

- Урок 6
 - Запуск стека для веб-приложения
 - Задания
 - Балансировщик
 - 1. gb-12 (мастер)
 - 2. gb-13 (слейв)
 - 3. gb-14 (слейв)

Урок 6

Запуск стека для веб-приложения

Задания

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

```
sudo su
apt -y install nginx-extras
systemctl start nginx
```



⚠ Небезопасно | 192.168.1.12

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.

```
apt -y install php-fpm curl
sed -i "/server_name _;/a  \\\n \
\t location ~ \\.php$ { \n \
\t\t include snippets/fastcgi-php.conf; \n \
\t\t root /var/www/html; \n \
\t\t fastcgi_pass unix:/run/php/php8.1-fpm.sock; \n \
\t }" /etc/nginx/sites-available/default
systemctl restart nginx
echo "<?php phpinfo() ?>" | sudo tee /var/www/html/info.php
```

PHP Version 8.1.2-1ubuntu2.14




System	Linux gb-12 5.15.0-1048-kvm #53-Ubuntu SMP Sun Nov 19 15:56:32 UTC 2023 x86_64
Build Date	Aug 18 2023 11:41:11
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/fpm
Loaded Configuration File	/etc/php/8.1/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/fpm/conf.d
Additional .ini files parsed	/etc/php/8.1/fpm/conf.d/10-opcache.ini, /etc/php/8.1/fpm/conf.d/10-pdo.ini, /etc/php/8.1/fpm/conf.d/20-calendar.ini, /etc/php/8.1/fpm/conf.d/20-ctype.ini, /etc/php/8.1/fpm/conf.d/20-exif.ini, /etc/php/8.1/fpm/conf.d/20-ffi.ini, /etc/php/8.1/fpm/conf.d/20-fileinfo.ini, /etc/php/8.1/fpm/conf.d/20-ftp.ini, /etc/php/8.1/fpm/conf.d/20-gettext.ini, /etc/php/8.1/fpm/conf.d/20-iconv.ini, /etc/php/8.1/fpm/conf.d/20-phar.ini, /etc/php/8.1/fpm/conf.d/20-posix.ini, /etc/php/8.1/fpm/conf.d/20-readline.ini, /etc/php/8.1/fpm/conf.d/20-shmop.ini, /etc/php/8.1/fpm/conf.d/20-sockets.ini, /etc/php/8.1/fpm/conf.d/20-sysvmsg.ini, /etc/php/8.1/fpm/conf.d/20-sysvsem.ini, /etc/php/8.1/fpm/conf.d/20-sysvshm.ini, /etc/php/8.1/fpm/conf.d/20-tokenizer.ini

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

```
apt -y install apache2
sed -i 's/Listen 80/Listen 8080/g' /etc/apache2/ports.conf
systemctl restart apache2
```

40 | 192.168.1.12:8080



Ubuntu

Apache2 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:

```
apt -y install libapache2-mod-php8.1
systemctl restart apache2
echo "<?php phpinfo() ?>" | sudo tee /var/www/html/info.php
```

3. ◦ Настроить схему обратного прокси для Nginx (динамика - на Apache).

```
rm -f /etc/nginx/sites-available/default
cat << EOF >> /etc/nginx/sites-available/default
server {
    listen 80 default_server;
    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;
    server_name _;
    location ~* ^.+.(jpg|jpeg|gif|png|ico|css|html|zip|pdf|txt|tar|js)\$ {
        root /var/www/html;
    }
    location ~ .php\$ {
        proxy_pass http://gb:8080;
        proxy_set_header Host \$host;
        proxy_set_header X-Forwarded-For
        \$proxy_add_x_forwarded_for;
        proxy_set_header X-Real-IP \$remote_addr;
        try_files \$uri \$uri/ =404;
    }
}
EOF
systemctl restart nginx
```

192.168.1.12/index.nginx-debian.html

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.

Небезопасно | 192.168.1.12/info.php



PHP Version 8.1.2-1ubuntu2.14



System	Linux gb-12 5.15.0-1048-kvm #53-Ubuntu SMP Sun Nov 19 15:56:32 UTC 2023 x86_64
Build Date	Aug 18 2023 11:41:11
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini

