, ubuntu, mysql, php, nginx

## **Configuring of web server in conjunction with MySQL and PHP7 on Ubuntu 16.04**

*Yakimov Anton Sergeevich*

*Sholom-Aleichem Priamursky State University*

*Student*

*Bazhenov Ruslan Ivanovich*

*Sholom-Aleichem Priamursky State University*

*Candidate of pedagogical sciences, associate professor, Head of the Department of Information systems, Mathematics and teaching methods*

### **Abstract**

This article describes the process of installing MySQL and PHP7 components on Ubuntu 16.04 operating system, and configuring PHP with Nginx server.

**Keywords:** web server, ubuntu, mysql, php, nginx

MySQL и PHP являются популярными технологиями для построения различных приложений. В текущей статье мы рассмотрим установку данных компонентов на операционной системе Ubuntu 16.04, где за основу веб-сервера лежит Nginx модуль.

С данной темой уже изучали Ю.И. Рядчин и Ф.Г. Петров. Они устанавливали MySQL на сервер [1]. А.С.Кулябин и др.. разработали свою систему на основе СУБД MySQL [2]. И многие другие исследователи также ознакомились с данной тематикой [3-8].

Прежде чем начать устанавливать пакеты, нужно сначала обновить все пакеты, которые имеются на данном сервере, для того чтобы повысить безопасность и устранить сбои для установки последующих пакетов. Для обновления пакетов необходимо ввести следующую строку в командной строке сервера Ubuntu 16.04:

```
sudo apt-get update
```

Теперь приступим к установке базы данных MySQL. Для этого нужно ввести следующую команду:

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

Во время процесса установки MySQL пакета, сервер предложит вам установить пароль для администратора под логином «root»:

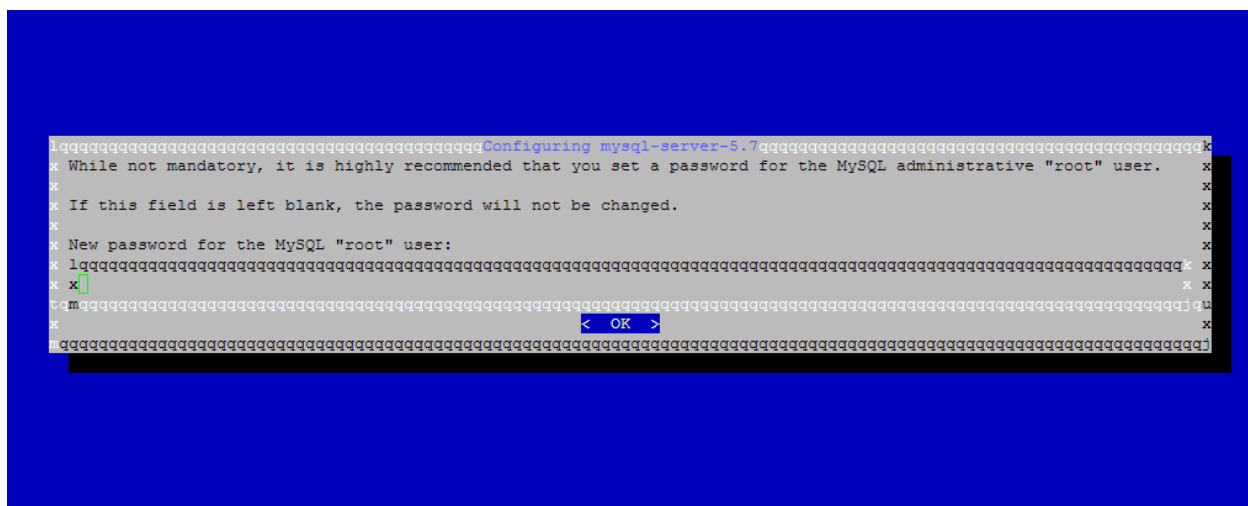


Рисунок 1 – Ввод пароля MySQL-пользователя

После ввода пароля, нужно применить его с помощью клавиши Enter, и вам предложат еще раз ввести свой пароль для надежности:

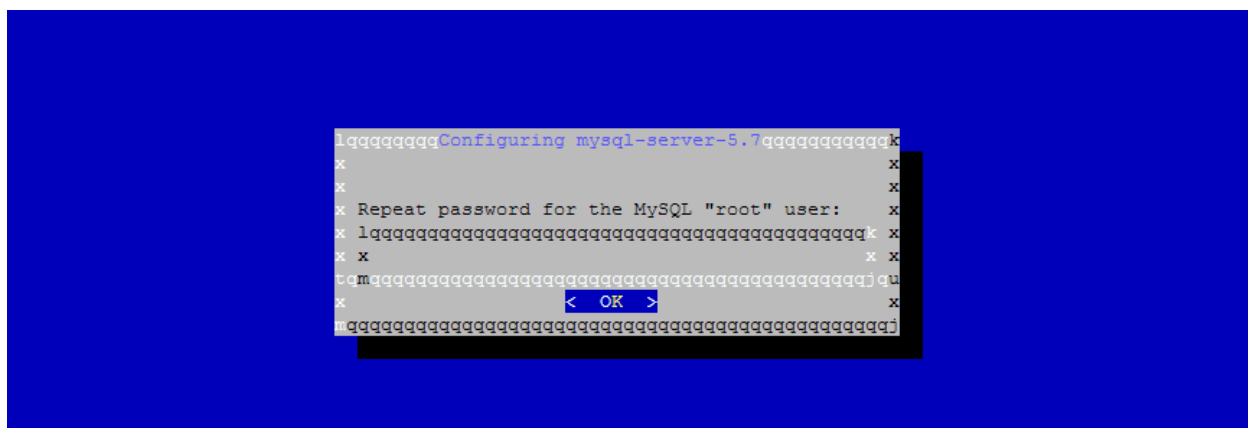


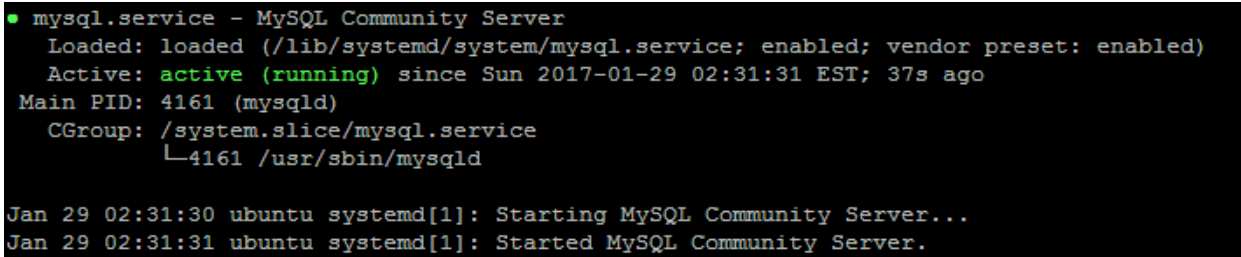
Рисунок 2 – Повторный ввод пароля MySQL-пользователя

Если вы вдруг ошиблись в повторных вводах пароля, то MySQL вам предоставит еще раз ввести новый пароль, и подтвердить ее.

После подтверждения пароля, сервер закончит свою установку MySQL пакета, и в дальнейшем можно будет использовать ее в работоспособном состоянии. Чтобы проверить, что MySQL работает, нужно ввести следующую команду:

```
sudo systemctl status mysql.service
```

Если все сделали верно, то данная команда отобразит примерно следующее содержание:



```
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2017-01-29 02:31:31 EST; 37s ago
 Main PID: 4161 (mysqld)
    CGroup: /system.slice/mysql.service
           └─4161 /usr/sbin/mysqld

Jan 29 02:31:30 ubuntu systemd[1]: Starting MySQL Community Server...
Jan 29 02:31:31 ubuntu systemd[1]: Started MySQL Community Server.
```

Рисунок 3 – Статус MySQL модуля

В данной информации особое внимание следует обратить на строку «Active: active (running) ... », которая означает, что MySQL запущен и находится в работоспособном состоянии.

