



# Администратор Linux

## Динамический веб



Проверить, идет ли запись

# Меня хорошо видно && слышно?



Ставим "+", если все хорошо  
"-", если есть проблемы

Тема вебинара

# Динамический веб



**Федоров Иван Романович**

Технический директор ГК "Инотех"

**Опыт:**

Более 10 лет в IT-сфере

Аспирант университета ИТМО по направлению "Информационная безопасность"

Многократный победитель различных конкурсов и хакатонов (команда IBI Solutions)

Эл. почта: [ifedorov.devops@gmail.com](mailto:ifedorov.devops@gmail.com)



# Правила вебинара



Активно  
участвуем



Off-topic обсуждаем  
в группе Telegram



Задаем вопрос  
в чат или голосом



Вопросы вижу в чате,  
могу ответить не сразу



# Маршрут вебинара

Знакомство

CGI/FCGI/WSGI

php-fpm/uWSGI

AJAX/WebSockets

Практика

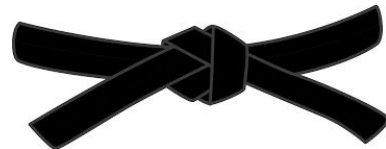
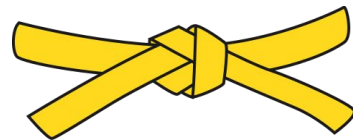
Рефлексия



# Цели вебинара

К концу занятия вы сможете

1. Понять принцип работы CGI/FastCGI и websocket
2. Ориентироваться в конфигурации php-fpm и uWSGI
3. Настраивать связки nginx + php-fpm, apache + php-fpm



# Смысл

## Зачем вам это уметь

1. Знания принципов работы “динамического веба” упрощает процесс траблшутинга при проблемах с сайтами (тормоза/не загружается контент/итд)
2. Вы сможете предусмотреть ряд проблем заранее при проектировании и начальной настройке инфраструктуры
3. Веб-сервисы — одна из самых часто встречающихся категорий сервисов в реальной жизни

**CGI**  
**FCGI**  
**WSGI**



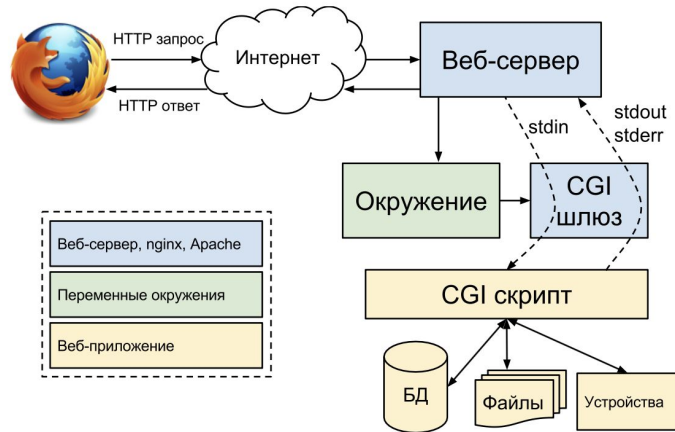


# CGI

**CGI (Common Gateway Interface)** — стандарт интерфейса, используемого для связи внешней программы с веб-сервером.

Веб-сервер, получая запрос:

- вызывает внешний скрипт и передает в него
  - параметры
  - переменные окружения
- возвращает его вывод в качестве ответа

