

ДЗ по "Операционные системы и виртуализация (Linux) (семинары)"

Веб-сервер на Linux

Оглавление

Домашнее задание	4
1 Начало	5
1.1 Коннектимся к Ubuntu через ssh соединение, используя терминал PowerShell.....	5
2 Установка Nginx	6
2.1 Устанавливаем пакет nginx	6
2.2 Удостоверимся, что сервер запустился	6
2.3 Попробуем загрузить стартовую страницу WEB сервера в консоли.....	6
2.4 Можно загрузить из браузера	6
3 Устанавливаем и настраиваем Apache.....	7
3.1 Устанавливаем пакет apache2	7
3.2 Редактируем файлы конфигурации ports.conf.....	7
3.3 Проверяем корректность конфигурационных файлов	7
3.4 Запускаем Apache	7
3.5 Проверяем запущена ли задача apache	7
3.6 Посмотрим сокеты на прослушивание.....	8
3.7 Редактируем файл 000-default.conf	8
3.8 Проверяем корректность конфигурационных файлов	8
3.9 Перезапускаем сервер, чтобы изменения в конфигурации применились	8
3.10 Проверка работы Apache. Загружаем стартовую страницу.....	8
4 Динамический и статический контент. Схема Reverse проху	9
4.1 Изменяем конфигурация сервера Nginix для работы схемы Reverse проху.....	9
4.2 Проверим корректность редактирования	10
4.3 Применим новую конфигурацию, перезапустим сервер Nginix	10
4.4 Проверим загрузку страницы в браузере	11
4.5 Проверим работу сервера, когда все запросы направляются на Apache	11
4.5.1 Отредактируем файл /etc/nginx/sites-enabled/default	11
4.5.2 Проверим корректность редактирования.....	12
4.5.3 Применим новую конфигурацию, сервера Nginix.....	12
4.5.4 Загрузим страницу через браузер.....	12
4.5.5 Проверка работы завершена, вернем конфигурацию в режим Reverse проху	13
5 Обработка PHP в Apache.....	14
5.1 Устанавливаем PHP с модулем Apache.....	14
5.2 Создаем файл PHP	14
5.3 Выходим из рута	14
5.4 Просмотр PHP в браузере	14
6 Обработка PHP в PHP-FPM.....	16
6.1 Установка сервера PHP-FPM	16
6.2 Проверяем список процессов	16
6.3 Настроим конфигурацию Nginx.....	16
6.4 Отредактируем файл /etc/nginx/sites-enabled/default	16
6.5 Проводим тест конфигурации	17
6.6 Применяем новую конфигурацию	17
6.7 Проверяем отработку PHP в браузере	18
7 СУБД MySQL.....	19
7.1 Установка сервера MySQL.....	19

7.2 Проверка запущенных задач	19
7.3 Посмотрим сокеты	19
7.4 Заходим в консоль MySQL	19
7.5 Информация о базах данных	19
7.6 Заходим в выбранную базу данных.....	19
7.7 Посмотрим какие таблицы там есть	19
7.8 Выполним запрос SQL	19
7.9 Создадим новую базу данных.....	19
7.10 Заходим в созданную базу данных.....	19
7.11 Создадим таблицу (сущность) в базе данных.....	20
7.12 Добавим данные в созданную таблицу	20
7.13 Посмотрим добавленные данные	20
7.14 Выходим из консоли MySQL, возвращаемся в консоль Linux.....	20
8 Установка пакета phpMyAdmin	21
8.1 Сделаем, чтобы все запросы отправлялись на сервер Apache	21
8.2 Обновим индекс пакетов	21
8.3 Установка пакетов.....	21
8.4 Настройка доступа для специально созданного пользователя MySQL.....	21
8.4.1 Войти в MySQL.....	21
8.4.2 Создать нового пользователя с именем 'uc' и паролем '1'	21
8.4.3 Предоставляем права доступа пользователю	21
8.4.4 Выходим из MySQL	22
8.5 Подключаем файл конфигурации phpMyAdmin к серверу Apache.....	22
8.6 Проверяем корректность конфигурационных файлов	22
8.7 Применяем изменения конфигурации	22
8.8 Зайдем в phpMyAdmin через WEB браузер	22
9 Балансировка трафика между несколькими серверами Apache на стороне Nginx с помощью модуля ngx_http_upstream_module	24
9.1 Клонирование виртуальной машины Linux	24
9.2 Изменение статического IP виртуальной машины.....	24
9.2.1 Коннектимся к виртуалке.....	24
9.2.2 Открываем через nano, файл сетевых настроек.....	24
9.2.3 Ставим другой статический IP	24
9.2.4 Применение сетевых настроек	25
9.2.5 Меняем сетевые настройки остальных виртуалок.....	25
9.3 Изменяем конфигурация сервера Nginix	25
9.4 Проверим корректность конфигурации	26
9.5 Применим новую конфигурацию	26
9.6 Проверим как балансируется трафик	26
Вывод	28
Литература.....	29

Домашнее задание

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

2. * Установить Apache. Настроить обработку PHP.

Добиться одновременной работы с Nginx

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

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

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

для управления MySQL

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

Результат

Текст команд, которые применялись при выполнении задания.

При наличии: часть конфигурационных файлов, которые решают задачу.

Присылаем в формате текстового документа: задание и команды для решения (без вывода). Формат — PDF (один файл на все задания).

1 Начало

1.1 Коннектимся к Ubuntu через ssh соединение, используя терминал PowerShell

У каждого свой IP и имя пользователя.

```
ssh uc@192.168.1.8
```

2 Установка Nginx

2.1 Устанавливаем пакет nginx

```
sudo apt install nginx
```

2.2 Удостоверимся, что сервер запустился

Просмотрим сокеты, которые слушаются.

```
sudo ss -ntlp
```

Посмотрим таблицу процессов.

```
ps afx
```

2.3 Попробуем загрузить стартовую страницу WEB сервера в консоли

```
curl localhost
```

2.4 Можно загрузить из браузера

В браузере вбиваем IP вашего сервера, и смотрим стартовую страничку (рисунок 1). Порт 80 должен быть открыт и задан статический IP (см. ДЗ семинара 5).



Рисунок 1. Стартовая страница Nginx сервера

3 Устанавливаем и настраиваем Apache

3.1 Устанавливаем пакет apache2

```
sudo apt install apache2
```

Apache не запустится, потому что он также сидит на 80-м порту как и Nginx, поэтому будем конфигурировать Apache.

3.2 Редактируем файлы конфигурации ports.conf

```
sudo nano /etc/apache2/ports.conf
```

Изменения подчеркнуты зеленым цветом (рисунок 2).

