

Администрирование Nginx/Angie

Обратный прокси (reverse proxy)



Проверить, идет ли запись

Меня хорошо видно & слышно?



Преподаватель



Лавлинский Николай

Технический директор «Метод Лаб»

Более 15 лет в веб-разработке

Преподавал в ВУЗе более 10 лет
Более 5 лет в онлайн-образовании

Специализация: оптимизация производительности, ускорение сайтов и веб-приложений

https://t.me/methodlab_tg

<https://www.methodlab.ru/>

<https://www.youtube.com/c/NickLavlinsky>

https://www.youtube.com/@site_support

<https://vk.com/nick.lavlinsky>



Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в учебной группе



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом

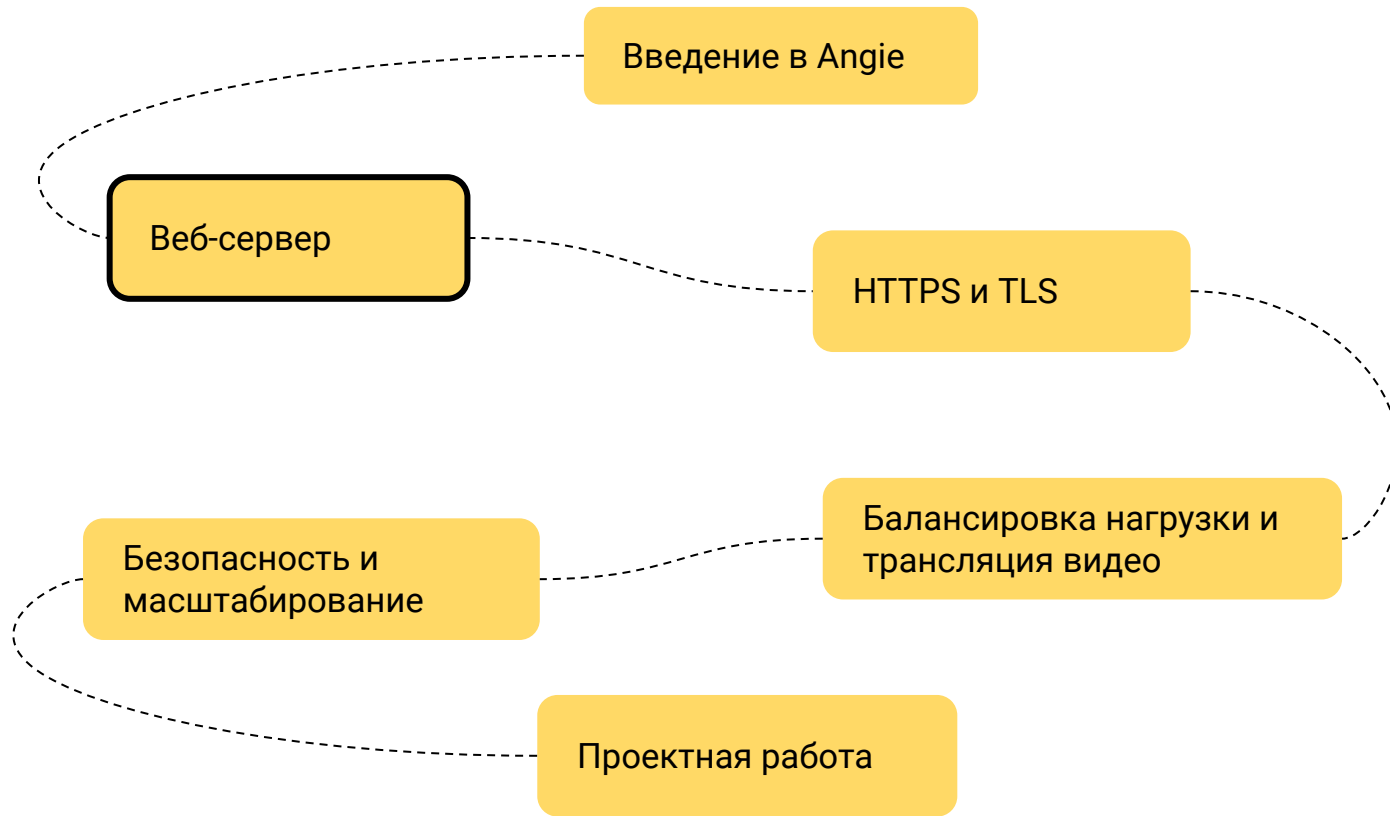


Документ



Ответьте себе или
задайте вопрос

Карта курса



Маршрут вебинара