Попробуем протестировать MySQL сервер. Введем следующую строку, которая позволит войти в MySQL под именем корневого пользователя «root»:

```
sudo mysql -u root -p
```

Сервер, после ввода данной команды, попросит вас ввести пароль от MySQL, а точнее, от «root» пользователя, которое вы ввели выше, во время установки данного пакета. После того как введете пароль, вы попадете в MySQL среду. Теперь приступим к созданию базы данных. Для этого необходимо ввести следующие данные:

```
create database test_database;
```

Затем вводим следующий синтаксис:

```
show databases;
```

После ввода вышеприведенных строк, мы увидим следующую картину:

```
mysql> create database test_database;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test_database |
+-----+
5 rows in set (0.00 sec)
```

Рисунок 4 – Тестирование базы данных

Как видим, база данных «*test\_database*» успешно создана, и мы убедились в работоспособности MySQL пакета.

Теперь нашим следующим шагом будет установка PHP7 модуля. Чтобы установить PHP7 пакет, нужно сначала выйти из MySQL среды:

*exit*

А затем вписать следующую команду для установки самого PHP7:

*sudo apt-get install php7.0*

После ввода текущей команды, сервер установит все необходимые компоненты и зависимости PHP7 пакета. Теперь далее нужно изменить настройку конфигурации файла *php.ini*. Для этого открываем файл для редактирования с помощью следующей командой:

*sudo vi /etc/php/7.0/fpm/php.ini*

В консоли, с помощью редактора Vim, откроется конфигурационный файл *php.ini*. В данном файле необходимо найти строку «*cgi.fix\_pathinfo*»:

```
; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for CGI. PHP's
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not grok
; what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. Setting
; this to 1 will cause PHP CGI to fix its paths to conform to the spec. A setting
; of zero causes PHP to behave as before. Default is 1. You should fix your scripts
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; http://php.net/cgi.fix-pathinfo
;cgi.fix_pathinfo=1
```

Рисунок 5 – Строка «*cgi.fix\_pathinfo*»

Данную строку нужно раскомментировать, убрав символ «*;*» в самом начале и установить значение данного параметра в «*0*». Чтобы редактировать в редакторе Vim, нужно нажать на клавишу *i*, и после этого редактор войдет в режим редактирования. После того как закончите редактировать,

необходимо нажать на клавишу **Esc**, В итоге строка должна иметь следующий вид:

```
; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for CGI. PHP's
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not grok
; what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. Setting
; this to 1 will cause PHP CGI to fix its paths to conform to the spec. A setting
; of zero causes PHP to behave as before. Default is 1. You should fix your scripts
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; http://php.net/cgi.fix-pathinfo
cgi.fix_pathinfo=0
```

Рисунок 6 – Исправленная строка «cgi.fix\_pathinfo»

Затем необходимо ввести следующую строку:

```
:wq
```

Данная строка сохраняет текущий файл и закрывает ее за собой. Чтобы внесенные изменения конфигурационного файла вступили в силу, надо перезагрузить процесс PHP:

```
sudo systemctl restart php7.0-fpm
```

Далее необходимо настроить Nginx сервер. Для этого откроем файл с настройками виртуального хоста:

```
sudo vi /etc/nginx/sites-available/default
```

Откроется файл со следующим содержимым:

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

    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;
    #}
}
```

Рисунок 7 – Содержимое файла default

И вносим в ней следующие изменения:

1. После директивы *index* необходимо на первом месте добавить значение «*index.php*». Это нужно для того, чтобы Nginx в первую очередь обрабатывал PHP файлы для генерации HTML-страницы.
2. Присвоить значение IP-адреса вашего сервера к директиве *server\_name*. Чтобы узнать IP-адрес, нужно ввести команду «*hostname -i*», и после этого покажется IP-адрес вашего настраиваемого сервера.
3. Раскомментировать блок, в котором описаны настройки PHP обработок.

В итоге должно получиться следующее содержимое файла:

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

    root /var/www/html;

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

    server_name 89.23.94.184;

    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;
    }
}
```