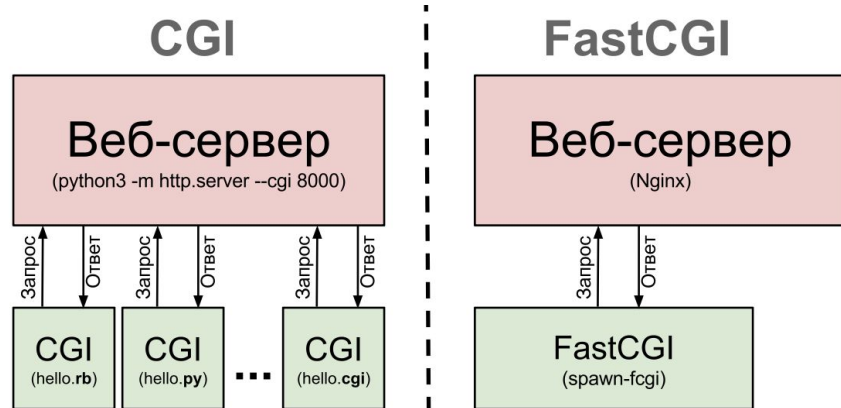


# FastCGI

**FastCGI** — протокол взаимодействия между веб-сервером и сервером FastCGI, дальнейшее развитие технологии CGI.

## Особенности:

- блокирующий
- взаимодействует через TCP или UNIX-socket

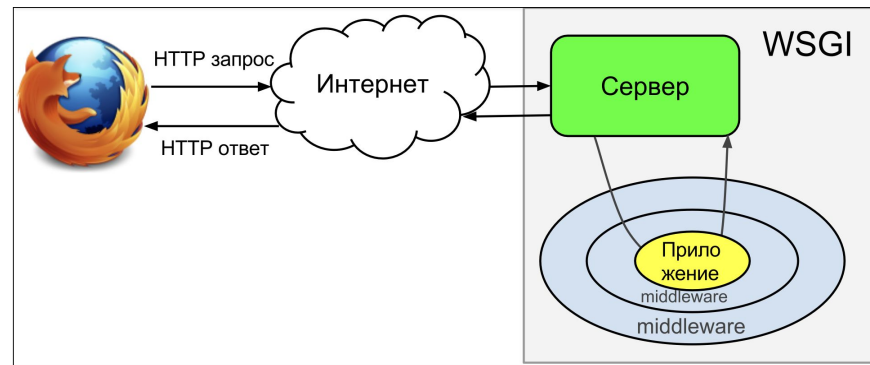


# WSGI

**WSGI** — стандарт взаимодействия между Python-программой, выполняющейся на стороне сервера, и самим веб-сервером

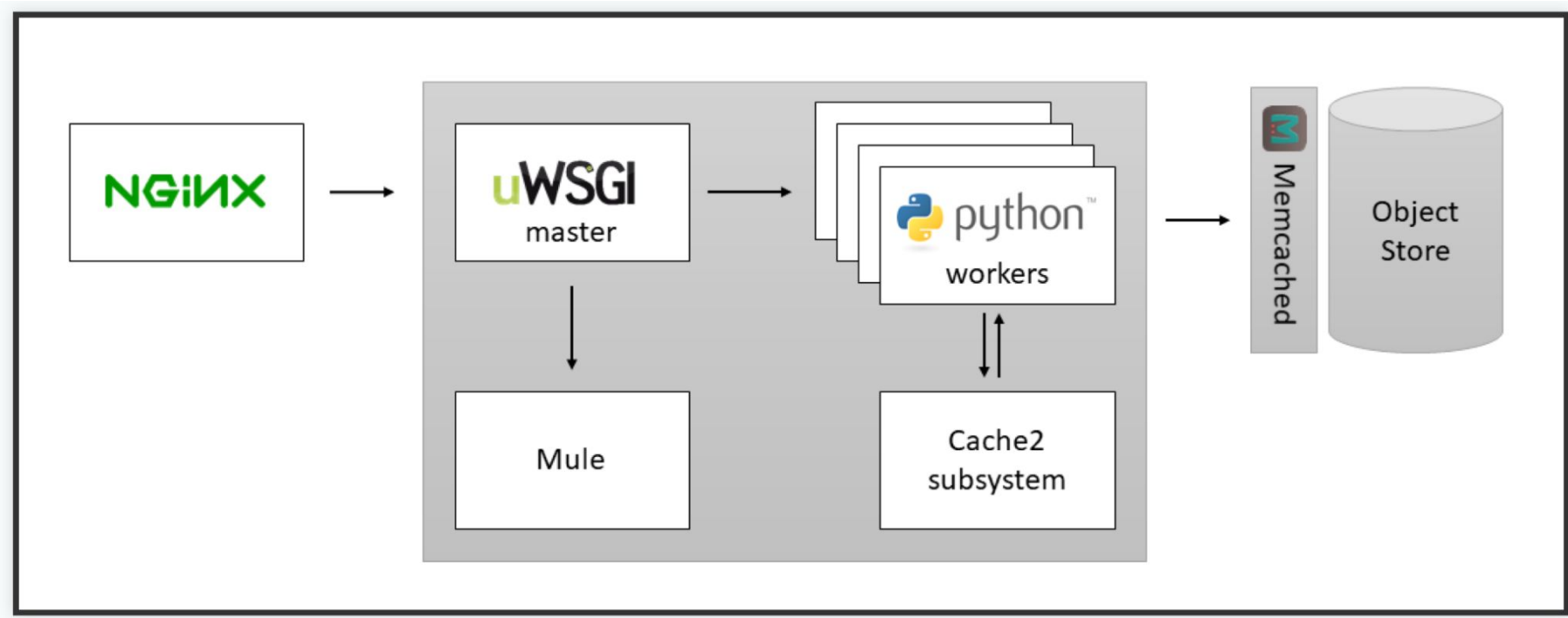
Особенности:

- Должен быть вызываемым (callable) объектом (обычно это функция или метод)
- Принимает два параметра:
  - словарь переменных окружения (environ)
  - обработчик запроса (start\_response)
  - вызывает обработчик запроса с кодом HTTP-ответа и HTTP-заголовками
  - возвращает итерируемый объект с телом ответа



# php-fpm uWSGI

# uWSGI



# Пример Flask + uWSGI + Nginx (Ubuntu 22.04)

```
# Установка Python
$ sudo apt update
$ sudo apt install python3-pip python3-dev build-essential libssl-dev libffi-dev python3-setuptools
$ sudo apt install python3-venv

# Создание папки проекта и виртуальной среды Python
$ mkdir ~/myproject
$ cd ~/myproject
$ python3.10 -m venv myprojectenv

# Активация виртуальной среды, установка пакетов
$ source myprojectenv/bin/activate
$ pip install wheel
$ pip install uwsgi flask
```



# Простое приложение (Flask)

```
# Создание приложения
$ vim ~/myproject/myproject.py
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "<h1 style='color:blue'>Hello Otus!</h1>"

if __name__ == "__main__":
    app.run(host='0.0.0.0')

# Проверка запуска приложения
$ python myproject.py
```



# Проверка

<http://Server-IP:5000>

Hello Otus!





# Точка входа приложения uWSGI

```
# Создание точки входа приложения
```

```
$ vim ~/myproject/wsgi.py
```

```
from myproject import app
```

```
if __name__ == "__main__":  
    app.run()
```

```
# Тестирование uWSGI
```

```
$ uwsgi --socket 0.0.0.0:5000 --protocol=http -w wsgi:app
```

```
$ deactivate
```

# Проверка

<http://Server-IP:5000>

Hello Otus!



# Конфигурационный файл uWSGI

```
# Установка PHP 7.3
$ vim ~/myproject/myproject.ini
[uwsgi]
module = wsgi:app

master = true
processes = 5

socket = myproject.sock
chmod-socket = 660
vacuum = true

die-on-term = true
```



# Настройка systemd

```
# Создание сервиса myproject
$ sudo vim /etc/systemd/system/myproject.service
[Unit]
Description=uWSGI instance to serve myproject
After=network.target

[Service]
User=ivan
Group=www-data
WorkingDirectory=/home/ivan/myproject
Environment="PATH=/home/ivan/myproject/myprojectenv/bin"
ExecStart=/home/ivan/myproject/myprojectenv/bin/uwsgi --ini myproject.ini

[Install]
WantedBy=multi-user.target

# Запуск сервиса myproject
$ sudo chgrp www-data /home/ivan
$ sudo systemctl start myproject
$ sudo systemctl enable myproject
$ sudo systemctl status myproject
```

# Настройка Nginx

```
# Настройка виртуального хоста Nginx
$ vim /etc/nginx/sites-available/myproject
server {
    listen 80;
    server_name demo.ru www.demo.ru;

    location / {
        include uwsgi_params;
        uwsgi_pass unix:/home/ivan/myproject/myproject.sock;
    }
}

# Добавление хоста в "активные", проверка конфига и рестарт веб-сервера
$ sudo ln -s /etc/nginx/sites-available/myproject /etc/nginx/sites-enabled
$ sudo nginx -t
$ sudo systemctl restart nginx
```

# Проверка

<http://demo.ru>

Hello Otus!

