

Урок 6. Запуск стека для веб-приложения

Задание:

- * Установить Nginx и настроить его на работу с PHP-FPM.
 - * Установить Apache. Настроить обработку PHP. Добиться одновременной работы с Nginx.
 - * Настроить схему обратного прокси для Nginx (динамика - на Apache).
 - * Установить MySQL. Создать новую базу данных и таблицу в ней.
 - ** Установить пакет phpmyadmin и запустить его веб-интерфейс для управления MySQL.
 - ** Настроить схему балансировки трафика между несколькими серверами Apache на стороне Nginx с помощью модуля ngx_http_upstream_module.
- ИЛИ установить и настроить WORDPRESS

- * Установить Nginx и настроить его на работу с PHP-FPM.

Для установки Nginx и настройки его на работу с PHP-FPM необходимо выполнить следующие шаги:

1. Установить Nginx, используя установочный пакет или исходные коды.

Установим пакеты, необходимые для подключения apt-репозитория:

```
sudo apt install curl gnupg2 ca-certificates lsb-release ubuntu-keyring
```

Теперь нужно импортировать официальный ключ, используемый apt для проверки подлинности пакетов. Скачаем ключ:

```
curl https://nginx.org/keys/nginx_signing.key | gpg --dearmor \
| sudo tee /usr/share/keyrings/nginx-archive-keyring.gpg >/dev/null
```

Проверим, верный ли ключ был загружен:

```
gpg --dry-run --quiet --no-keyring --import --import-options import-
show /usr/share/keyrings/nginx-archive-keyring.gpg
```

Вывод команды должен содержать полный отпечаток ключа
573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62:

```
pub rsa2048 2011-08-19 [SC] [expires: 2024-06-14]
573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62
uid nginx signing key <signing-key@nginx.com>
```

Если отпечаток отличается от вышеуказанного, удалим файл ключа.

Для подключения apt-репозитория для стабильной версии nginx, выполним следующую команду:

```
echo "deb [signed-by=/usr/share/keyrings/nginx-archive-
keyring.gpg] \
```

```
http://nginx.org/packages/ubuntu `lsb_release -cs` nginx" \  
| sudo tee /etc/apt/sources.list.d/nginx.list
```

Чтобы установить nginx, выполним следующие команды:

```
sudo apt update  
sudo apt install nginx
```

Установим модуль php:

```
sudo apt install php7.2
```

* Установить Apache. Настроить обработку PHP. Добиться одновременной работы с Nginx.

Установим модули Apache для корректной установки и работы Wordpress

```
sudo apt update  
sudo apt install apache2 \  
ghostscript \  
libapache2-mod-php \  
mysql-server \  
php \  
php-bcmath \  
php-curl \  
php-imagick \  
php-intl \  
php-json \  
php-mbstring \  
php-mysql \  
php-xml \  
php-zip
```

2. После установки необходимо настроить конфигурационный файл Nginx.

Для этого откроем файл nginx.conf и добавим следующие директивы:

```
vim nginx.conf
```

Вставляем туда:

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

Нажимаем Esc для выхода из режима редактирования, затем набираем команду :wq и Enter для сохранения файла конфигурации.

3. Настроим PHP-FPM для использования с Nginx. Для этого откроем php-fpm.conf и добавим следующие директивы:

```
listen = /run/php/php7.2-fpm.sock listen.mode = 0666
```

4. Перезапустим Nginx и PHP-FPM для применения изменений.

5. Проверим доступность PHP из Nginx на локальной машине, создав файл test.php в домашней директории с содержимым:

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

Затем откройте этот файл в браузере и убедитесь, что все работает правильно.

Установим Wordpress

```
sudo mkdir -p /srv/www
```

```
sudo chown www-data: /srv/www
```

```
curl https://wordpress.org/latest.tar.gz | sudo -u www-data tar zx -  
C /srv/www
```

Создадим сайт Apache для Wordpress

/etc/apache2/sites-available/wordpress.conf с помощью следующих строк:

```
<VirtualHost *:80>
```

```
DocumentRoot /srv/www/wordpress
```

```
<Directory /srv/www/wordpress>
```

```
Options FollowSymLinks
```

```
AllowOverride Limit Options FileInfo
```

```
DirectoryIndex index.php
```

```
Require all granted
```

```
</Directory>
```

```
<Directory /srv/www/wordpress/wp-content>
```

```
Options FollowSymLinks
```

```
Require all granted
```

```
</Directory>
```

```
</VirtualHost>
```

Включим сайт:

```
sudo a2ensite wordpress
```

Включим перезапись URL:

```
sudo a2enmod rewrite
```

Отключим сайт по-умолчанию:

```
sudo a2dissite 000-default
```

Далее я запутался с Апачем и решил настроить сайт на nginx:

Установим MariaDB базу данных:

```
sudo apt-get install mariadb-server mariadb-client
```

После установки MariaDB следующие команды можно использовать для остановки , запуска и включения службы MariaDB, чтобы она всегда запускалась при загрузке сервера:

```
sudo systemctl stop mariadb.service
```

```
sudo systemctl start mariadb.service
```

```
sudo systemctl enable mariadb.service
```

Затем запустим приведенные ниже команды, чтобы защитить сервер базы данных паролем root, если не было предложено сделать это во время установки.

```
sudo mysql_secure_installation
```

Чтобы проверить и убедиться, что MariaDB установлена и работает, войдём в консоль базы данных, используя следующие команды:

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

Войдём в консоль базы данных MariaDB, используя следующие команды:

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

Затем создадим базу данных с именем wpdb

```
CREATE DATABASE wpdb;
```

Создадим пользователя базы данных с именем wpdbuser и установим пароль

```
CREATE USER 'wpdbuser'@'localhost' IDENTIFIED BY  
'new_password_here';
```

Затем предоставим пользователю полный доступ к базе данных.

```
GRANT ALL ON wpdb.* TO 'wpdbuser'@'localhost' WITH GRANT  
OPTION;
```

Наконец, сохраним изменения и выйдем.

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

На этом этапе WordPress готов к загрузке и установке. Используйте команды ниже, чтобы загрузить последнюю версию WordPress.

```
cd /tmp
```

```
wget https://wordpress.org/latest.tar.gz
```

```
tar -xvzf latest.tar.gz
sudo mv wordpress /var/www/wordpress
```

Затем запустим команду ниже, чтобы позволить www-data пользователю владеть каталогом WordPress

```
sudo chown -R www-data:www-data /var/www/wordpress/
sudo chmod -R 755 /var/www/wordpress/
```

Создадим и отредактируем файл конфигурации WordPress wp-config.php.

```
cd /var/www/html/wordpress
mv wp-config-sample.php wp-config.php
nano wp-config.php
```

Отредактируем информацию о соединении с базой данных MySQL, расположенную на wp-config.file.

Проверяем работоспособность сайта.