

Ubuntu 16.04.1 Üzerine NGINX, PHP-FPM ve MariaDB Kurulumu

if

Ocak, 2017

İçindekiler

1	NGINX kurulumu	2
2	PHP-FPM Kurulumu	3
3	MariaDB Kurulumu	4

Ubuntu Türkiye Takımı olarak sunucularımızda NGINX web sunucusu, PHP'nin FastGCI uygulaması olan FastGCI Süreç Yöneticisi (FPM) ve MariaDB veri tabanı yönetimi sistemini kullanmaktayız. Bu makalede bu uygulamaların Ubuntu'nun 16.04.1 sürümüne kurulumu anlatılmaktadır.

1 NGINX kurulumu

Ubuntu'nun 16.04.1 sürümünde, NGINX'in 1.10 kararlı sürümünü sunmaktadır. Kurulum için aşağıdaki komutu uygulamanız yeterli.

```
1 sudo apt install nginx
2 Aşağıdaki YENİ paketler kurulacak:
3 nginx nginx-common nginx-core
```

Kurulumun ardından NGINX servisinin başladığını teyit etmek için aşağıdaki komuyu uygulayın.

```
1 systemctl status nginx.service
2 nginx.service — A high performance web server and a reverse proxy server
3   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
4   Active: active (running) since Prş 2017-01-14 21:46:37 +03; 35min ago
5   Process: 1224 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
6   Process: 1168 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
7   Main PID: 1226 (nginx)
8   CGroup: /system.slice/nginx.service
9           1226 nginx: master process /usr/sbin/nginx -g daemon on; master_process on
10          1227 nginx: worker process
11          1228 nginx: worker process
12          1229 nginx: worker process
13          1230 nginx: worker process
14 Oca 14 21:46:33 lts systemd[1]: Starting A high performance web server and a reverse proxy server...
15 Oca 14 21:46:37 lts systemd[1]: Started A high performance web server and a reverse proxy server.
```

Servis yüklenmiş (loaded) ve çalışır vaziyette (active) ise devam edebilirsiniz. Değilse servisi

```
1 sudo systemctl restart nginx.service
```

komutu ile başlatmayı deneyin. NGINX'in TCP protokolünün HTTP için 80 ve HTTPS için 443 portları üzerinden IPv4 ve IPv6 adreslerinden gelen istekleri kabul edebilmesi için güvenlik duvarı uygulamanız aracılığıyla bu portların açılması gerekmektedir. Güvenlik duvarı yapılandırması bu anlatımın dışında olduğu için burada anlatılmayacaktır. Fakat kolaylık olması açısından UFW için temel komutlar aşağıda verilmiştir.

```
1 sudo ufw default deny incoming
2 sudo ufw default allow outgoing
3 sudo ufw allow 'Nginx HTTP'
4 sudo ufw allow 'Nginx HTTPS'
```

Portların NGINX tarafından kullanıldığını görmek için

```
1 sudo netstat -tlnp | grep nginx
```

komutunu kullanabilirsiniz. Tarayıcının adres satırına *localhost* yazdığınızda NGINX'in "hoşgeldin" sayfasıyla karşılaşmanız gerek.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

Şekil 1:

2 PHP-FPM Kurulumu

Ubuntu 16.04.1 sürümünde PHP'nin 7 sürümü kullanılmaya başlandı. PHP-FPM'yi kurmak için

```
1 sudo apt install php-fpm
2 Aşağıdaki YENİ paketler kurulacak:
3 php-common php7.0-cli php7.0-common php7.0-fpm
4 php7.0-json php7.0-opcache php7.0-readline
```

PHP-FPM servisinin başladığını teyit edin:

```
1 systemctl status php7.0-fpm.service
2 php7.0-fpm.service — The PHP 7.0 FastCGI Process Manager
3   Loaded: loaded (/lib/systemd/system/php7.0-fpm.service; enabled; vendor preset: enabled)
4   Active: active (running) since Prş 2017-01-14 21:56:38 +03; 50min ago
5   Process: 1166 ExecStartPre=/usr/lib/php/php7.0-fpm-checkconf (code=exited, status=0/SUCCESS)
6   Main PID: 1286 (php-fpm7.0)
7   Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic: 0req/sec"
8   CGroup: /system.slice/php7.0-fpm.service
9           1286 php-fpm: master process (/etc/php/7.0/fpm/php-fpm.conf)
10          1288 php-fpm: pool www
11          1289 php-fpm: pool www
12 Oca 14 21:56:33 lts systemd[1]: Starting The PHP 7.0 FastCGI Process Manager..
13 Oca 14 21:56:38 lts systemd[1]: Started The PHP 7.0 FastCGI Process Manager.
```

Eğer sunucunuzun ana/indeks sayfası *index.php* ile sunuluyorsa, **/etc/nginx/sites-available** dizini altındaki sunucu ayar dosyanızın server bloğunun index satırına *index.php* eklemeniz gerekmektedir. NGINX'in PHP betiklerini/dosyalarını PHP-FPM'ye iletmesi için yine ayar dosyanızın PHP için ayrılan location direktifinin aşağıdaki gibi yapılandırılması gerekmektedir.

```
1 location ~ /\.php$ {
2   include snippets/fastcgi-php.conf;
3   fastcgi_pass unix:/run/php/php7.0-fpm.sock;
4 }
```

PHP-FPM'nin PHP betiklerini/dosyalarını yorumlayabildiğini teyit etmek için öntanımlı sunucu dizinine PHP ayarlarıyla ilgili bilgi veren *phpinfo()* fonksiyonunu çalıştıracak bir PHP betiği ekleyin:

```
1 echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
```

Tarayıcının adres satırına *localhost/info.php* yazdığınız takdirde *phpinfo()* fonksiyonunun sağladığı bilgilere ulaşabilirsiniz gerek.