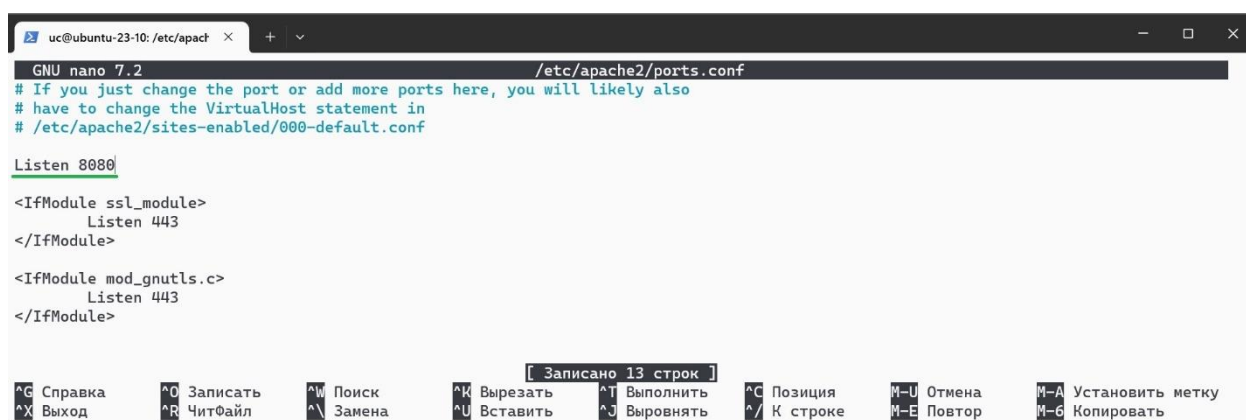


Рисунок 2. Изменение файла /etc/apache2/ports.conf. При выходе из редактора нажать Ctrl + O, Enter, Ctrl + X

3.3 Проверяем корректность конфигурационных файлов

```
sudo apachectl -t
```

3.4 Запускаем Apache

```
sudo systemctl start apache2
```

3.5 Проверяем запущена ли задача apache

```
ps afx
```

Еще можно так

```
sudo systemctl status apache2
```

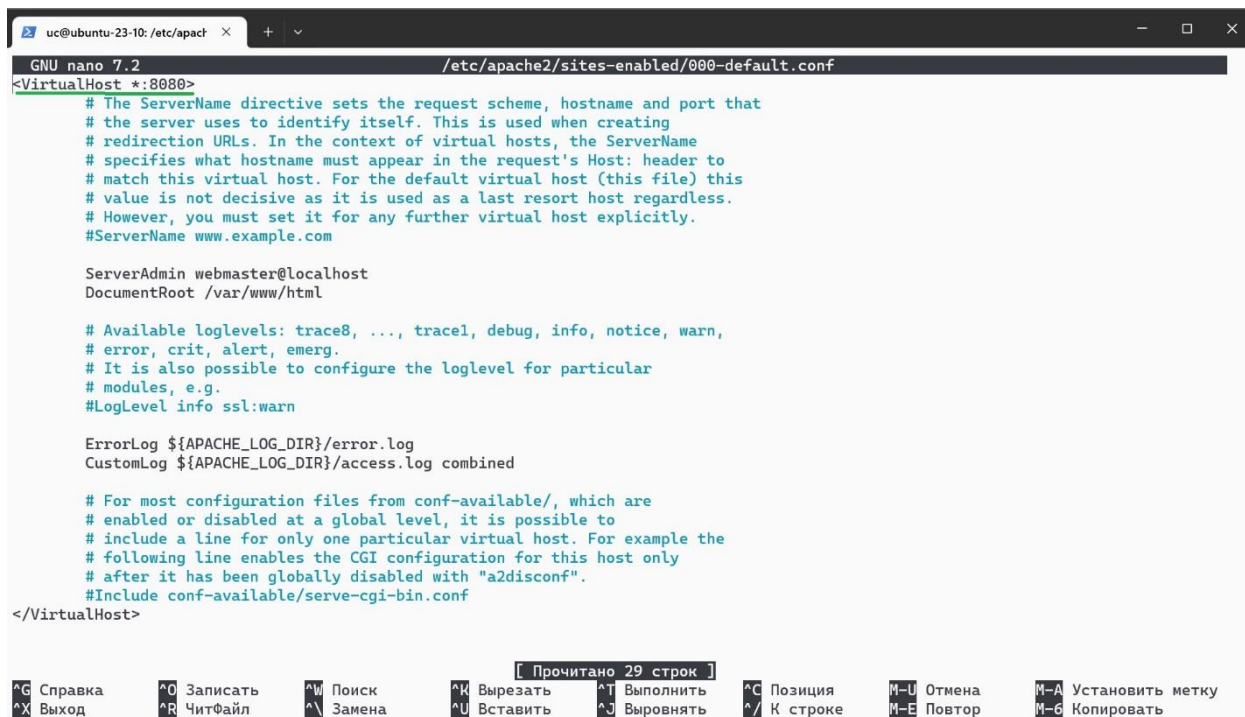
3.6 Посмотрим сокеты на прослушивание

```
sudo ss -ntlp
```

3.7 Редактируем файл 000-default.conf

```
sudo nano /etc/apache2/sites-enabled/000-default.conf
```

Изменения подчеркнуты зеленым цветом (рисунок 3).



```
GNU nano 7.2 /etc/apache2/sites-enabled/000-default.conf
<VirtualHost *:8080>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Рисунок 3. Изменение файла /etc/apache2/sites-enabled/000-default.conf. При выходе из редактора нажать Ctrl + O, Enter, Ctrl + X

3.8 Проверяем корректность конфигурационных файлов

```
sudo apachectl -t
```

3.9 Перезапускаем сервер, чтобы изменения в конфигурации применились

```
sudo systemctl reload apache2
```

3.10 Проверка работы Apache. Загружаем стартовую страницу

```
curl localhost:8080
```


4 Динамический и статический контент. Схема Reverse proxy

4.1 Изменяем конфигурация сервера Nginix для работы схемы Reverse proxy

```
sudo nano /etc/nginx/sites-enabled/default
```

Допишем в файл конфигурации следующий текст.

```
# Динамические запросы
location / {
    proxy_pass http://localhost: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;
}

# Статические запросы
location ~* ^.+.(jpg|jpeg|gif|png|ico|css|zip|pdf|txt|tar|js)$ {
    root /var/www/html;
}
```

Пропишем локейшены для динамических запросов, которые отправляются на Apache, статические запросы, которые будут обрабатываться на Nginix . Отредактированный файл с новой настройкой конфигурации – рисунок 4, изменения выделены зелёным.



```
uc@uc-linux: ~
GNU nano 6.2 /etc/nginx/sites-enabled/default *
# Default server configuration
#
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;
        # Динамические запросы
        proxy_pass http://localhost: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;
    }

    # pass PHP scripts to FastCGI server
    #
    #location ~ \.php$ {
    #    include snippets/fastcgi-php.conf;
    #    root /var/www/html;
    #
    #    # With php-fpm (or other unix sockets):
    #    fastcgi_pass unix:/run/php/php7.4-fpm.sock;
    #    # With php-cgi (or other tcp sockets):
    #    fastcgi_pass 127.0.0.1:9000;
    #}

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    # Статические запросы
    location ~* ^.+.(jpg|jpeg|gif|png|ico|css|zip|pdf|txt|tar|js)$ {
        root /var/www/html;
    }
}

^G Справка      ^O Записать    ^W Поиск       ^K Вырезать   ^T Выполнить  ^С Позиция     М-У Отмена     М-А Установить метку
^X Выход        ^R ЧитФайл    ^_ Замена      ^U Вставить   ^J Выводить    ^/ К строке    М-Е Повтор     М-6 Копировать
```

Рисунок 4. Настройка конфигурации схемы Reverse proxy. При выходе из редактора нажать Ctrl + O, Enter, Ctrl + X

4.2 Проверим корректность редактирования

```
sudo nginx -t
```

4.3 Применим новую конфигурацию, перезапустим сервер Nginix

