



Веб-сервера



Веб-сервера

- Понятие Веб-сервера может относиться как к железу, так и к программному обеспечению (ПО).
 - С точки зрения железа Веб-сервер — это компьютер, который хранит ресурсы сайта (HTML документы, CSS, JavaScript, PHP, Java и т.д.) и доставляет их на устройство конечного пользователя (веб-браузер и т.д.).
 - С точки зрения ПО, Веб-сервер контролируют доступ пользователей к размещенным на сервере файлам, обрабатывает HTTP запросы или проксирует их к серверному программному обеспечению.



Порты

- Как правило веб-сервера используют следующие порты:
 - Стандартные - 80, 443
 - Часто используемые - 8080, 8443



Распространенные веб-сервера

- Apache HTTP



- Nginx



- lighttpd





Клиент-серверное взаимодействие

- HTTP запрос/ответ

Клиент



Веб-сервер



Session Request (TCP port 80)



Session Response



GET Request



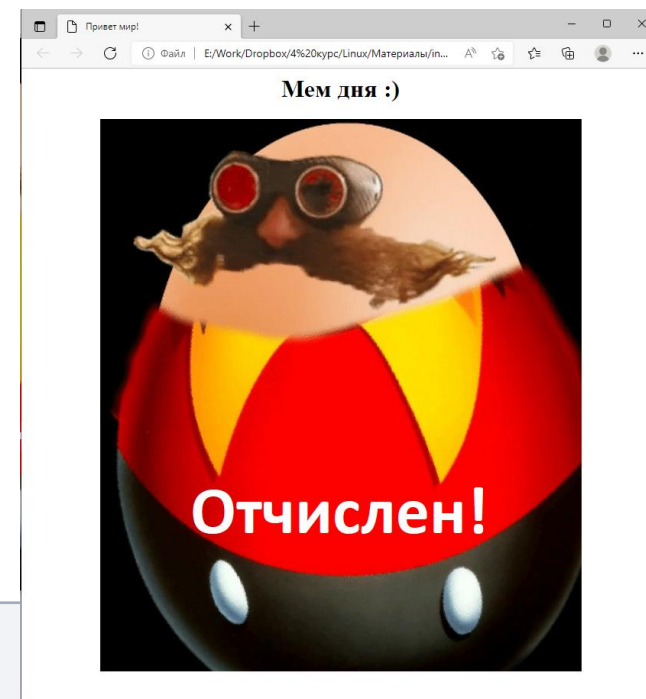
GET Response



Цикл, до тех пор,
пока все файлы
не загрузятся

index.html

```
<html>
  <head>
    <title>Привет мир!</title>
  </head>
  <body>
    <div align="center">
      <h1>Мем дня :)</h1>
      
    </div>
  </body>
</html>
```





Apache HTTP



Конфигурация Apache HTTP / CentOS

- Директория основной конфигурации
 - /etc/httpd
- Основная конфигурация
 - /etc/httpd/conf/*
- Дополнительные конфигурационные файлы:
 - /etc/httpd/conf.d/*.conf
 - /etc/httpd/conf.modules.d/*.conf



Конфигурация Apache HTTP / Ubuntu

- Директория основной конфигурации
 - /etc/apache2
- Основная конфигурация
 - /etc/apache2/*.conf
- Дополнительные конфигурационные файлы:
 - /etc/apache2/conf-*/*
 - /etc/apache2/mods-*/*
 - /etc/apache2/sites-*/*



Virtual Hosts

- Структура:

```
<VirtualHost *:80>  
    ServerName "uvp.local"  
    DocumentRoot "/var/www/uvp"  
</VirtualHost>
```



Virtual Hosts и логирование

- Структура:

```
<VirtualHost *:80>  
    ServerName "mysite.ru"  
    DocumentRoot "/var/www/html"  
    ErrorLog "logs/mysite_error.log"  
    CustomLog "logs/ mysite_access.log" common  
</VirtualHost>
```



Virtual Hosts и доступ по IP

- Структура:

```
<VirtualHost *:80>  
    ServerName "mysite.ru"  
    DocumentRoot "/var/www/html"  
    <Location />  
        Require ip 127.0.0.1  
    </Location>  
</VirtualHost>
```



Virtual Hosts и управление доступом

- Структура:

```
<VirtualHost *:80>
    ServerName "mysite.ru"
    DocumentRoot "/var/www/html"
    <Directory "/var/www/html">
        Require valid-user
        AuthType Basic
        AuthName "Private Access"
        AuthBasicProvider file
        AuthUserFile "/etc/httpd/conf.d/key"
    </Directory>
</VirtualHost>
```



apachectl

- Интерфейс управления HTTP-сервером Apache
- Синтаксис:

```
apachectl [command]
```