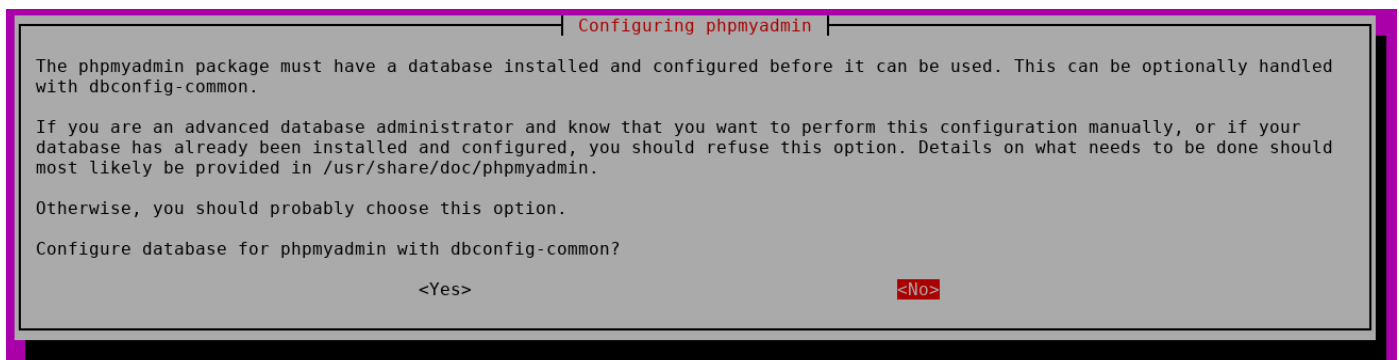
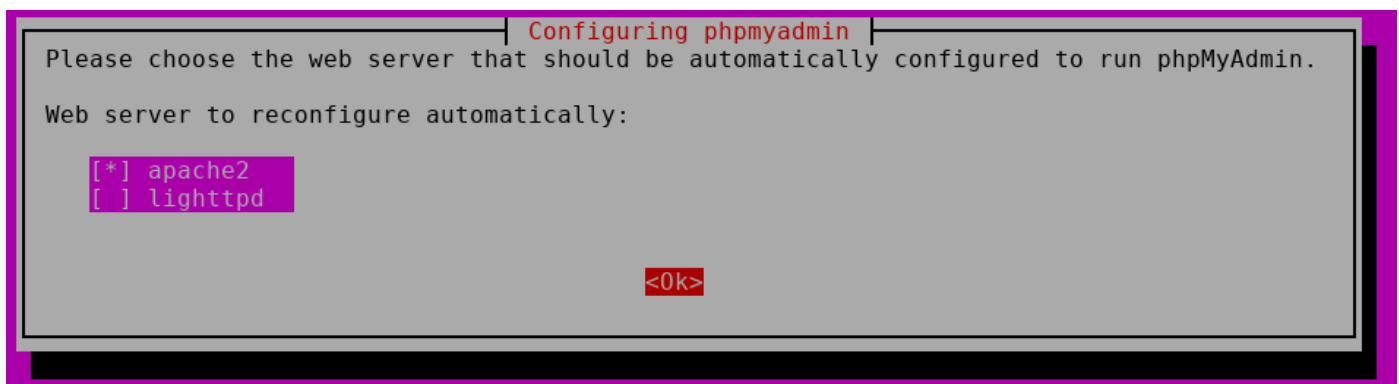
4. ◦ Установить MySQL. Создать новую базу данных и таблицу в ней.

```
apt -y install mysql-client mysql-server
systemctl start mysql
```

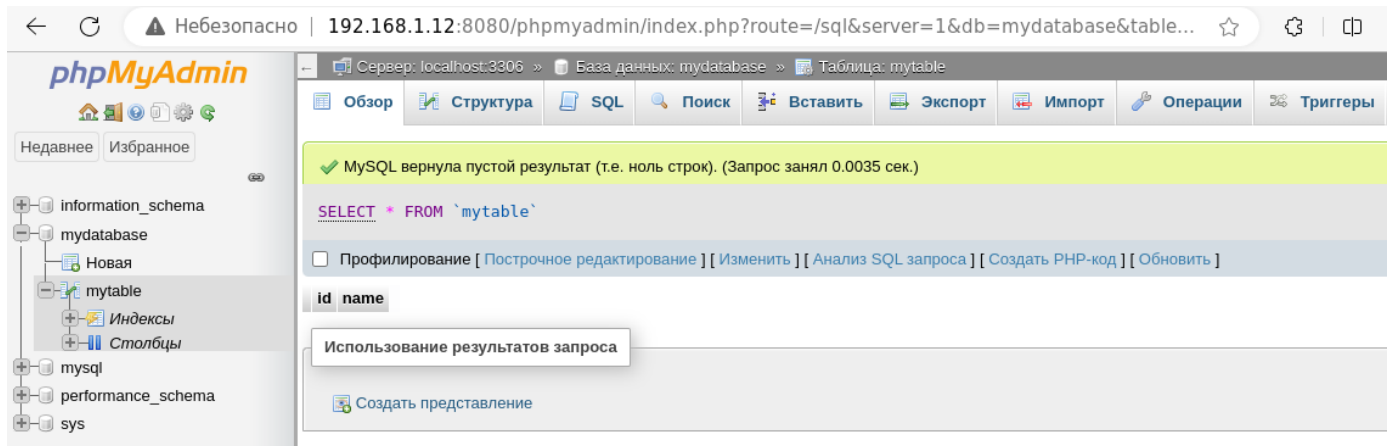
```
mysql -uroot << MYSQL_SCRIPT
CREATE DATABASE mydatabase;
USE mydatabase;
CREATE TABLE mytable (id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, name
VARCHAR(50));
DESCRIBE mytable;
MYSQL_SCRIPT
```