```
sudo systemctl reload nginx
```

4.4 Проверим загрузку страницы в браузере

В браузере зададим IP нашего сервера и еще проверим функцию переадресации портов, с 8090 на 80 (рисунок 5). На сервер Nginx, отправляются только статические запросы, поэтому логотип не отображается, потому что картинка находится в Apache директории.

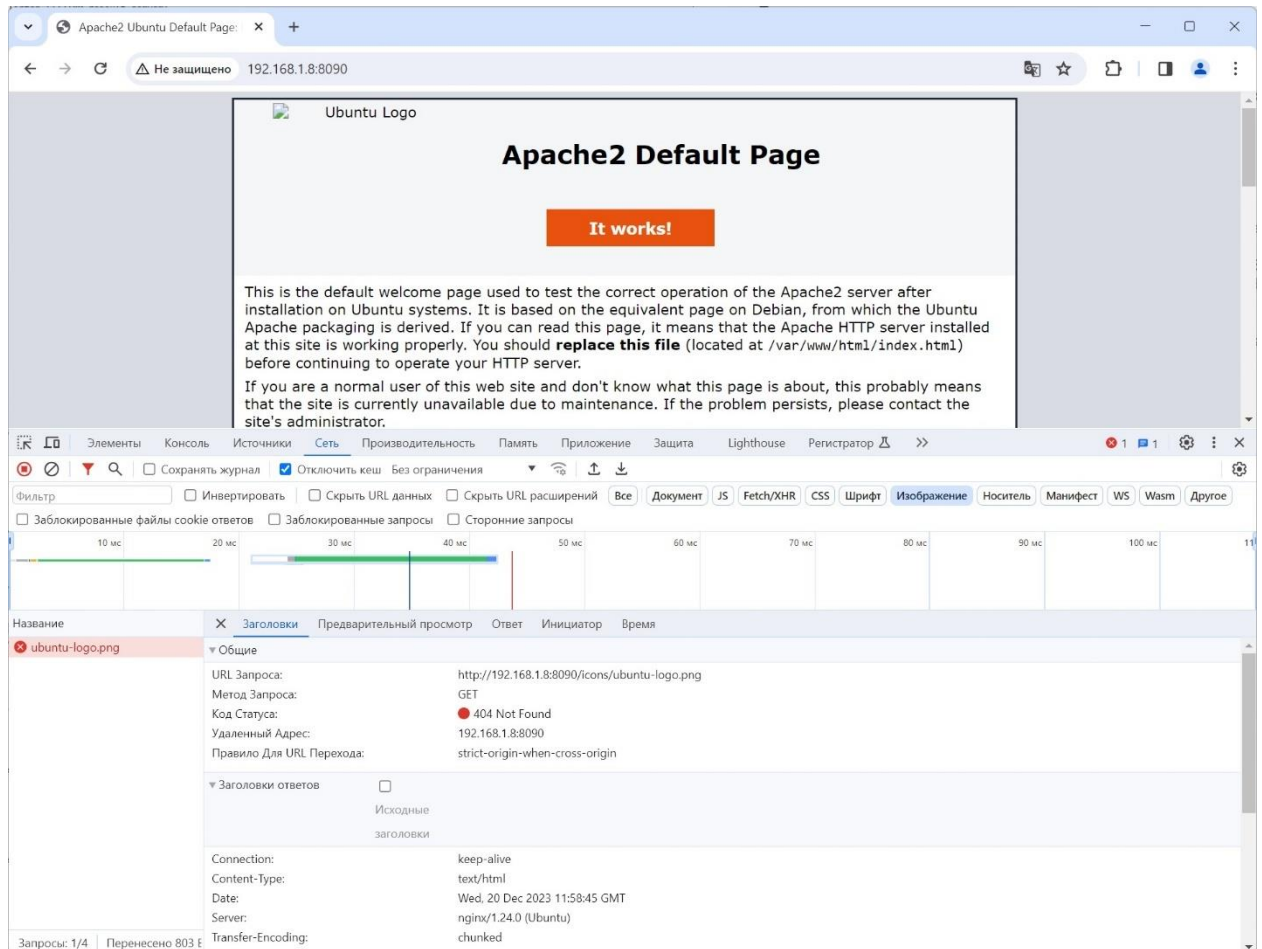


Рисунок 5. Работа серверов Nginx и Apache в схеме Reverse proxy

4.5 Проверим работу сервера, когда все запросы направляются на Apache

4.5.1 Отредактируем файл /etc/nginx/sites-enabled/default

```
sudo nano /etc/nginx/sites-enabled/default
```

Нужно закомментировать фрагмент в зеленом прямоугольнике, показанный на (рисунок 6)

```
GNU nano 6.2 /etc/nginx/sites-enabled/default *
# Default server configuration
#
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;
        # Динамические запросы
        proxy_pass http://localhost: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;

    }

    # pass PHP scripts to FastCGI server
    #
    #location ~ \.php$ {
    #    include snippets/fastcgi-php.conf;
    #    root /var/www/html;
    #
    #    # With php-fpm (or other unix sockets):
    #    fastcgi_pass unix:/run/php/php7.4-fpm.sock;
    #    # With php-cgi (or other tcp sockets):
    #    fastcgi_pass 127.0.0.1:9000;
    #}

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    # Статические запросы
    #location ~* ^.+.(jpg|jpeg|gif|png|ico|css|zip|pdf|txt|tar|js)$ {
    #root /var/www/html;
    #}

}

^G Справка      ^O Записать    ^W Поиск      ^K Вырезать   ^T Выполнить  ^C Позиция    M-U Отмена    M-A Установить метку
^X Выход      ^R ЧитФайл    ^M Замена    ^U Вставить   ^J Выровнять  ^/_ К строке  M-E Повтор    M-6 Копировать
```

Рисунок 6. Редактируем файл так, чтобы все запросы направлялись на Apache. При выходе из редактора нажать Ctrl + O, Enter, Ctrl + X

4.5.2 Проверим корректность редактирования

