

## Развёртывание связки PHP-FPM + NGINX на виртуальном хостинге и в докер-контейнере.

### Задания

Перейдите по адресу <https://kodaktor.ru/cgi-bin/script.cgi> и сопоставьте URL с тем, что отображается на веб-странице. На что указывают последние четыре символа в URL? Есть ли у вас какие-нибудь способы узнать, адрес чего находится в адресной строке?

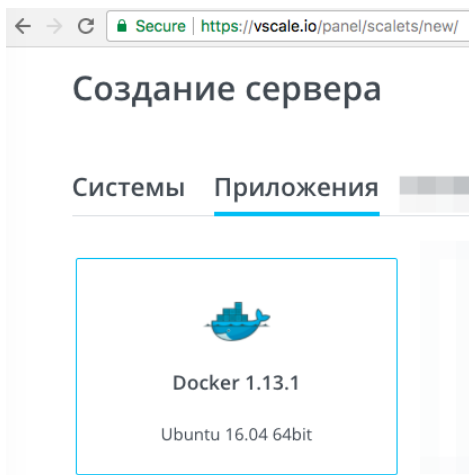
Проанализируйте код программы по адресу [https://github.com/GossJS/php\\_starters1/blob/cgi/script.c](https://github.com/GossJS/php_starters1/blob/cgi/script.c).  
Найдите в нём инструкции, ответственные за вывод в стандартный поток.

В качестве дополнительного чтения ознакомьтесь с документом по адресу <https://kodaktor.ru/g/cgi>

I. Следуя инструкциям, осуществите развёртывание PHP/FPM + NGINX.  
Для этого:

1. Зарегистрируйтесь на сайте [vscale.io](https://vscale.io) — в результате у вас должен появиться собственный аккаунт.

2. Перейдите по адресу <https://vscale.io/panel/scalets/new/> для создания нового виртуального сервера. Мы установим докер поверх Linux-дистрибутива Ubuntu 16.04, затем создадим докер-контейнер с Linux-дистрибутивом Alpine и уже в нём развернём PHP-FPM.



Нам нужно также указать SSH-ключи для веб-сервера (эта тема выходит за рамки данного курса).

После успешного создания сервера мы получим самую ценную информацию о нём: его IP адрес.

## Серверы



3. Подключитесь к серверу по протоколу SSH:

```
$ ssh root@212.92.98.139
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-109-generic x86_64)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage
```

The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/\*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.

```
root@cs196843:~# █
```

4. Выполните команду докера для создания нового контейнера, в котором будет находиться дистрибутив Linux Alpine.

```
docker run --name php1 -p 4321:1234 -itd alpine:latest tail -f /dev/null
root@cs196843:~# docker run --name php1 -p 4321:1234 -itd alpine:latest tail -f /dev/null
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
ff3a5c916c92: Pull complete
Digest: sha256:7df6db5aa61ae9480f52f0b3a06a140ab98d427f86d8d5de0bedab9b8df6b1c0
Status: Downloaded newer image for alpine:latest
ec0f9a28b64d225435c035858e6a9f35fd686ca6efd62a53021bb9af47928d6a
root@cs196843:~# █
```

5. Войдите внутрь контейнера

```
docker exec -it php1 sh
```

6. Выполните команду менеджера зависимостей apk

```
apk add --update nginx php7-fpm nano mc
```