PHP Version 7.0.13-0ubuntu0.16.04.1	
System	Linux lts 4.4.0-57-generic #78-Ubuntu SMP Fri Dec 9 23:50:32 UTC 2016 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

Şekil 2:

Additional Modules	
Module Name	
Environment	
Variable	Value
USER	www-data
HOME	/var/www

Şekil 3:

3 MariaDB Kurulumu

Ubuntu'nun bu sürümünde MariaDB'in güncel kararlı sürümü 10 sunulmaktadır. Kurulum için MariaDB sunucu ve istemci paketlerinin yanı sıra PHP için gerekli modülleri sağlayan php-mysql paketinin de kurulması gerekmektedir.

```

1 sudo apt install mariadb-server mariadb-client php-mysql
2 Aşağıdaki YENİ paketler kurulacak:
3 libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl
4 libmysqlclient20 libreadline5 libterm-readkey-perl mariadb-client
5 mariadb-client-10.0 mariadb-client-core-10.0 mariadb-common mariadb-server
6 mariadb-server-10.0 mariadb-server-core-10.0 mysql-common php7.0-mysql

```

PHP-FPM servisini yeniden başlatın:

```

1 sudo systemctl restart php7.0-fpm.service

```

MariaDB'i güvenli kılmak için Ubuntu'nun sağladığı güvenlik betiğini çalıştırmak gerekmektedir. Betik, MariaDB root kullanıcısı için bir şifre belirlememizi, test amaçlı gelen öntanımlı kullanıcı ve veritabanının kaldırılmasını, root kullanıcısını kullanarak uzaktan veritabanına bağlanmanın iptal edilmesini isteyip bu isteklere verilen cevaplara göre izin tablolarını yeniden yükleyerek kurulumu tamamlayacaktır. Betiği çalıştırmadan önce root kullanıcısı için istenen şifreyi önceden oluşturunuz.

Güvenli bir şifre oluşturmak için **apg** uygulamasından faydalanacağız. Ayrı bir uçbirim açarak aşağıdaki komutu yürütün:

```

1 apg -a 1 -n 10 -m 16 -M SNCL -s
2 Please enter some random data (only first 16 are significant)
3 (eg. your old password):>

```

Komut sizden rast gele 16 adet değer girmenizi isteyecek. Girdikten sonra ENTER tuşuna basınca şifreleriniz üretilmiş olacak. Oluşturulan şifrelerden birini seçerek diğer uçbirimden betiği çalıştırın.

```
1 sudo mysql_secure_installation
2 NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
3 SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
4
5 In order to log into MariaDB to secure it, we'll need the current
6 password for the root user. If you've just installed MariaDB, and
7 you haven't set the root password yet, the password will be blank,
8 so you should just press enter here.
9
10 Enter current password for root (enter for none):
```

Daha önce kurulmuş bir MariaDB olmadığı için root kullanıcı için bir şifre de bulunmamaktadır. Bu yüzden ilk adımda sadece ENTER tuşuna basıp devam edebilirsiniz. Daha önce tanımladığınız bir MariaDB root kullanıcınız varsa, bu kullanıcıya ait şifreyi girmelisiniz.

```
1 Enter current password for root (enter for none):
2 OK, successfully used password, moving on...
```

Sonraki adımda betik sizden MariaDB root kullanıcısı için bir şifre belirlemenizi isteyecek. Seçtiğiniz şifreyi girip onaylayın:

```
1 Setting the root password ensures that nobody can log into the MariaDB
2 root user without the proper authorisation.
3
4 Set root password? [Y/n] y
5 New password:
6 Re-enter new password:
7 Password updated successfully!
8 Reloading privilege tables..
9 ... Success!
```

Öntanımlı olarak gelen kullanıcının kaldırılması istenecek. *Evet* deyip devam edin:

```
1 By default, a MariaDB installation has an anonymous user, allowing anyone
2 to log into MariaDB without having to have a user account created for
3 them. This is intended only for testing, and to make the installation
4 go a bit smoother. You should remove them before moving into a
5 production environment.
6
7 Remove anonymous users? [Y/n]
8 ... Success!
```

root hesabının sadece localhost üzerinden bağlanmasını sağlamak için bir sonraki adımda da evet deyip devam edin:

```
1 Normally, root should only be allowed to connect from 'localhost'. This
2 ensures that someone cannot guess at the root password from the network.
3
4 Disallow root login remotely? [Y/n]
5 ... Success!
```

Test veritabanının kaldırılması istenecek, kaldırın:

```
1 By default, MariaDB comes with a database named 'test' that anyone can
2 access. This is also intended only for testing, and should be removed
3 before moving into a production environment.
4
5 Remove test database and access to it? [Y/n]
6 - Dropping test database...
7 ... Success!
8 - Removing privileges on test database...
9 ... Success!
```

Bu zamana kadar yapılan deęişikliklerin kaydedilmesi istenecek:

```
1  Reloading the privilege tables will ensure that all changes made so far
2  will take effect immediately.
3
4  Reload privilege tables now? [Y/n]
5  ... Success!
6
7  Cleaning up...
8
9  All done! If you've completed all of the above steps, your MariaDB
10 installation should now be secure.
11
12 Thanks for using MariaDB!
```

Son adımdan sonra MariaDB kurulumu da tamamlanmış oldu.