Архитектура обратного прокси

Разделение динамики и статики

Запуск приложения WordPress

Цели вебинара

К концу занятия вы сможете

1. Понимать принцип работы обратного прокси
2. Научиться настраивать разделение статики и динамики
3. Запустить веб-приложение с использованием схемы обратного прокси

Смысл

Зачем вам это уметь

1. Настраивать стандартную схему frontend-backend
2. Запускать реальные веб-приложения
3. Настраивать разделение статики и динамики

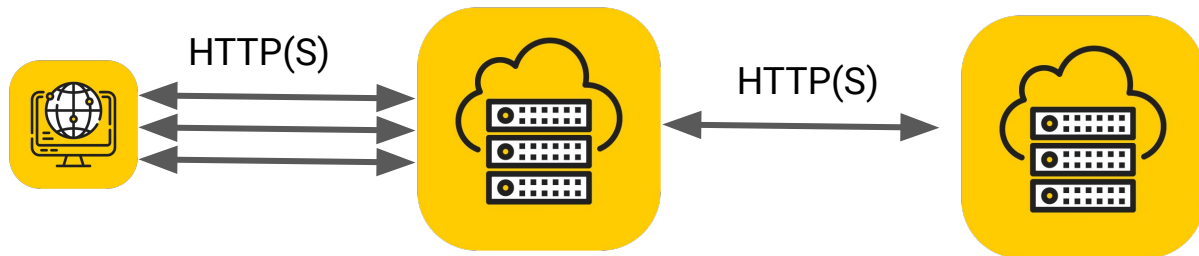


Обратный прокси frontend и backend-сервер

Front-end и back-end

Front: IP 1.2.3.4
Порты: 80 , 443

Back: IP 127.0.0.1
Порты: 8080



- TLS
- Отдача статики
- Проксирование на back-end

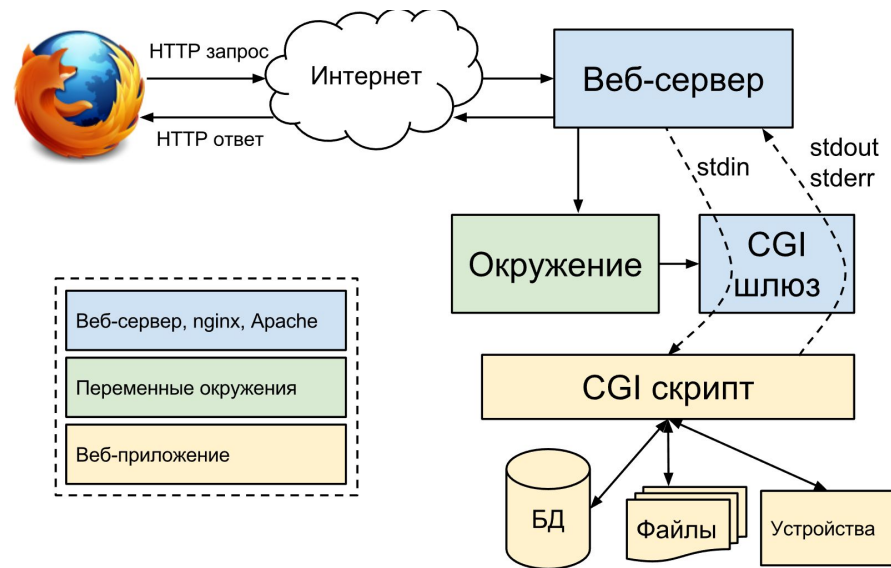
- Динамика
- mod_php, mod_perl, mod_fcgi, mod_uwsgi