```
sudo nginx -t
```

4.5.3 Применим новую конфигурацию, сервера Nginix

```
sudo systemctl reload nginx
```

4.5.4 Загрузим страницу через браузер

В браузере зададим IP нашего сервера (рисунок 7).

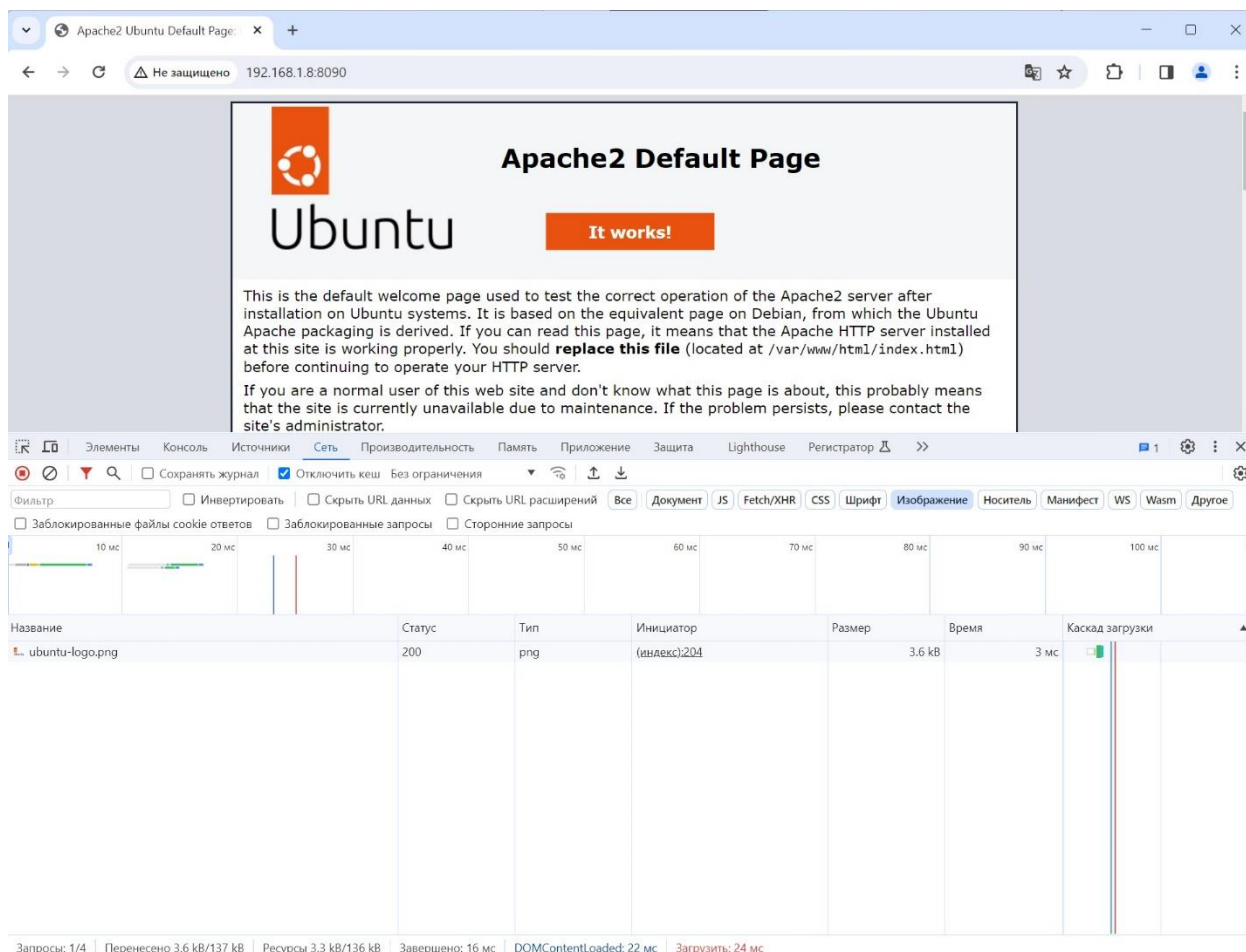


Рисунок 7. Проверка работы сервера Apache

Теперь все запросы отправляются на Apache (логотип загружается, потому что директория известна Apache)

4.5.5 Проверка работы завершена, вернем конфигурацию в режим Reverse proxy

Для этого выполним заново пункты 4.1 – 4.3 (то есть раскомментируем статические запросы).

5 Обработка PHP в Apache

5.1 Устанавливаем PHP с модулем Apache

```
sudo apt install php8.1 libapache2-mod-php8.1
```

5.2 Создаем файл PHP

Создаем и редактируем файл info.php. Для выхода из режима редактирования нажать Ctrl + D.

```
sudo su

cat > /var/www/html/info.php

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

5.3 Выходим из рута

```
exit
```

5.4 Просмотр PHP в браузере

Откроем файл PHP в браузере, вводим в адресную строку IP нашего сервера Ubuntu и название PHP файла (рисунок 8). Server API – Apache 2.0 Handler.

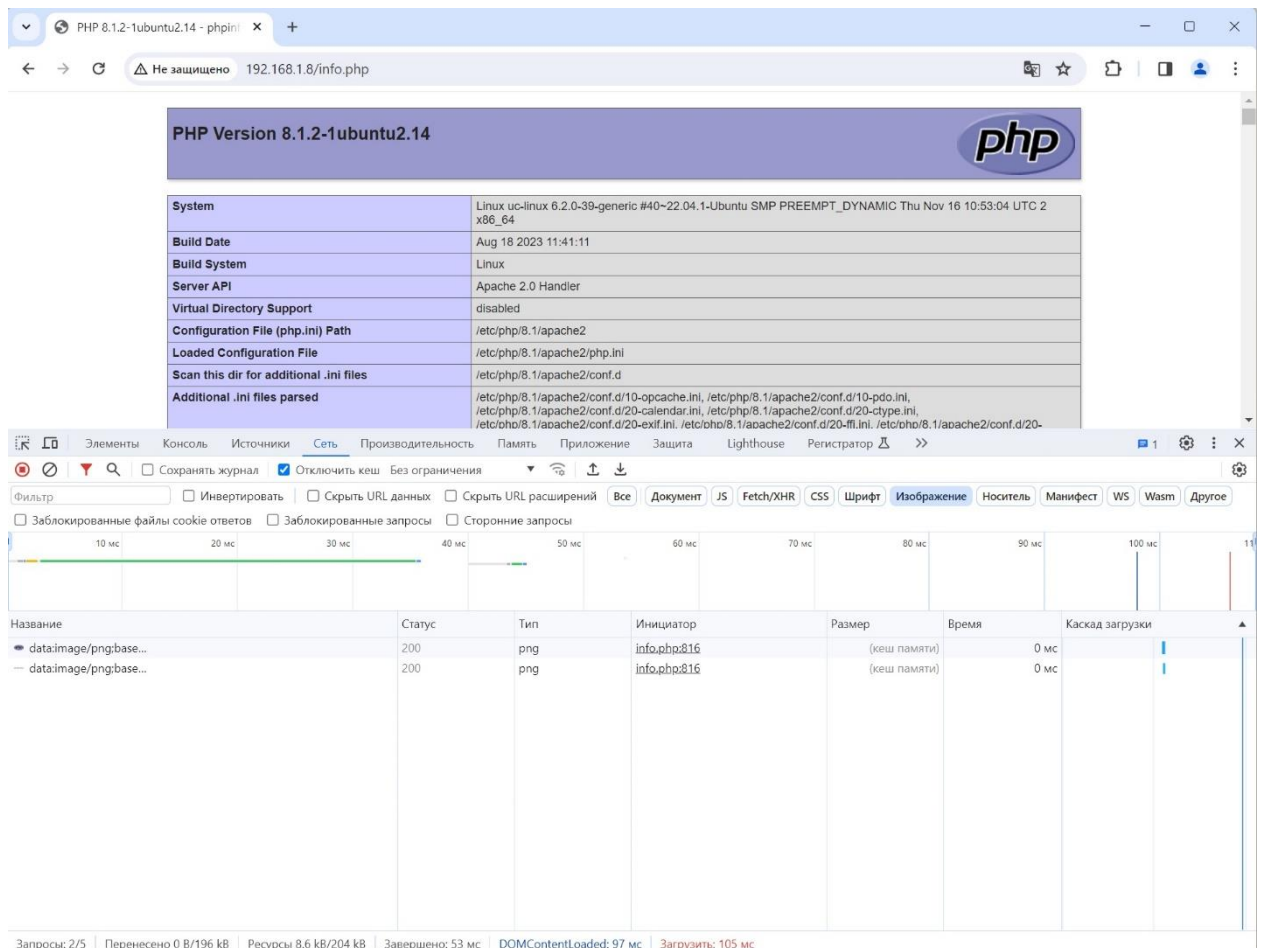


Рисунок 8. Обработка PHP файлов сервером Apache

6 Обработка PHP в PHP-FPM

6.1 Установка сервера PHP-FPM

```
sudo apt install php8.1-fpm
```

6.2 Проверяем список процессов

```
ps afx
```

6.3 Настроим конфигурацию Nginx

```
sudo nano /etc/nginx/sites-enabled/default
```