Рисунок 7 – Исправленное содержимое файла default

Сохраняем файл и выходим из редактора Vim. Далее можно проверить конфигурационный файл на наличие ошибок:

```
sudo nginx -t
```

Если во всех вышеприведенных процедурах сделали правильно, то у вас должно отобразиться следующее:

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Рисунок 8 –Nginx статус об наличии ошибок

Данное сообщение говорит о том, что в конфигурационном файле отсутствуют синтаксические ошибки, и полностью исправен к дальнейшей работе. Для того чтобы измененный файл вступил в силу, нужно перезагрузить Nginx сервер:

*sudo systemctl reload nginx*

Теперь остался последний шаг, это проверка работоспособности PHP компонента. Попробуем создать простую PHP-страницу и вывести его в браузере. Для этого нужно перейти в нужный каталог и создать новый файл:

```
sudo vi /var/www/html/test.php
```

В открывшемся пустом редакторе впишем следующий PHP код:

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

Рисунок 9 – Образец кода

Сохраняем файл и выходим из редактора. Затем перейдем в браузер по адресу «<http://10.10.10.10/test.php>». Только в данном случае, вместо «10.10.10.10» нужно ввести IP адрес вашего сервера. Если сделали все правильно, то увидим следующую страницу:

PHP Version 7.0.13-0ubuntu0.16.04.1	
System	Linux ubuntu 2.6.32-042stab113.11 #1 SMP Fri Dec 18 17:32:04 MSK 2015 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-opcache.ini, /etc/php/7.0/fpm/conf.d/10-pdo.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-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-gettext.ini, /etc/php/7.0/fpm/conf.d/20-iconv.ini, /etc/php/7.0/fpm/conf.d/20-json.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-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-sysshm.ini, /etc/php/7.0/fpm/conf.d/20-tokenizer.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
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	enabled
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

Рисунок 10 – Страница phpinfo()

На этом настройка сервера заканчивается. В данной статье мы рассмотрели пошаговую установку MySQL и PHP пакетов, ознакомились с



редактором Vim и научились настраивать конфигурационные файлы PHP модуля и Nginx веб-сервера.

### Библиографический список

1. Петров Ф.Г., Рядчин Ю.И. Установка и конфигурирование web-сервера Apache, интерпретатора PHP и сервера баз данных MySQL // Научно-технический вестник информационных технологий, механики и оптики. 2006. №29 С.121-126.
2. Ураксеев М.А., Левина Т.М., Шамаев Ф.Ф., Кулябин А.С. Разработка волоконно-оптических систем для учета, мониторинга и прогнозирования работы высоковольтного оборудования в СУБД с web-интерфейсом // Электротехнические и информационные комплексы и системы. 2015. №1 С.97-102.
3. Токарчук А.М. Применение грид-систем при развертывании web-сайта // Информационно-управляющие системы. 2010. №3 С.51-55.
4. Буренков А. О. Разработка и внедрение систем технологического учета и контроля с использованием web-технологий // Литьё и металлургия. 2014. №3 (76) С.74-80.
5. Бухаров Д.С. О некоторых особенностях веб-сервера IIS // Инновационная наука. 2015. №6-2 С.35-39.
6. Ботыгин И.А., Каликин К.А. Исследование методов увеличения производительности WEB-приложений // Известия ТПУ. 2008. №5 С.109-114.
7. Дмитриевич Г.Д., Мохсен А.А, Ларистов А.И. Архитектура Web-ориентированных САПР // Информационно-управляющие системы. 2010. №5 С.20-23.
8. Титов А.Ф., Вдовицын В.Т., Крышень А.М., Лебедев В.А., Полин А.К. Современные информационные технологии и развитие сети особо охраняемых природных территорий (на примере Республики Карелия) // Труды КарНЦ РАН. 2011. №1 С.82-86.