

# Домашнее задание 6

## Настройка веб-серверов

Цель домашнего задания подготовить инфраструктуру для будущего проекта и научиться настраивать веб-серверы. *Внимание! Вместо `pupkin` используйте свою фамилию.*

### 1. Настройка gunicorn для запуска WSGI скриптов

Установите gunicorn

```
$ pip install gunicorn
```

Затем, пользуясь [документацией Gunicorn](#), напишите конфигурационный файл который запустит ваше приложение `askme_pupkin.wsgi` с двумя процессам-воркерами.

### 2. Создание простого WSGI-скрипта

Отдельно от вашего приложения создать простой WSGI-скрипт (функция или класс). Этот скрипт должен:

- запускаться с помощью gunicorn;
- выводить список переданных **GET** и **POST** параметров;
- выполняться при запросе **localhost:8081**;
- работать без использования Django.

### 3. Настройка nginx для отдачи статического контента

Необходимо настроить nginx следующим образом. Все файлы с URL начинающимся с **/uploads/** отдаются из **ask\_pupkin/uploads**. Все файлы с расширением (**.js .css .jpeg** и т.д) — из директории **ask\_pupkin/static**. Файлы должны отдаваться с заголовками, кэширующими файлы на стороне браузера. Файлы должны сжиматься на сервере для уменьшения размера передаваемых файлов. Размер конфига nginx не должен превышать 50 строк. Полученную конфигурацию необходимо запустить и проверить, для этого нужно разместить какой-либо файл (например **sample.html**) в директории **static** и загрузить его с помощью браузера **localhost/sample.html**.

### 4. Настройка проксирования в nginx

- Настроить nginx для проксирования всех нестатических запросов (URL без расширения, например **/** или **/login/**) на gunicorn;
- Настроить **upstream**;
- Настроить **proxy\_cache** и проверить его работу.

### 5. Сравнение производительности

С помощью утилиты Apache Benchmark (**ab**, идет в комплекте с Apache, для Ubuntu пакет **apache2-utils**) или **wrk** сравните производительность nginx (отдача статики) и gunicorn (запуск wsgi скриптов).

Необходимо провести пять измерений:

- Отдача статического документа напрямую через nginx;
- Отдача статического документа напрямую через gunicorn
- Отдача динамического документа напрямую через gunicorn;
- Отдача динамического документа через проксирование запроса с nginx на gunicorn;
- Отдача динамического документа через проксирование запроса с nginx на gunicorn, при кэшировании ответа на nginx (proxy cache).

**Размеры всех документов должны быть примерно одинаковыми.** По результатам измерений необходимо ответить на вопросы:

- Насколько быстрее отдается статика по сравнению с WSGI?
- Во сколько раз ускоряет работу proxy\_cache?

## 6. Результат выполнения домашнего задания

Результат выполнения домашнего задания является:

- директория с созданным проектом;
- конфиг nginx;
- конфиг gunicorn;
- результаты нагрузочного тестирования (вывод утилиты ab/wrk).

## 7. Полезные ссылки

- [Nginx](#).
- [О проксировании в nginx](#);
- [Gunicorn](#);
- [ab](#).