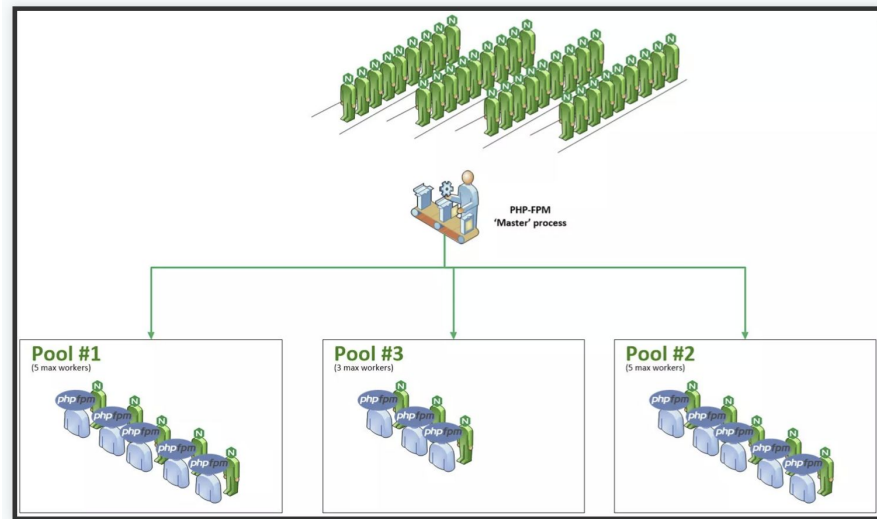


# php-fpm

**php-fpm (PHP FastCGI Process Manager)** — программный пакет, позволяющий выполнять обработку скриптов, написанных на языке PHP.

## Особенности:

- Один процесс одновременно обрабатывает один запрос
- Количество процессов определяет, сколько одновременно может «висеть» запросов в обработке



# AJAX

# WebSockets

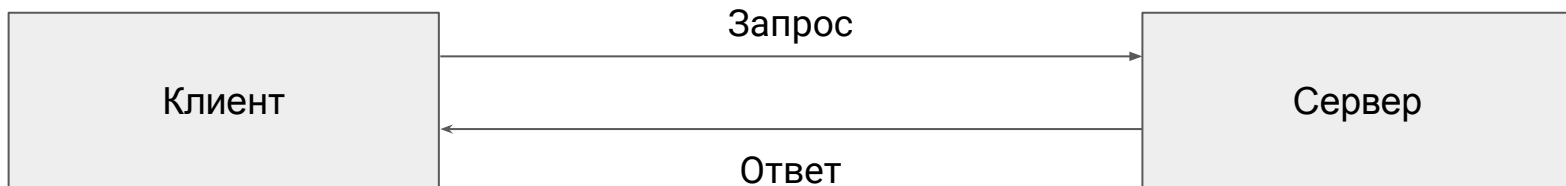


# HTTP

**HTTP (HyperText Transfer Protocol)** — широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекста.

## Особенности:

- Не поддерживает соединение после ответа на запрос
- В заголовке (HTTP Headers) требуется заранее указывать действие, которое клиент хочет выполнить (GET/POST/DELETE и т д)



# AJAX

**AJAX (Asynchronous JavaScript and XML)** — асинхронные запросы с помощью JavaScript.

## Особенности:

- Все еще не поддерживает соединение после ответа на запрос (как и HTTP-запрос)
- В заголовке (HTTP Headers) все еще требуется заранее указывать действие, которое клиент хочет выполнить (GET/POST/DELETE и т д)
- Работает асинхронно с помощью JavaScript, не требует перезагрузки страницы



# WebSocket

**WebSocket** — протокол для общения между клиентом и сервером, предоставляющий двустороннее общение поверх протокола TCP.

## Особенности:

- Поддерживает двухстороннее соединение в реальном времени
- Отправляет заголовок только один раз



# Когда обычно используют WebSocket

- Когда требуются обновления в реальном времени (мессенджеры, карты, приложения для совместной работы)

# Когда обычно НЕ используют WebSocket

- В простых приложениях (лента новостей, любые приложения, где требуется только обновить контент без получения информации взамен)

**Все хорошо?  
Есть ли вопросы?**

# Практика

# 2 версии PHP на одном хосте (CentOS 8)

## # Установка PHP 7.3

```
$ dnf install http://rpms.remirepo.net/enterprise/remi-release-8.rpm  
$ dnf module list php  
$ dnf module reset php  
$ dnf module enable php:remi-7.3  
$ dnf install php73 php73-php-fpm -y
```

## # Установка PHP 8.2

```
$ dnf module reset php  
$ dnf module enable php:remi-8.2  
$ dnf install php82 php82-php-fpm -y
```