6.4 Отредактируем файл /etc/nginx/sites-enabled/default

Раскомментируем как показано на (рисунок 9) фрагмент, обозначенный зеленым прямоугольником и изменяем название файла (подчеркнутый зеленым) в соответствии со строчкой в файле /etc/php/8.1/fpm/pool.d/www.conf. Нас будет интересовать строчка:

```
listen = /run/php/php8.1-fpm.sock
```

где php8.1-fpm.sock – это unix сокет, название которого и впишем в файл конфигурации /etc/nginx/sites-enabled/default.


```
uc@uc-linux: /var/www/html X + v
GNU nano 6.2 /etc/nginx/sites-enabled/default *
# 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;
    # Динамические запросы
    proxy_pass http://localhost: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;

}

# pass PHP scripts to FastCGI server
#
location ~ /\.php$ {
    include snippets/fastcgi-php.conf;
    root /var/www/html;

    #
    # With php-fpm (or other unix sockets):
    fastcgi_pass unix:/run/php/php8.1-fpm.sock;
    # With php-cgi (or other tcp sockets):
    fastcgi_pass 127.0.0.1:9000;
}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
# Статические запросы
location ~* ^.(jpg|jpeg|gif|png|ico|css|zip|pdf|txt|tar|js)$ {
    root /var/www/html;
}

}

# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink that
# to sites-enabled/ to enable it.
#
#server {
#    listen 80;
#    listen [::]:80;
#
#    server_name example.com;
#
#    root /var/www/example.com;
#    index index.html;
#
#    location / {
#        try_files $uri $uri/ =404;
#    }
#}
```

Рисунок 9. Конфигурируем под PHP-FPM сервер. При выходе из редактора нажать Ctrl + O, Enter, Ctrl + X

6.5 Проводим тест конфигурации

```
sudo nginx -t
```

6.6 Применяем новую конфигурацию

```
sudo systemctl reload nginx
```

6.7 Проверяем отработку PHP в браузере

В строке адреса указываем IP нашего сервера и название PHP файла. Рисунок 10 показывает вывод в браузере нашего PHP файла. Server API – FPM/FastCGI.

The screenshot shows a web browser window with the address bar displaying "192.168.1.8/info.php". The main content area displays the PHP version and configuration details. Below the browser window, a network inspector is open, showing a list of requests. The first request is a GET request to "data:image/png;base64..." with a status of 200 and a type of png. The second request is also a GET request to "data:image/png;base64..." with a status of 200 and a type of png. The network inspector also shows a timeline of requests and a summary of the page load.

System	Linux uc-linux 6.2.0-39-generic #40~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Nov 16 10:53:04 UTC 2 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-sysmsg.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
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902.NTS

Название	Статус	Тип	Инициатор	Размер	Время	Каскад загрузки
data:image/png;base64...	200	png	info.php:751	(кеш памяти)	0 мс	
data:image/png;base64...	200	png	info.php:751	(кеш памяти)	0 мс	

Запросы: 2/5 | Перенесено 0 B/190 kB | Ресурсы 8.6 kB/198 kB | Завершено: 50 мс | DOMContentLoaded: 73 мс | Загрузить: 80 мс

Рисунок 10. Отработка PHP файлов на сервере PHP-FPM

7 СУБД MySQL

7.1 Установка сервера MySQL

```
sudo apt install mysql-server-8.0
```

7.2 Проверка запущенных задач

```
ps afx
```

7.3 Посмотрим сокеты

```
sudo ss -ntlp
```

7.4 Заходим в консоль MySQL

```
sudo mysql
```

Далее мы будем работать уже в консоли MySQL, а не Linux. Команды консоли MySQL выделю другим цветом.

7.5 Информация о базах данных

```
show databases;
```

7.6 Заходим в выбранную базу данных

```
use mysql;
```

7.7 Посмотрим какие таблицы там есть

```
show tables;
```

7.8 Выполним запрос SQL

Посмотрим записи в таблице user.

```
SELECT * FROM user\G;
```

7.9 Создадим новую базу данных

```
CREATE DATABASE uc;
```

7.10 Заходим в созданную базу данных

```
use uc;
```

7.11 Создадим таблицу (сущность) в базе данных

Создадим таблицу с единственным столбцом 'i' с типом данных INT.

```
CREATE TABLE example_table (i INT);
```

7.12 Добавим данные в созданную таблицу

```
INSERT INTO example_table (i) VALUES (1),(2),(3),(4);
```

7.13 Посмотрим добавленные данные

```
SELECT * FROM example_table;
```

7.14 Выходим из консоли MySQL, возвращаемся в консоль Linux

```
exit
```

8 Установка пакета phpMyAdmin

8.1 Сделаем, чтобы все запросы отправлялись на сервер Apache

Выполним пункты 4.5.1 - 4.5.3.

8.2 Обновим индекс пакетов

```
sudo apt update
```

8.3 Установка пакетов

Помимо пакета `phpmyadmin`, официальная документация рекомендует установить несколько расширений PHP для возможности использования определенной функциональности и улучшения производительности, а именно:

- `php-mbstring` - модуль для работы со строками, не поддерживающими кодировку ASCII;
- `php-zip` – расширение, поддерживающее загрузку файлов .zip;
- `php-gd` - поддержка библиотеки GD Graphics;
- `php-json` - поддержка сериализации JSON;
- `php-curl` – расширение, позволяющее PHP взаимодействовать с разными типами серверов, используя разные протоколы.

```
sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

При установке необходимо выбрать следующие параметры при запросах:

- При выборе сервера выбирайте `apache2`.
- На вопрос о необходимости использования `dbconfig-common` для настройки базы данных, следует выбрать «Да».
- Затем необходимо выбрать и подтвердить пароль приложения MySQL для `phpMyAdmin`.

8.4 Настройка доступа для специально созданного пользователя MySQL

8.4.1 Войти в MySQL

```
sudo mysql
```

8.4.2 Создать нового пользователя с именем 'uc' и паролем '1'

```
CREATE USER 'uc'@'localhost' IDENTIFIED WITH caching_sha2_password BY '1';
```

8.4.3 Предоставляем права доступа пользователю

```
GRANT ALL PRIVILEGES ON *.* TO 'uc'@'localhost' WITH GRANT OPTION;
```

8.4.4 Выходим из MySQL

```
exit
```

8.5 Подключаем файл конфигурации phpMyAdmin к серверу Apache

Откроем в редакторе nano файл конфигурации Apache /etc/apache2/apache2.conf и добавим в конец файла следующую строку:

```
# Подключаем файл конфигурации phpmyadmin  
IncludeOptional /etc/phpmyadmin/apache.conf
```

```
sudo nano /etc/apache2/apache2.conf
```

8.6 Проверяем корректность конфигурационных файлов