5. ** Установить пакет phpmyadmin и запустить его веб-интерфейс для управления MySQL.

```
apt -y install phpmyadmin
```



```
mysql -uroot << MYSQL_SCRIPT
ALTER USER 'user'@'%' IDENTIFIED BY '12345';
GRANT ALL PRIVILEGES ON *.* TO 'user'@'%';
FLUSH PRIVILEGES;
MYSQL_SCRIPT
```



Балансировщик

6. *** Настроить схему балансировки трафика между несколькими серверами Apache на стороне Nginx с помощью модуля ngx_http_upstream_module.

Подготовим три машины с ubuntu для серверов nginx (пусть будет без апачей и php) на примере статике

1. gb-12 (мастер)

```
cat << EOF >> /etc/nginx/conf.d/upstream.conf
upstream gb{
    server 192.168.1.13;
    server 192.168.1.14;
}

log_format custom '\$proxy_host \$upstream_addr [$request]'
                'status $status $body_bytes_sent '
                '"\$http_referer" "\$http_user_agent"';

access_log /var/log/nginx/upstream.log custom;
EOF

rm -f /etc/nginx/sites-available/default
cat << EOF >> /etc/nginx/sites-available/default
server {
    listen 80 default_server;
    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;
    server_name _;
    location / {
        proxy_pass http://gb;
        try_files \$uri \$uri/ =404;
    }
}
EOF
systemctl restart nginx
```

2. gb-13 (слейв)

```
sudo su
apt -y install nginx-extras
systemctl start nginx
apt -y install php-fpm curl
rm -f /etc/nginx/sites-available/default
cat << EOF >> /etc/nginx/sites-available/default
server {
    listen 80 default_server;
    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;
    server_name _;
    location / {
        try_files $uri $uri/ =404;
    }
}
EOF

systemctl restart nginx
echo "<?php phpinfo() ?>" | sudo tee /var/www/html/info.php
```

3. gb-14 (слейв)

```
sudo su
apt -y install nginx-extras
systemctl start nginx
apt -y install php-fpm curl
rm -f /etc/nginx/sites-available/default
cat << EOF >> /etc/nginx/sites-available/default
server {
    listen 80 default_server;
    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;
    server_name _;
    location ~ .php$ {
        include snippets/fastcgi-php.conf;
        root /var/www/html;
        fastcgi_pass unix:/run/php/php8.1-fpm.sock;
    }
    location / {
        try_files $uri $uri/ =404;
    }
}
EOF

systemctl restart nginx
echo "<?php phpinfo() ?>" | sudo tee /var/www/html/info.php
```

Можно протестировать доступность ресурса, поочередно отключая nginx на gb-14

```
systemctl stop nginx
```

⚠ Небезопасно | 192.168.1.12

Welcome to gb-13

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.

И на первом резервном сервере gb-13

```
systemctl stop nginx
```

192.168.1.12

Welcome to gb-14

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.

При обращении на мастер сервер gb-12, видно поочередную смену заголовка страницы gb-13 или gb-14 - это nginx перенаправляет трафик в зависимости от доступности на тот или иной слейв.

Но, интереснее распределение нагрузки на сервера с помощью дополнительных модулей балансировки, которые используют разные методы и алгоритмы.

Модуль ngx_http_upstream использует метод **round-robin** не учитывает время отклика серверов. Даже если один из серверов отвечает медленнее из-за высокой нагрузки, запросы все равно будут отправляться на этот сервер.

Для преодоления этих недостатков можно рассмотреть другие методы балансировки нагрузки в Nginx, такие как `least_conn`, `ip_hash`, или использование `ngx_http_upstream_fair_module`, который учитывает загруженность серверов при распределении запросов. Именно поэтому я выбрал `ngx_http_upstream_fair_module` для балансировщика.