- Пример:

```
# Проверяет корректность конф. файлов  
apachectl configtest
```



Работа с Apache HTTP

- Обновим реестр пакетов:

```
sudo apt-get update
```

- Ставим apache2:

```
sudo apt-get install -y apache2
```

- Проверяем статус:

```
systemctl status apache2
```



Работа с Apache HTTP

- Установим дополнительные пакеты:

```
sudo apt-get install -y psmisc tree
```

- Посмотрим дерево процессов:

```
pstree -psT $(pgrep -U 0 apache2)
```

- И дерево файлов (внимательно рассмотрите директории):

```
tree /etc/apache2
```



Работа с Apache HTTP

- Проверить текущие конфигурации можно:

```
apachectl configtest
```

- Apache вам скажет что стоило бы дать FQDN имя серверу, поэтому:

```
sudo nano /etc/apache2/apache2.conf
```

- И в файле укажите уникальное имя:

```
ServerName uvp.lab
```




Работа с Apache HTTP

- Также установим сетевое имя компьютера в /etc/hosts:

```
<ваш ip сервера> uvp.lab
```

- Далее установим веб-браузер для консольной строки:

```
sudo apt-get install -y lynx
```

- И обратимся к странице (для выхода нажмите q):

```
lynx http://uvp.lab
```



Работа с Apache HTTP

- Настроим удаленный доступ (добавим правило в фаервол):

```
sudo ufw allow 80/tcp
```

- Далее попробуйте открыть страницу в браузере в Windows:

```
http://<ваш ip VM>
```

- Изменим html доступный по умолчанию:

```
echo '<h1>My first deploy!</h1>' | sudo tee /var/www/html/index.html
```

- И обновим страницу в браузере.



Работа с Apache HTTP

- А что если нам нужно заhostить несколько сайтов? Отредактируйте конфигурацию по умолчанию:

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

- Измените параметр:

```
ServerName uvp.lab
```

- Проверим конфигурацию:

```
apachectl configtest
```



Работа с Apache HTTP

- Скопируйте файл `/etc/apache2/sites-available/000-default.conf` в новый файл - `/etc/apache2/sites-available/001-mysite.conf` и измените в нем:

```
<VirtualHost *:80>  
    ServerName mysite.lab  
    DocumentRoot /var/www/mysite  
</VirtualHost>
```

- Включите данный сайт:

```
sudo a2ensite 001-mysite.conf
```



Работа с Apache HTTP

- Проверим конфигурацию:

```
apachectl configtest
```

- Нам скажут что директория /var/www/mysite не существует
- Создаем:

```
sudo mkdir /var/www/mysite
```

- И проверяем:

```
apachectl configtest
```



Работа с Apache HTTP

- Ловко создадим файл сайта в его директории:

```
echo "<h1>My Site</h1>" | sudo tee /var/www/mysite/index.html
```

- Рестартим сервер:

```
sudo systemctl restart apache2
```

- Добавим строку в /etc/hosts:

```
sudo echo '127.0.0.1      mysite.lab' | sudo tee -a /etc/hosts
```



Работа с Apache HTTP

- Проверяем:

```
lynx http://uvp.lab
```

```
lynx http://mysite.lab
```

- Также можно альтернативно использовать команду curl:

```
curl http://uvp.lab
```

```
curl http://mysite.lab
```



NGINX HTTP



Конфигурация NGINX HTTP / CentOS

- Директория основной конфигурации
 - /etc/nginx
- Основная конфигурация
 - /etc/nginx/*.conf
- Дополнительные конфигурационные файлы:
 - /etc/nginx/conf.d/*.conf
- Журналы сервера
 - /var/log/nginx/access.log
 - /var/log/nginx/error.log
- Контент
 - /usr/share/nginx/html



Конфигурация NGINX HTTP / Ubuntu

- Директория основной конфигурации
 - /etc/nginx
- Основная конфигурация
 - /etc/nginx/*.conf
- Дополнительные конфигурационные файлы:
 - /etc/nginx/sites-*/*
 - /etc/nginx/snippets
- Журналы сервера
 - /var/log/nginx/access.log
 - /var/log/nginx/error.log
- Контент
 - /var/www/html



Virtual Hosts

Что такое виртуальный хост?

Это конфигурационный файл, который описывает настройку условно одного домена.