```
sudo apachectl -t
```

8.7 Применяем изменения конфигурации

```
sudo systemctl reload apache2
```

8.8 Зайдем в phpMyAdmin через WEB браузер

Набираем в адресной строке IP_сервера/phpMyAdmin/ см. рисунок 11, далее откроется интерфейс пользователя (рисунок 12).

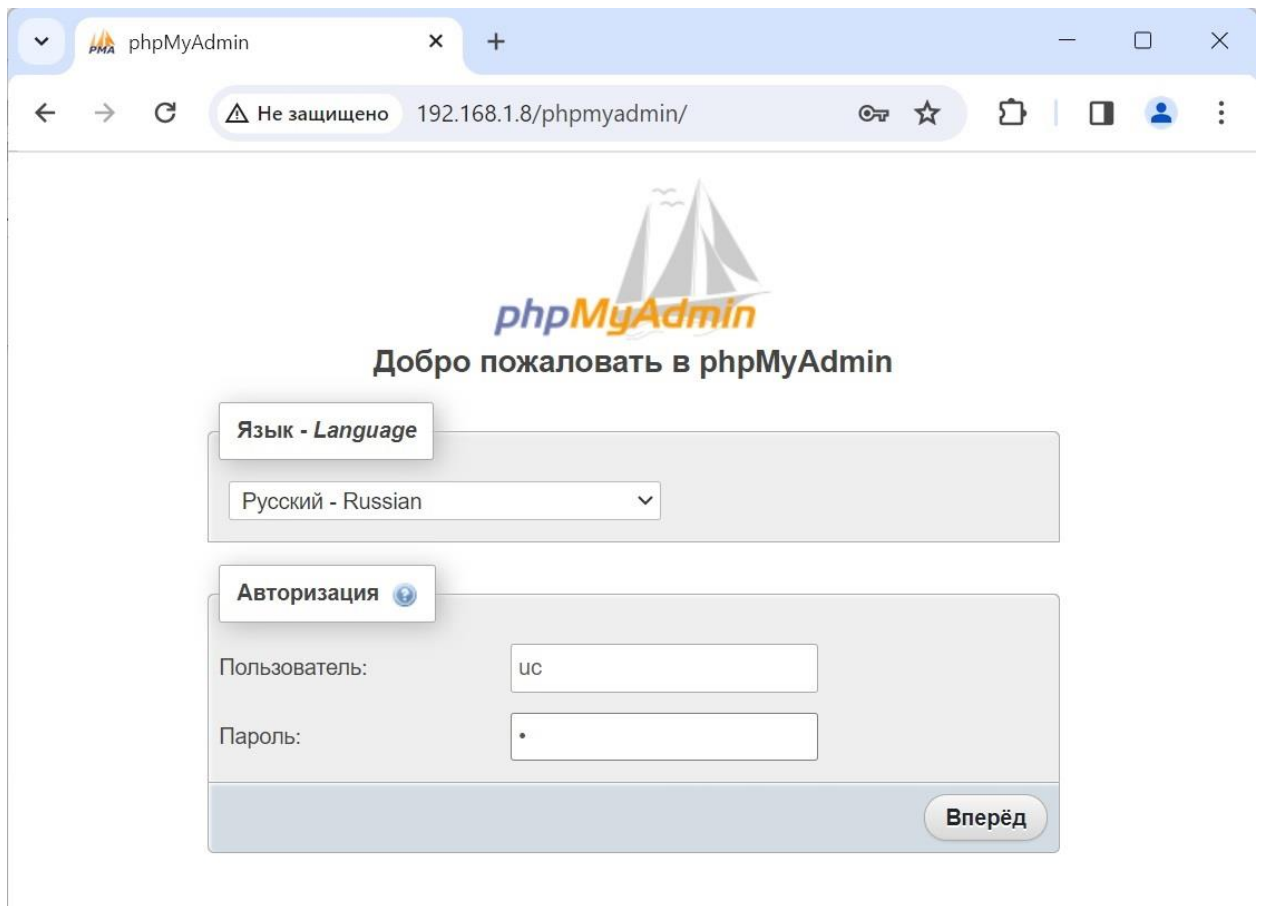


Рисунок 11. Подключение к phpMyAdmin

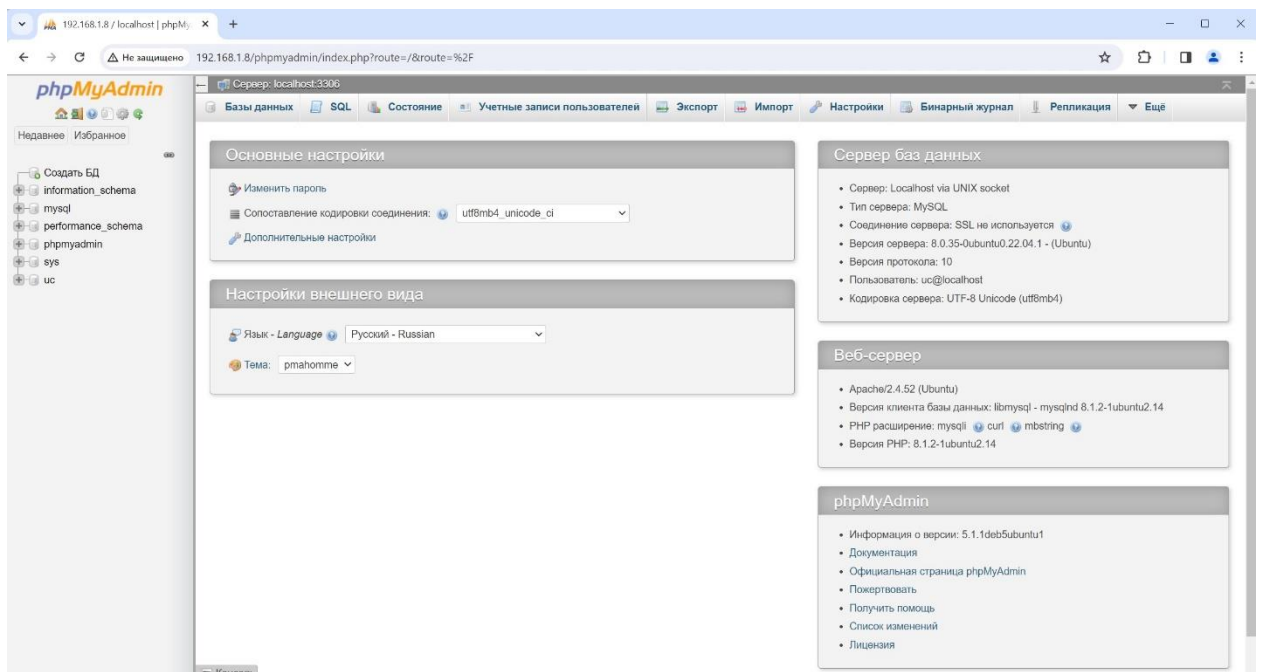


Рисунок 12. Интерфейс пользователя phpMyAdmin

9 Балансировка трафика между несколькими серверами Apache на стороне Nginx с помощью модуля ngx_http_upstream_module

9.1 Клонирование виртуальной машины Linux

Выполним клонирование текущей версии виртуальной машины Linux, в программе VirtualBox (рисунок 13). Выбрать текущую версию виртуалки (на которой выполнялись задания) и с помощью правой кнопки мыши сделаем несколько клонов виртуальной машины.

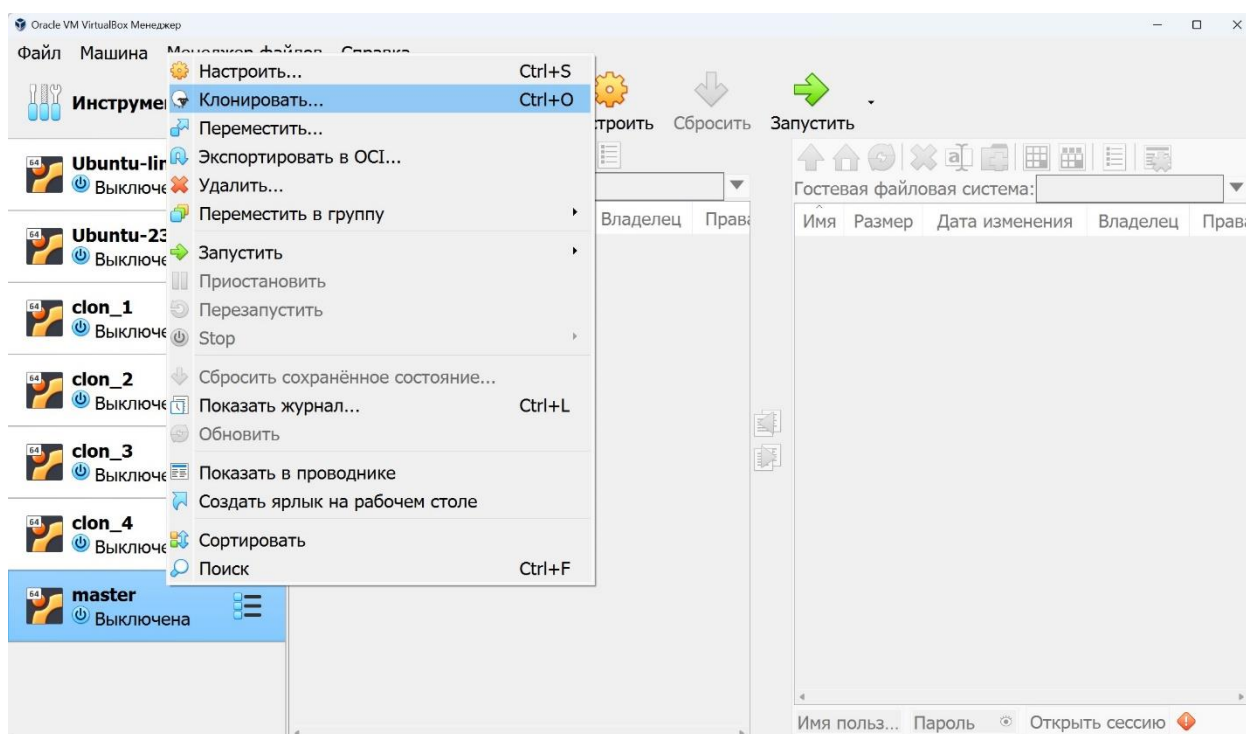


Рисунок 13. Клонирование виртуальной машины в программе VirtualBox

9.2 Изменение статического IP виртуальной машины

9.2.1 Коннектимся к виртуалке

Выполняем подключение через SSH соединение, как в пункте 1.1 по старому IP адресу.

9.2.2 Открываем через nano, файл сетевых настроек