Варианты проксирования

- **proxy_pass** - HTTP proxy
- **fastcgi_pass** - FastCGI proxy (php)
- **uwsgi_pass** - UWSGI proxy (python)
- **scgi_pass** - SCGI proxy
- **memcached_pass** - memcached proxy (memcached)
- **grpc_pass** - GRPC proxy (grpc)

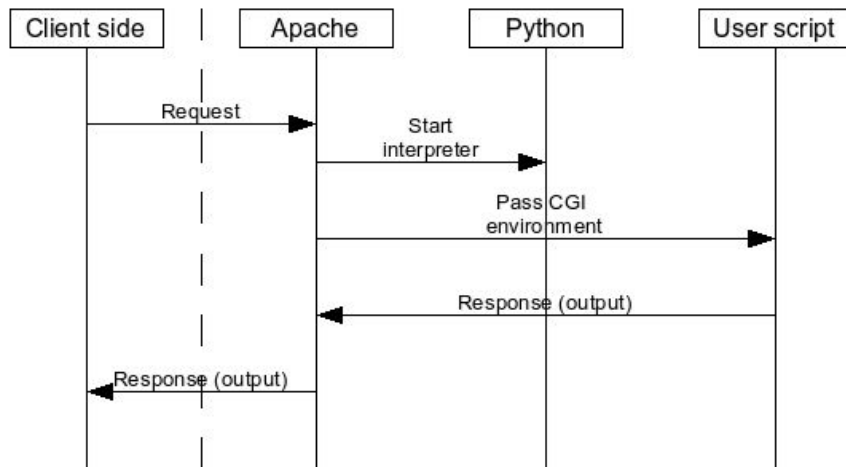
CGI

- CGI (Common Gateway Interface) - стандарт интерфейса, используемого для связи внешней программы с веб-сервером
- Скрипт запускается при каждом запросе
- CGI-параметры передаются в виде переменных окружения
- Скрипт возвращает результат в STDOUT, который транслируется сервером клиенту
- Высокие накладные расходы на интерпретацию кода (каждый запрос)



CGI – процесс обработки запроса

CGI



CGI — конфигурация (Apache)

```
LoadModule cgi_module /usr/lib/apache2/modules/mod_cgi.so
```

```
AddHandler cgi-script .cgi .pl
```

```
ScriptAlias /cgi-bin/ /home/www/cgi-bin/
```

<https://httpd.apache.org/docs/2.4/howto/cgi.html>

FastCGI

- Протокол взаимодействия между веб-сервером и сервером приложений
- Блокирующий доступ
- Используется TCP или UNIX socket
- Пример конфигурации в Nginx:

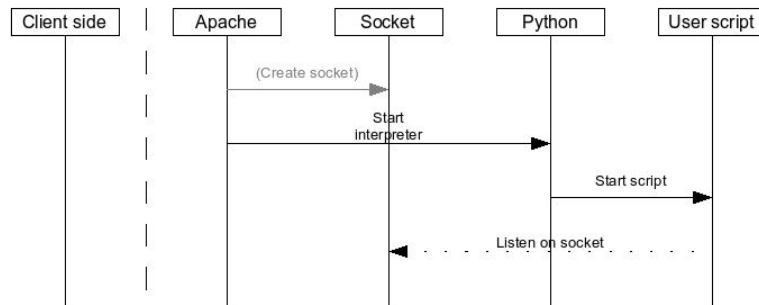
```
location ~ ^/(status|ping)$ {  
    allow 127.0.0.1;  
    deny all;  
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;  
    fastcgi_index index.php;  
    include fastcgi_params;  
    fastcgi_pass unix:/var/run/phpfpm-api.sock;  
}
```