После дефолтной установки nginx у вас уже будет один виртуальный хост, который описан конфигом *default.conf*.

```
/etc/nginx/sites-available/default.conf
```

Обычно конфиги с виртуальными хостами расположены в директории .

```
/etc/nginx/conf.d
```



Virtual Hosts

- Структура:

```
server {  
    listen 80 default_server; # Порты для IPv4 клиентов  
    listen [::]:80 default_server; # Порты для IPv6 клиентов  
    root /var/www/html; # Место хранения файлов сайта  
    index index.html index.htm; # Файлы для отображения по умолчанию (при  
    указании только доменного имени в адресе)  
    server_name _; # Доменное имя сайта  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```



Virtual Hosts и логирование

- Структура:

```
server {  
    listen 80 default_server; # Порты для IPv4 клиентов  
    listen [::]:80 default_server; # Порты для IPv6 клиентов  
    root /var/www/example.com;  
    access_log /var/log/nginx/example.com-access.log main;  
    error_log /var/log/nginx/example.com-error.log error;  
    index index.html index.htm;  
    server_name example.com;  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```



Интерфейс управления HTTP-сервером Nginx

- Синтаксис:

```
nginx [option]
```

- Пример:

```
# Проверяет корректность конф. файлов  
nginx -t  
# Проверяет используемую версию сервера  
nginx -v
```



Работа с Nginx HTTP

- Обновим реестр пакетов:

```
sudo apt-get update
```

- Ставим Nginx:

```
sudo apt install nginx
```

- Проверяем статус:

```
systemctl status nginx
```



Работа с Nginx HTTP

- Скопируем дефолтную конфигурацию :

```
cp /etc/nginx/sites-available/default /etc/nginx/sites-available/domain-name.com.conf
```

- Отредактируем её:

```
nano /etc/nginx/sites-available/domain-name.com.conf
```

- В файле укажите уникальное имя сайта и место расположения сайта:

```
root /var/www/domain-name.com/html  
server_name domain-name.com
```




Работа с Nginx HTTP

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
    root /var/www/domain-name.com/html;  
    server_name domain-name.com;  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```



Работа с Nginx HTTP

- Создаём директорию для хранения файлов вебсайта:

```
sudo mkdir -p /var/www/domain-name.com/html
```

- Устанавливаем права:

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



Работа с Nginx HTTP

- Создаём символическую ссылку на файл конфигурации в директории `/etc/nginx/sites-enabled/` :

```
ln -s /etc/nginx/sites-available/domain-name.com.conf  
/etc/nginx/sites-enabled/domain-name.com.conf
```

- Проверяем конфигурацию на ошибки:

```
nginx -t
```

- Перезагружаем сервер:

```
sudo systemctl restart nginx
```



Работа с Nginx HTTP

- Также установим сетевое имя компьютера в /etc/hosts:

```
<ваш ip сервера> domain-name.com
```

- Далее установим веб-браузер для консольной строки:

```
sudo apt-get install -y lynx
```

- И обратимся к странице (для выхода нажмите q):

```
lynx http://domain-name.com
```



Работа с Nginx HTTP

- Создадим и отредактируем файл сайта в его директории:

```
sudo nano /var/www/domain-name.com/html/index.html
```

- Пропишем:

```
Hello - domain-name.com
```



Работа с Nginx HTTP

- Настроим удаленный доступ (добавим правило в фаервол):

```
sudo ufw allow 80/tcp
```

- Далее попробуйте открыть страницу в браузере в Windows:

```
http://<ваш IP адрес>
```



Работа с Nginx HTTP

- А что если нам нужно заhostить несколько сайтов?

- Останавливаем сервер: `sudo systemctl stop nginx`

- Скопируем дефолтную конфигурацию :

```
cp /etc/nginx/sites-available/default /etc/nginx/sites-available/domain-name1.com.conf
```

- Отредактируем её:

```
nano /etc/nginx/sites-available/domain-name1.com.conf
```

- В файле укажите уникальное имя нового сайта и место расположения нового сайта:

```
root /var/www/domain-name1.com/html
server_name domain-name1.com
```



Работа с Nginx HTTP

- Проверим конфигурацию:

```
nginx -t
```

- Создаем:

```
sudo mkdir -p /var/www/domain-name1.com/html
```

- И проверяем:

```
nginx -t
```




Работа с Nginx HTTP

- Создадим и отредактируем файл сайта в его директории:

```
sudo nano /var/www/domain-name1.com/html/index.html
```

- Пропишем:

```
Hello - it is not domain-name.com
```



Работа с Nginx HTTP

- Создаём символическую ссылку на файл конфигурации в директории /etc/nginx/sites-enabled/ :

```
ln -s /etc/nginx/sites-available/domain-name1.com.conf  
/etc/nginx/sites-enabled/domain-name1.com.conf
```

- Проверяем конфигурацию на ошибки:

```
nginx -t
```

- Перезагружаем сервер:

```
sudo systemctl restart nginx
```



Работа с Nginx HTTP

- Добавим строку в /etc/hosts:

```
sudo echo '127.0.0.1      domain-name1.com' | sudo tee  
-a /etc/hosts
```



Работа с Nginx HTTP

- Проверяем:

```
lynx http://domain-name1.com
```

```
lynx http://domain-name.com.lab
```

- Также можно альтернативно использовать команду curl:

```
curl http://domain-name1.com
```

```
curl http://domain-name.com
```



Есть вопросы?

miitqa@outlook.com