```
sudo nano /etc/netplan/01-network-manager-all.yaml
```

9.2.3 Ставим другой статический IP

Рисунок 14 – статический IP обозначен зелёным прямоугольником.

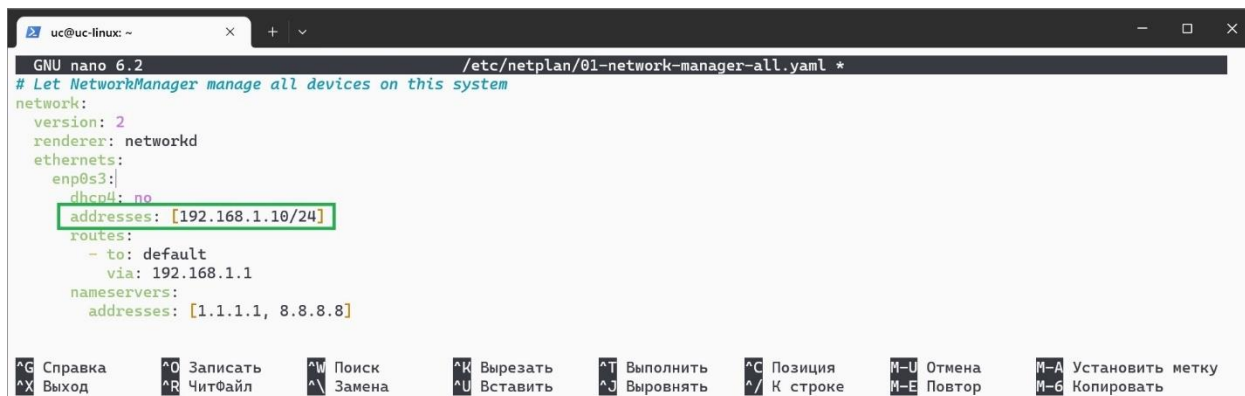


Рисунок 14. Изменение статического IP адреса. При выходе из редактора нажать Ctrl + O, Enter, Ctrl + X

9.2.4 Применение сетевых настроек

Запоминаем для себя IP-ники которые назначаем, потом будем прописывать их файле. Все IP-ники должны быть разные.

```
sudo netplan apply
```

9.2.5 Меняем сетевые настройки остальных виртуалок

Продолжаем пункты 9.2.1 - 9.2.4 на остальных виртуалках.

9.3 Изменяем конфигурация сервера Nginix

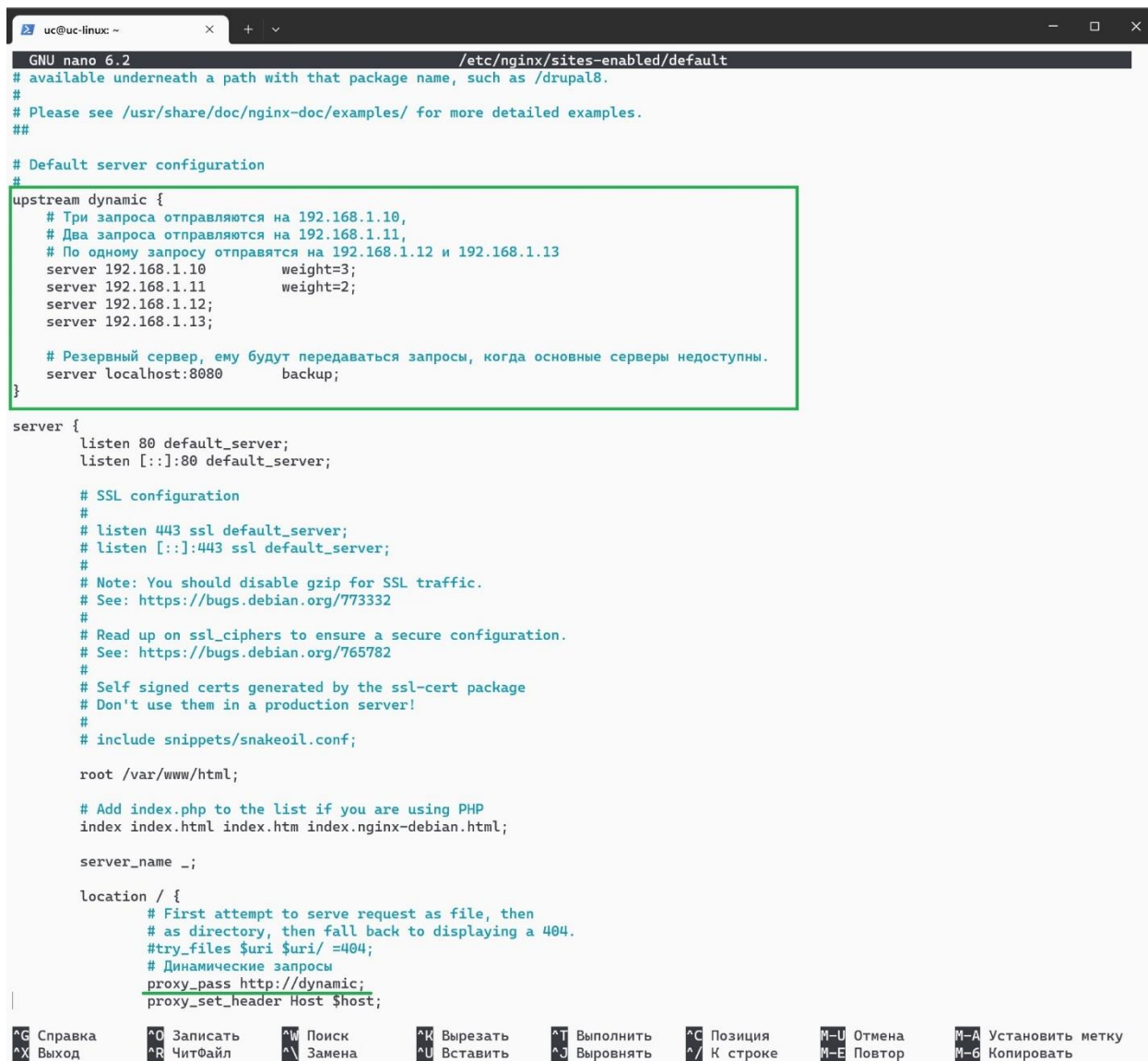
```
sudo nano /etc/nginx/sites-enabled/default
```

Допишем в файле виртуальной машины (ВМ) 'master' следующий текст:

```
upstream dynamic {
    # Три запроса отправляются на 192.168.1.10,
    # Два запроса отправляются на 192.168.1.11,
    # По одному запросу отправятся на 192.168.1.12 и 192.168.1.13
    server 192.168.1.10    weight=3;
    server 192.168.1.11    weight=2;
    server 192.168.1.12;
    server 192.168.1.13;

    # Резервный сервер, ему будут передаваться запросы, когда основные серверы
    # недоступны.
    server localhost:8080    backup;
}
```

Добавленный текст выделен зелёным прямоугольником, а также нужно отредактировать строку, подчеркнутую зелёной линией (рисунок 15).



```
GNU nano 6.2 /etc/nginx/sites-enabled/default
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
upstream dynamic {
    # Три запроса отправляются на 192.168.1.10,
    # Два запроса отправляются на 192.168.1.11,
    # По одному запросу отправляются на 192.168.1.12 и 192.168.1.13
    server 192.168.1.10 weight=3;
    server 192.168.1.11 weight=2;
    server 192.168.1.12;
    server 192.168.1.13;

    # Резервный сервер, ему будут передаваться запросы, когда основные серверы недоступны.
    server localhost:8080 backup;
}

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;
        # Динамические запросы
        proxy_pass http://dynamic;
        proxy_set_header Host $host;
    }
}
```

Рисунок 15. Балансировка запросов на сервера

9.4 Проверим корректность конфигурации

```
sudo nginx -t
```

9.5 Применим новую конфигурацию

```
sudo systemctl reload nginx
```

Теперь при запросе на IP 192.168.1.8 (у меня он такой) трафик будет перенаправляться в соответствии со схемой которую мы описали (рисунок 15).

9.6 Проверим как балансируется трафик

Перезапишем наш файл info.php, чтобы выводилось имя VM. Нужно на каждой VM перезаписать файл info.php.

Для VM 'master', (рисунок 13):

```
sudo su  
  
echo "<?php echo 'master'; ?>" > /var/www/html/info.php
```

Для ВМ 'clon_1', (рисунок 13):

```
sudo su  
  
echo "<?php echo 'clon_1'; ?>" > /var/www/html/info.php
```

Для ВМ 'clon_2', (рисунок 13):

```
sudo su  
  
echo "<?php echo 'clon_2'; ?>" > /var/www/html/info.php
```

Для ВМ 'clon_3', (рисунок 13):

```
sudo su  
  
echo "<?php echo 'clon_3'; ?>" > /var/www/html/info.php
```

Для ВМ 'clon_4', (рисунок 13):

```
sudo su  
  
echo "<?php echo 'clon_4'; ?>" > /var/www/html/info.php
```

Тогда если в браузере зайти на страницу info.php (например <http://192.168.1.8/info.php>), тогда будут отображаться разные названия ВМ, в зависимости от схемы балансировки трафика (рисунок 15).

Вывод

Выполнив данное ДЗ, я научился устанавливать и настраивать сервера Nginx, Apache, PHP-FPM, MySQL, настраивать работу серверов в режиме конфигурации Reverse proxy, а также создавать базы данных, таблицы и добавлять новые данные с помощью командной строки на сервере MySQL. Научился устанавливать и настраивать пакет phpMyAdmin для работы с MySQL через WEB интерфейс, а также научился балансировать трафик между группой серверов.

Данное ДЗ выполнял на Ubuntu 20.04, по сколько некоторые пакеты на Ubuntu 23.10 не ставились, сервер Apache устанавливается без картинки логотипа. С Ubuntu 20.04 никаких проблем не было.

Литература

- Evgeny, E. (23 12 2023 г.). *ДЗ семинар 5*. Получено из https://glonassgps-my.sharepoint.com/:b:/g/personal/uc20100_glonassgps_onmicrosoft_com/EdofcMN3lktLhH_3NYSedncByV70EyPLpPEGtv90DwbInQ?e=ALLSsV
- GeekBrains. (23 12 2023 г.). *Веб-сервер на Linux (Лекция)*. Получено из <https://gbcdn.mrgcdn.ru/uploads/record/246040/attachment/54759d94f2f64e9c46bf44792c77ae86.mp4>
- GeekBrains. (23 12 2023 г.). *Веб-сервер на Linux (Презентация)*. Получено из <https://gbcdn.mrgcdn.ru/uploads/asset/4327196/attachment/4fa430da04b3138a2e23e8bb03494df3.pdf>
- GeekBrains. (23 12 2023 г.). *Веб-сервер на Linux (Презентация)*. Получено из <https://gbcdn.mrgcdn.ru/uploads/asset/4327200/attachment/5f1dfd3a0177ca7b0fa5149eb80b9b16.pdf>
- GeekBrains. (17 12 2023 г.). *Сетевые возможности Linux (Лекция)*. Получено из <https://gbcdn.mrgcdn.ru/uploads/record/246039/attachment/0bbae3dc0266d4ab0a39dc0504954409.mp4>
- GeekBrains. (17 12 2023 г.). *Сетевые возможности Linux (Презентация)*. Получено из <https://gbcdn.mrgcdn.ru/uploads/asset/4327093/attachment/2264165087c249d1b7b58353d6e84f5e.pdf>
- GeekBrains. (17 12 2023 г.). *Сетевые возможности Linux (Конспект)*. Получено из <https://gbcdn.mrgcdn.ru/uploads/asset/4327097/attachment/0c7e44f4cc0f4f2073d6dd20338f46a2.pdf>
- <https://engine.taobao.org/>. (23 12 2023 г.). *Модуль ngx_http_upstream_module*. Получено из https://engine.taobao.org/nginx_docs/ru/docs/http/nginx_http_upstream_module.html
- RU VDS. (22 12 2023 г.). *Как установить phpMyAdmin на Ubuntu 20.04*. Получено из <https://ruvds.com/ru/helpcenter/kak-ustanovit-phpmyadmin/>
- Runebook.dev. (23 12 2023 г.). *Module ngx_http_upstream_module*. Получено из https://runebook.dev/ru/docs/nginx/http/nginx_http_upstream_module?