

# Настройка веб-сервера Nginx

*Докладчик: Григорий Кострюков*

# Ссылки

- Официальная документация ([EN/RU](#))
- [Руководство администратора \(EN\)](#)
- [Руководство для начинающих](#)
- [Более подробно о проксировании запросов](#)

# Назначение технологии

Веб-сервера изначально предназначены для обслуживания статических страниц, например HTML и CSS. Позднее появилась возможность переадресовывать запросы к серверам приложений, отвечающих за генерацию динамического содержимого путём выполнения кода.

Nginx позиционируется производителем как простой, быстрый и надёжный сервер, не перегруженный функциями.

Применение nginx целесообразно прежде всего для статических веб-сайтов и как обратного прокси-сервера перед динамическими сайтами

# Установка и запуск

Необходимо установить пакет **nginx** или **nginx-full** (расширенная версия)

```
apt install nginx
```

Управлять службой можно с помощью команд `systemd`

```
systemctl start nginx
```

Также, запустив исполняемый файл приложения, можно контролировать работу с помощью внутренних команд `nginx` (см. руководство для начинающих)

# Формат конфигурационного файла

Nginx состоит из модулей, которые настраиваются директивами, указанными в конфигурационном файле. Директивы делятся на *простые* и *блочные*

- Простая состоит из имени и параметров, разделённых пробелами, и оканчивается точкой с запятой
- У блочной вместо точки с запятой после имени и параметров следует набор дополнительных инструкций, помещённых внутри фигурных скобок.

Если у блочной директивы внутри фигурных скобок можно задавать другие директивы, то она называется *контекстом*.

Часть строки после символа '#' считается комментарием.

```
user nobody; # a directive in the 'main' context

events {
    # configuration of connection processing
}

http {
    # Configuration specific to HTTP for all virtual servers

    server {
        # configuration of HTTP virtual server 1

        location /one {
            # processing URIs starting with '/one'
        }

        location /two {
            # processing URIs starting with '/two'
        }
    }

    server {
        # configuration of HTTP virtual server 2
    }
}
```

# Конфигурация виртуального сервера

Для описания виртуального сервера используется директива **server** внутри контекста **http**. Контекст **http** может содержать несколько таких блоков для описания различных серверов.

```
http {  
    server {  
        # Server configuration  
    }  
}
```

Внутри блока **server** может быть указан IP адрес и порт, по которым сервер слушает запрос. Поддерживается как IPv4, так и IPv6 формат. В примере показан сервер, настроенный на ip-адрес **172.0.0.1** и порт **8080**

```
server {  
    listen 127.0.0.1:8080;  
    # Additional server configuration  
}
```

Если порт не указан явно, то используется стандартный. При отсутствии адреса, обрабатывается любой адрес. Если директива **listen** не указана, используется стандартный порт 80 или 8000 в зависимости от привилегий пользователя.

# Конфигурация виртуального сервера

Имена сервера задаются с помощью директивы `server_name` и определяют, в каком блоке `server` будет обрабатываться тот или иной запрос. Имена могут быть заданы точно, с помощью маски или регулярного выражения:

```
server {  
    listen 80;  
    server_name example.org www.example.org;  
    #...  
}
```

# Конфигурация виртуального сервера

Ngинх может пересылать запросы или предоставлять различные файлы в зависимости от переданного URI. Для этого используется директива `location` внутри `server`. Её параметром может служить как префикс, так и регулярное выражение.

```
location /some/path/ {  
    #...  
}  
  
location ~ /\.html {  
    #...  
}
```

В первом случае будут обработаны все запросы, начинающиеся с `/some/path`.

Например, `/some/path/cat.png`. Во втором – содержащие подстроку `.html`.

Замечание: если URI удовлетворяет сразу нескольким обработчикам, используется обработчик с [большим приоритетом](#).

# Структура каталогов веб-сервера

Администрирование сервера nginx в основном заключается в настройке и поддержке его файлов конфигурации, которые находятся в папке `/etc/nginx`.

- `/etc/nginx/nginx.conf` – главный файл конфигурации nginx.
- `/etc/nginx/conf.d` - общие конфигурационные параметры.

Местоположение подключаемых файлов определяется с помощью директивы `include`. В качестве параметра передаётся путь к файлу конфигурации

```
include /etc/nginx/sites-enabled/*;
```

# Структура каталогов веб-сервера

- **`/etc/nginx/modules-available`** – модули, доступные для подключения к основной конфигурации
- **`/etc/nginx/modules-enabled`** - модули, которые надо включить в конфигурацию nginx. Содержит символические ссылки на файлы из каталога **`modules-available`**
- **`/etc/nginx/sites-available`** – конфигурации виртуальных хостов, т.е. каждый файл, находящийся в этом каталоге, содержит информацию о конкретном сайте – его имени, IP адресе, рабочей директории и т.д.
- **`/etc/nginx/sites-enabled`** – ссылки на конфигурации сайтов из **`sites-available`**, которые будут обслуживаться nginx.