FastCGI — принципы работы

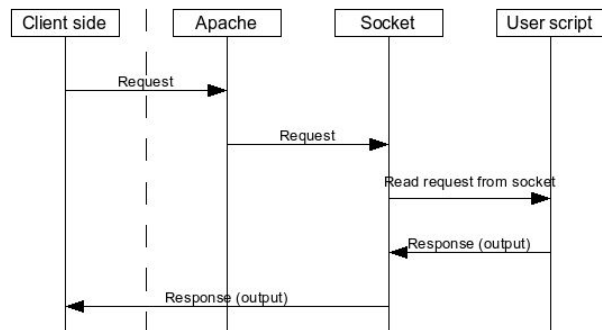
- Программа загружается в память в качестве при первом обращении к серверу
- Не требуется процесс интерпретации и подготовки кода к исполнению
- Один и тот же процесс обрабатывает множество запросов
- Высокая производительность
- Больше потребление оперативной памяти

FastCGI – процесс обработки запроса

FastCGI (Startup)

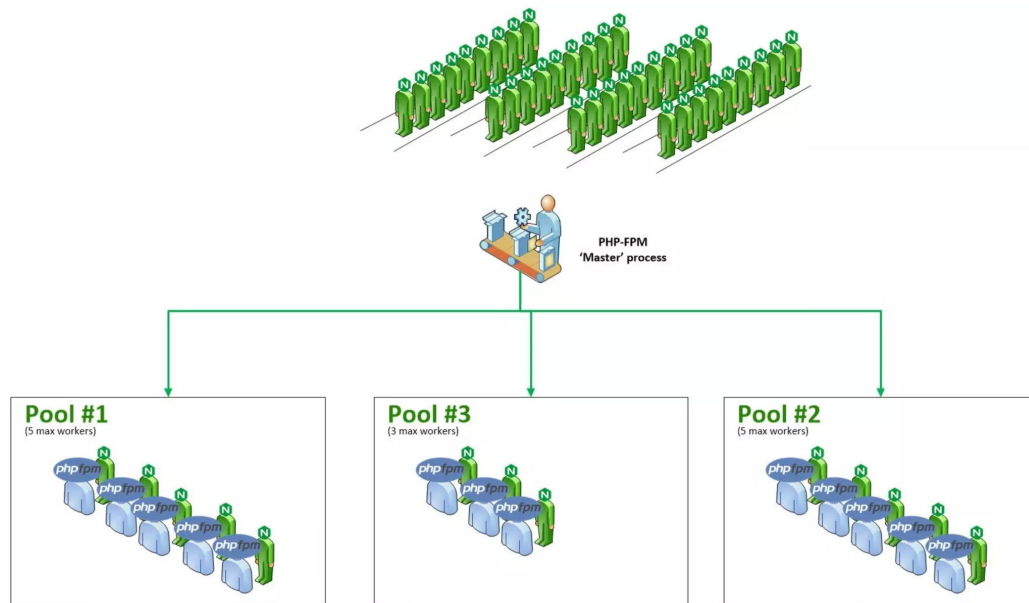


FastCGI (Request handling)



PHP-FPM

- Популярная реализация FastCGI-сервера для PHP
- Работает с большинством веб-приложений
- Позволяет настраивать пулы для приложений
- Статическое и динамическое количество воркеров



Настройка проксирования HTTP

```
location / {  
    proxy_pass http://localhost:8000/uri/;  
    proxy_pass http://unix:/tmp/backend.socket:/uri/;  
    proxy_pass http://upstream;  
    proxy_set_header Host $host;  
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
    proxy_set_header X-Real-IP $remote_addr;  
}
```

Параметры проксирования

- `proxy_buffers` — размеры буферов в памяти для чтения запроса (при превышении пишет на диск)
- `proxy_connect_timeout` — таймаут соединения с проксируемым сервером (ошибка 504)
- `proxy_read_timeout` — таймаут чтения ответа от сервера (между двумя чтениями)
- `proxy_send_timeout` — таймаут передачи запроса серверу (между двумя передачами)
- `proxy_set_header` — установка заголовка запроса к серверу

https://angie.software/configuration/modules/http/http_proxy/

Использование keepalive-соединений

```
location / {  
    proxy_http_version 1.1;  
    proxy_set_header Connection "";  
    proxy_pass http://backend;  
}
```

https://angie.software/configuration/modules/http/http_proxy/

Проксирование на HTTPS

```
location / {  
    proxy_pass https://localhost:8000;  
    proxy_http_version 1.1;  
    proxy_set_header Connection "";  
    proxy_ssl_name $proxy_host;  
    proxy_ssl_server_name on;  
    proxy_ssl_verify off;  
    proxy_ssl_session_reuse on;  
    proxy_ssl_protocols TLSv1.2 TLSv1.3;  
    proxy_ssl_ciphers HIGH:!aNULL:!MD5;  
}
```

https://angie.software/configuration/modules/http/http_proxy/#proxy-ssl-protocols