## # Запуск сервисов

```
$ systemctl start php73-php-fpm  
$ systemctl enable php73-php-fpm  
$ systemctl status php73-php-fpm  
$ systemctl start php82-php-fpm  
$ systemctl enable php82-php-fpm  
$ systemctl status php82-php-fpm
```

# 2 версии PHP на одном хосте (CentOS 8)

```
# Подготовка директорий для сайта 1 и создания файла index
```

```
$ mkdir /var/www/sitel
```

```
$ chown -R apache:apache /var/www/sitel
```

```
$ chmod -R 755 /var/www/sitel
```

```
$ vim /var/www/sitel/info.php
```

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

```
# Настройка Apache (сайт 1)
```

```
$ vim /etc/httpd/conf.d/sitel.conf
```

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

```
    ServerAdmin admin@sitel.ru
```

```
    ServerName sitel.demo.ru
```

```
    DocumentRoot /var/www/sitel
```

```
    DirectoryIndex info.php
```

```
    ErrorLog /var/log/httpd/sitel-error.log
```

```
    CustomLog /var/log/httpd/sitel-access.log combined
```

```
    <IfModule !mod_php7.c>
```

```
        <FilesMatch \.(php|phar)$>
```

```
            SetHandler "proxy:unix:/var/opt/remi/php73/run/php-fpm/www.sock|fcgi://localhost"
```

```
        </FilesMatch>
```

```
    </IfModule>
```

```
</VirtualHost>
```



# 2 версии PHP на одном хосте (CentOS 8)

```
# Подготовка директорий для сайта 2 и создания файла index
$ mkdir /var/www/site2
$ chown -R apache:apache /var/www/site2
$ chmod -R 755 /var/www/site2
$ vim /var/www/site2/info.php
<?php phpinfo(); ?>

# Настройка Apache (сайт 2)
$ vim /etc/httpd/conf.d/site2.conf
<VirtualHost *:80>
    ServerAdmin admin@site2.ru
    ServerName site1.demo.ru
    DocumentRoot /var/www/site2
    DirectoryIndex info.php
    ErrorLog /var/log/httpd/site2-error.log
    CustomLog /var/log/httpd/site2-access.log combined
    <IfModule !mod_php7.c>
        <FilesMatch \.(php|phar)$>
            SetHandler "proxy:unix:/var/opt/remi/php82/run/php-fpm/www.sock|fcgi://localhost"
        </FilesMatch>
    </IfModule>
</VirtualHost>

# Рестарт сервиса
$ systemctl restart httpd
```

# Nginx и php-fpm на разных хостах (Centos 8)

```
# Хост 1 (nginx)
$ dnf install nginx
$ systemctl enable nginx
$ systemctl start nginx

# Настройка fastcgi
$ vim /etc/nginx/nginx.conf
...
location ~ \.php$ {
    include fastcgi_params;
    include fastcgi.conf;
    fastcgi_pass <IP>:<PORT>;
}
...

# Рестарт сервиса
$ systemctl restart nginx
```



# Nginx и php-fpm на разных хостах (Centos 8)

```
# Хост 2 (php-fpm)
$ vim /etc/opt/remi/php74/php-fpm.d/www.conf
...
listen = IP:PORT
listen.allowed_clients = IP1,IP2
...

$ systemctl restart php74-php-fpm

# !!! на втором хосте должна быть синхронизирована директория с файлами PHP,
# которые нужно выполнить
```

# Вопросы?



Ставим “+”,  
если вопросы есть



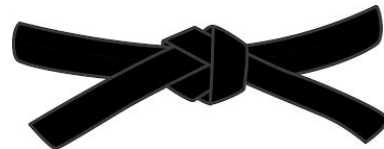
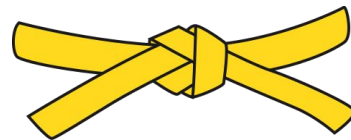
Ставим “-”,  
если вопросов нет

# Рефлексия

# Цели вебинара

К концу занятия вы сможете

1. Понять принцип работы CGI/FastCGI и websocket
2. Ориентироваться в конфигурации php-fpm и uWSGI
3. Настраивать связки nginx + php-fpm, apache + php-fpm



# Рефлексия



С какими впечатлениями уходите с вебинара?



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

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