# Предоставление статического контента

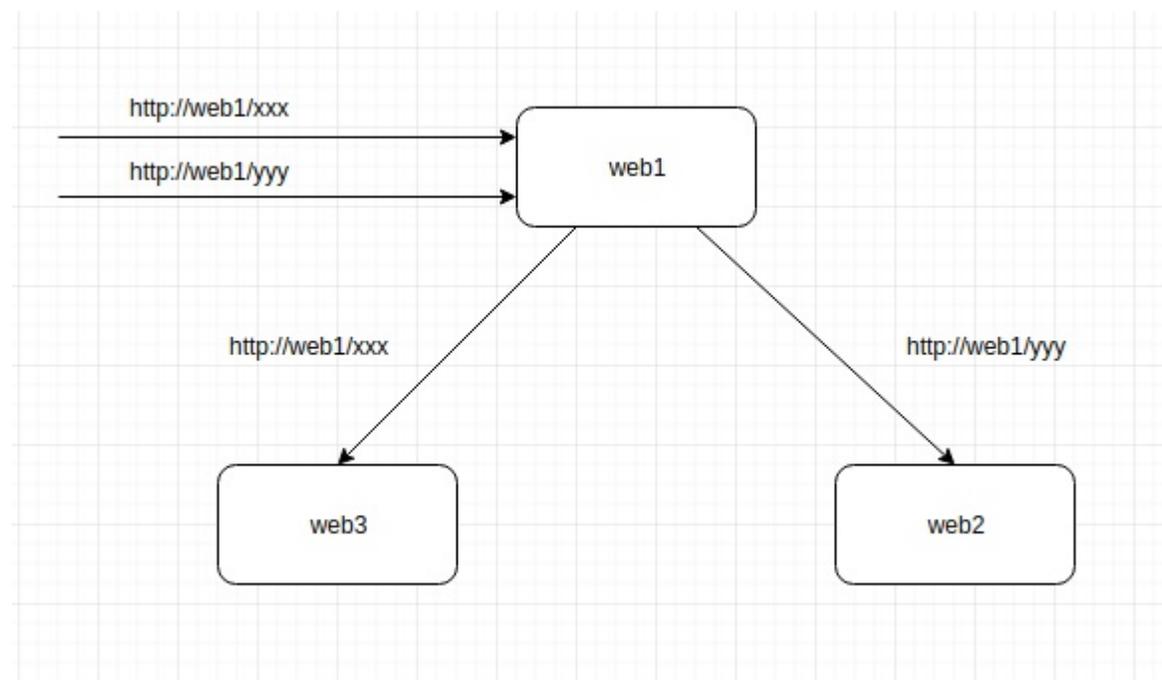
Одна из важных задач конфигурации nginx — раздача файлов, таких как изображения или статические HTML-страницы. Директива `root` указывает, в каком каталоге должен осуществляться поиск файла. Nginx формирует путь к файлу, добавляя текущий URI к пути, указанному в `root`.

```
server {  
    root /var/www/data;  
  
    location / {  
    }  
  
    location /images/ {  
    }  
  
    location ~ /\.mp3 {  
        root /www/media;  
    }  
}
```

Например, на запрос <http://localhost:80/images/cat.jpg> будет предоставлен файл `/var/www/data/images/cat.jpg`

# Обратное проксирование запросов

Иногда бывает нужно чтобы различные url запросы обрабатывались на разных серверах, но первоначально приходили на один сервер. Например вы пробросили порт на своем роутере на один веб-сервер в вашей внутренней сети. Но хотите чтобы каталог `/xxx` открывался на втором веб-сервере, а `/yyy` открывался на третьем, и не хотите пробрасывать порты на каждый web-сервер. Или же если помимо статического контента у вас имеется сервер приложения, к которому тоже нужно предоставить доступ.



# Обратное проксирование запросов

Когда nginx проксирует запрос, он посылает его на указанный проксируемый сервер, получает ответ и пересылает обратно клиенту. В качестве адреса «внутреннего» сервера может быть указано как доменное имя, так и ip-адрес. Данный адрес может включать в себя и конкретный порт

```
location /some/path/ {  
    proxy_pass http://www.example.com/link/;  
}
```

Например, запрос `/some/path/page.html` будет перенаправлен на `http://www.example.com/link/page.html`. Также nginx позволяет конфигурировать заголовок запроса при пересылке, обращаться к не-HTTP серверам, буферизировать ответы и т.д.

# Задания

1. В каталоге `/var/www/html` создайте html-документ `index.html` с приветствием для пользователя. Можно использовать [шаблон Foundation](#) (для этого загрузите архив и распакуйте его в указанный каталог). Настройте nginx, чтобы `/var/www/html` был корневым каталогом сайта (при обращении к <http://ip-виртуальной-машины:80> отображался созданный ранее документ). Для этого:

- Создайте конфигурационный файл `sample-site` для виртуального сервера в каталоге `/etc/nginx/sites-available` (в качестве шаблона можно использовать `default`). Сервер должен слушать порт 80.
- Замените ссылку `default` в `/etc/nginx/sites-enabled` на ссылку на `sample-site`
- Перезапустите службу nginx, чтобы применить внесённые изменения

Замечание: с помощью команды `nginx -t` можно проверить корректность текущей конфигурации

# Задания

2. В каталоге `/var/www/images` разместите несколько изображений формата png или jpg. Сконфигурируйте nginx так, чтобы для запросов, содержащих имена файлов с расширениями .jpg или .png корневым каталогом был **`/var/www/images`**.

3. Создайте ещё один виртуальный сервер, слушающий порт **8080** и работающий в рамках того же экземпляра nginx. Перенаправьте на него все запросы, не содержащие файлов с расширениями .jpg или .png. Создайте свою страницу с сообщением об ошибке в случае обращения по несуществующему адресу и настройте её отображение.