HTTPS с клиентским сертификатом

```
location / {  
    proxy_pass https://localhost:8000;  
    proxy_ssl_certificate      /etc/angie/client.pem;  
    proxy_ssl_certificate_key  /etc/angie/client.key;  
    proxy_ssl_trusted_certificate /etc/angie/trusted_ca_cert.crt;  
    proxy_ssl_session_reuse on;  
    proxy_ssl_protocols       TLSv1.2 TLSv1.3;  
}  
...  
server {  
    listen      8000 ssl;  
    server_name backend1.example.com;  
    ssl_certificate      /etc/ssl/certs/server.crt;  
    ssl_certificate_key  /etc/ssl/certs/server.key;  
    ssl_client_certificate /etc/ssl/certs/ca.crt;  
    ssl_verify_client    optional;  
    ...  
}
```

https://angie.software/configuration/modules/http/http_proxy/#proxy-ssl-protocols

Разделение статики и динамики

Разделение статики и динамики

- Статический запрос — отдача файла с диска (CSS, JS, PNG, JPEG)
- Динамический запрос — исполнение серверного кода (PHP, Python, NodeJS, Perl...)
- Варианты разделения (location)
 - По доменному имени: `server_name static.site.ru;`
 - По префиксу URL: `/images {}`
 - По регулярному выражению: `location ~* \.(css|gif|ico|jpeg|jpg|js|png)$ {}`
 - С использованием `try_files`: `try_files $uri $uri/ /index.php?$args;`

Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет

Запуск CMS WordPress

WordPress компоненты

- Angie — веб-сервер, прокси
 - Обработка клиентских соединений
 - Обработка статических запросов
 - Проксирование динамических запросов
- PHP-FPM — FastCGI сервер
 - Исполнение серверного (PHP) кода
- MySQL, MariaDB — СУБД, база данных
 - Хранение данных (контент, пользователи)

Конфигурация Angie

- Разделение статики и динамики
- Обработка статики с помощью префиксных и regex location, а также try_files
- Отправка PHP на PHP-FPM
- Запрет доступа к кэшу и php-файлам извне

LIVE

Практика

Домашнее задание

1. Развернуть на вашей учебной виртуальной машине копию CMS Wordpress или аналогичное веб-приложение с использованием стандартного дистрибутива или Docker-контейнера.
2. Настроить Angie в качестве обратного прокси для бэкенда.
3. Разделить обработку статических и динамических запросов.

В чат ДЗ отправьте ссылку на ваш git-репозиторий



Сроки выполнения: указаны в личном кабинете

Рефлексия

Цели вебинара

Проверка достижения целей

1. Понимать принцип работы обратного прокси
2. Научиться настраивать разделение статики и динамики
3. Запустить веб-приложение с использованием схемы обратного прокси

Рефлексия



С какими впечатлениями уходите с вебинара?



Как будете применять на практике то, что узнали на вебинаре?

Следующий вебинар



Логирование



Ссылка на вебинар
будет в ЛК за 15 минут



Материалы
к занятию в ЛК —
можно изучать



Обязательный материал
обозначен красной
лентой

**Заполните, пожалуйста,
опрос о занятии
по ссылке в чате**

Спасибо за внимание!

Приходите на следующие вебинары



Лавлинский Николай

Технический директор “Метод Лаб”

https://t.me/methodlab_tg

<https://www.methodlab.ru/>

<https://www.youtube.com/c/NickLavlinsky>

https://www.youtube.com/@site_support

<https://vk.com/nick.lavlinsky>