для установки веб-сервера NGINX, менеджера процессов PHP-FPM, текстового редактора nano и панельного менеджера mc.

```

root@cs196843:~# docker exec -it php1 sh
/ # apk add --update nginx php7-fpm nano mc
fetch http://dl-cdn.alpinelinux.org/alpine/v3.7/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.7/community/x86_64/APKINDEX.tar.gz
(1/19) Installing libffi (3.2.1-r4)
(2/19) Installing libintl (0.19.8.1-r1)
(3/19) Installing libuuid (2.31-r0)
(4/19) Installing libblkid (2.31-r0)
(5/19) Installing libmount (2.31-r0)
(6/19) Installing pcre (8.41-r1)
(7/19) Installing glib (2.54.2-r0)
(8/19) Installing ncurses-terminfo-base (6.0_p20170930-r0)
(9/19) Installing ncurses-terminfo (6.0_p20170930-r0)
(10/19) Installing ncurses-libs (6.0_p20170930-r0)
(11/19) Installing libssh2 (1.8.0-r2)
(12/19) Installing mc (4.8.20-r0)
(13/19) Installing libmagic (5.32-r0)
(14/19) Installing nano (2.9.1-r0)
(15/19) Installing nginx (1.12.2-r3)
Executing nginx-1.12.2-r3.pre-install
(16/19) Installing php7-common (7.1.12-r0)
(17/19) Installing libedit (20170329.3.1-r3)
(18/19) Installing libxml2 (2.9.7-r0)
(19/19) Installing php7-fpm (7.1.12-r0)
Executing busybox-1.27.2-r7.trigger
Executing glib-2.54.2-r0.trigger
OK: 30 MiB in 30 packages
/ # █

```

## 7. Выполните

```
nano /etc/php7/php-fpm.conf
```

для редактирования файла настройки PHP-FPM.

## 8. Добавьте строки в конце этого файла:

```
listen=/run/php/php7-fpm.sock
listen.owner = nginx
```

мы таким образом указываем,  
от чьего имени php-fpm7 создаст сокет для обмена информацией  
и сохраните изменения.

## 9. Измените конфигурационный файл сервера NGINX:

```
nano /etc/nginx/conf.d/default.conf
```

[https://github.com/GossJS/php\\_starters1/blob/nginx/default.conf](https://github.com/GossJS/php_starters1/blob/nginx/default.conf)

```
server
{
    listen 1234;
    listen [::]:1234;
    root /data/www;

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

В этом файле мы указали веб-серверу NGINX, что он должен обслуживать каталог /data/www и в случае если там окажутся файлы с расширением .php, передавать их на промежуточное обслуживание серверу PHP-FPM

10. Создайте файл 1.php в каталоге /data/www

```
mkdir /data && mkdir /data/www && touch /data/www/1.php
```

и поместите в него какие-нибудь простейшие команды PHP типа вывода текущей даты в заголовок h1:

```
GNU nano 2.9.1 /data/www/1.php

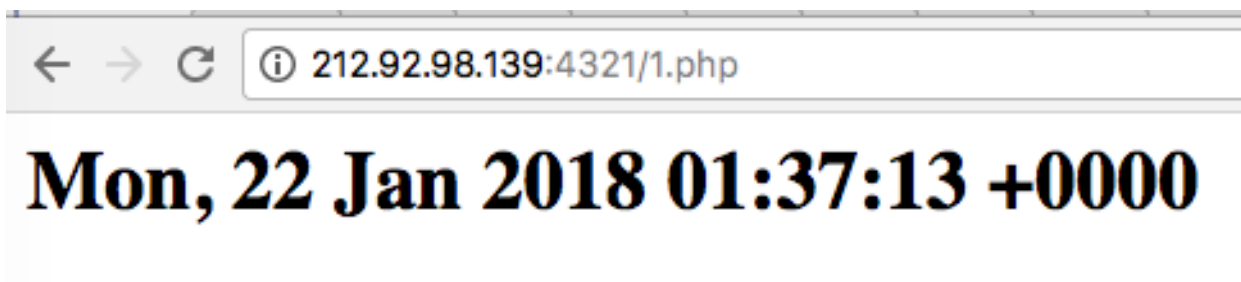
<?php
    echo '<h1>' . date('r') . '</h1>';
```

11. Создайте нужные для работы папки и запустите оба сервера: PHP-FPM и NGINX:

```
mkdir /run/php && php-fpm7
mkdir /run/nginx && nginx
```

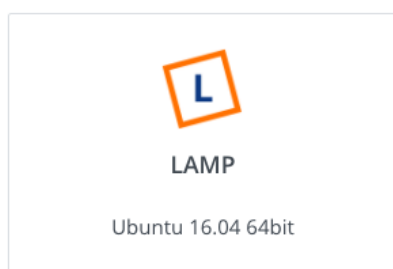
12. И перейдите по адресу <http://212.92.98.139:4321/1.php>

чтобы убедиться, что сценарий работает.



#### Задания для самостоятельного выполнения:

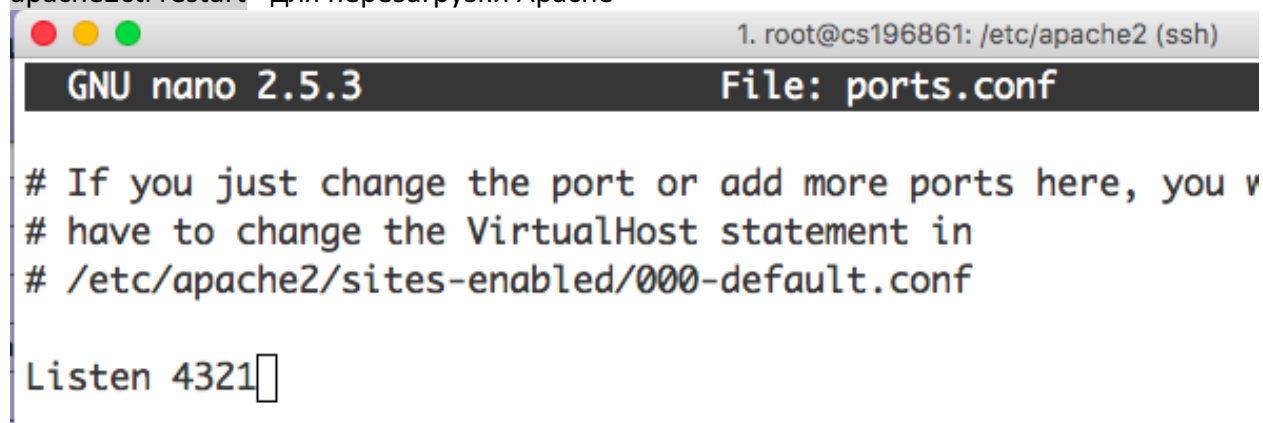
1. Создайте сценарий `i.php` в той же папке, содержащий команду `phpinfo()`;
2. Передайте адрес сервера и полный путь к этому сценарию преподавателю, убедившись что он работает и выводит информацию о сервере: `http://212.92.98.139:4321/i.php`
3. Создайте в той же папке сценарий `author.php`, выводящий в заголовках `h1` и `h2` соответственно ваши фамилию и имя.
4. Создайте ещё один сервер с приложением LAMP



и также создайте файл `i.php` в папке `/var/www/html` с командой `phpinfo()`;

Пользуйсь справочными материалами в Интернете, настройте сервер Apache так, чтобы он слушал не порт 80, как это настроено по умолчанию, а порт 4321. Передайте адрес сервера и полный путь к сценарию `i.php` преподавателю для проверки.

`apache2ctl restart` - для перезагрузки Apache



```
1. root@cs196861: /etc/apache2 (ssh)
GNU nano 2.5.3 File: sites-enabled/000-default.conf
<VirtualHost *:4321>
# The ServerName directive sets the request scheme, host
```

← → ↻ ⓘ 46.21.248.14:4321/i.php ☆ ⓘ ⚙ 🌐 ⌚ ↺ 📄 ▼

**PHP Version 7.0.22-0ubuntu0.16.04.1**

System	Linux cs196861 4.4.0-109-generic #132-Ubuntu SMP Tue Jan 9 19:52:39 UTC 2018 x86_64
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-mysqlnd.ini; /etc/php/7.0/apache2/conf.d/10-opcache.ini

Поскольку серверы платные, выполнение этих заданий должно быть достаточно динамичным.