

Задание №4. Запускаем свой Веб-сервер

Немного теории про NGINX про DNS

У вашего виртуального сервера Linux есть свой IP-адрес. Его можно посмотреть с помощью команды `ip address`:

```
ip address
```

В выводе будет:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether bc:24:11:bb:d6:c6 brd ff:ff:ff:ff:ff:ff
    inet 10.255.37.127/24 metric 100 brd 10.255.37.255 scope global dynamic
eth0
    valid_lft 226sec preferred_lft 226sec
    inet6 fe80::be24:11ff:febb:d6c6/64 scope link
        valid_lft forever preferred_lft forever
```

IP-адреса бывают «белые», т.е. доступные из сети Интернет. И «серые», т.е. недоступные из сети Интернет внутренние адреса. На вашем сервере «серый» IP-адрес. IP-адреса, которые начинаются на 10.x.x.x – серые.

Для того, чтобы появилась возможность обращаться к вашему серверу из сети Интернет используется прокси-сервер Nginx. У этого прокси-сервера «белый» IP-адрес и он доступен из сети Интернет.

Обращение пользователей к веб-страницами происходит не по IP-адресам, а по именам. Например, `data-pool.ru`. Когда вы пишете в строке браузера `data-pool.ru` браузер обращается к DNS-серверу. Задача DNS-серверов – преобразование имен в IP-адреса. В базах данных серверов DNS хранятся таблицы соответствия имен IP-адресам.

Вашему виртуальному серверу в нашей системе присвоено имя и для этого имени на DNS-сервере прописано соответствие прокси-серверу Nginx. Т.е. когда пользователь сети Интернет напишет в браузере имя вашего сервера то попадет на прокси-сервер Nginx. Прокси-сервер примет запрос и, в зависимости от имени отправит запрос на внутренний IP-адрес вашего виртуального сервера.

Important: { тут схема }

Установка веб-сервера

Для того, чтобы ваш виртуальный сервер смог обработать входящий запрос пользователя (прошедший через nginx) у вас на сервере должен быть запущен веб-сервер. Будем использовать Apache. Установим Apache через apt:

```
sudo apt update
sudo apt install apache2
```

проверим, что установили:

```
apt list | grep -E 'apache2.*installed'
```

Должны увидеть в списке строку:

```
Apache2/stable, now ... [installed]
```

Настройки веб-сервера лежат в каталоге `/etc/apache2/`

Посмотрим настройки. Надо ли что-то менять?

Файлы веб-сервера лежат в каталоге `/var/www/`

Запустим **mc**, идем в `/var/www`, там есть папка `html` и в ней один файл: **index.html**. Этот файл отобразится при открытии страницы вашего сервера.

Демоны

Демоны в Linux это программы, которые запускаются системой и работают в фоновом режиме. Веб-сервер должен быть демоном. Он должен быть всегда запущен и ждать входящего обращения.

Проверим какие демоны у нас запущены (так как у обычного пользователя нет доступа к системным функциям, выполним функцию от имени администратора с помощью `sudo`):

```
sudo systemctl status
```

Увидим дерево запущенных демонов системы. Для выхода нажмем `q`.

Проверим отдельно запущен ли у нас демон Apache:

```
sudo systemctl status apache2
```

В выводе должен быть зеленый текст «active (running)»

Для остановки демона Apache введем команду:

```
sudo systemctl stop apache2
```

Снова посмотрим статус.

Запустим демона Apache командой:

```
sudo systemctl start apache2
```

И снова проверим статус.

Если демон Apache запущен значит наш веб-сервер работает и ждёт запросов. Проверим это – зайдём на свою страницу из сети Интернет. Откройте в браузере:

```
bsk23-01-1.sibgu-itlab.data-pool.ru
```

Вы попадете на дефолтную страницу веб-сервера apache = index.html

Первая html-страница

Сделаем свою первую простую веб-страницу.

Перейдите в папку /var/www/html:

```
cd /var/www/html
```

Создайте файл first.html:

```
sudo nano first.html
```

В связи с тем, что права на запись в папку веб-сервера ограничены для обычного пользователя, используем sudo для выполнения команды от имени администратора.

В файле first.html напишите любой текст и сохраните файл.

Откройте в браузере ссылку:

```
bsk23-01-1.sibgu-itlab.data-pool.ru/first.html
```

Вы увидите свой текст на своём сайте.

Для того, чтобы писать более сложные страницы сайта нужно разобраться с html.

База по HTML

Note: HTML – не язык программирования. Это язык разметки.

Страница HTML состоит из элементов. Элементы завернуты в теги.

В начале элемента стоит открывающий тэг `<xxx>`. В конце элемента стоит закрывающий тег `</xxx>`. Где `xxx` - тип тега.

Например, тег `<p>` оформляет текст внутри элемента отдельным абзацем. Наберите в html-файле следующий текст:

```
<p>Текст в отдельном абзаце №1</p>
<p>Текст в отдельном абзаце №2</p>
```

Тег `` делает текст внутри элемента жирным. Добавьте в html-файл строку:

```
<p>Текст в отдельном абзаце №1</p>
<p>Текст в отдельном абзаце №2</p>
<b>Жирный текст</b>
```

Теги могут быть вложенными один в другой. Например, жирный текст может быть внутри абзаца.

```
<p>Текст в отдельном абзаце №1</p>
<p>Текст <b>жирный</b> в отдельном абзаце №2</p>
<b>Жирный текст</b>
```

Структура HTML

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Пример веб-страницы</title>
  </head>
  <body>
    <h1>Заголовок</h1>
    <!-- Комментарий -->
    <p>Первый абзац.</p>
    <p>Второй абзац.</p>
  </body>
</html>
```

Таблицы в HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```
<title>Тег TABLE</title>
</head>
<body>
  <table border="1" width="100%" cellpadding="5">
    <tr>
      <th>Ячейка 1</th>
      <th>Ячейка 2</th>
    </tr>
    <tr>
      <td>Ячейка 3</td>
      <td>Ячейка 4</td>
    </tr>
  </table>
</body>
</html>
```

Скрипты для веб-сервера

Выполните команду для включения модуля CGI (Common Gateway Interface – используется для запуска скриптов веб-сервером):

```
sudo a2enmod cgi
```

Перезапустите веб-сервер, чтобы изменения вступили в силу:

```
sudo systemctl restart apache2
```

Скрипты веб-сервера лежат в каталоге `/usr/lib/cgi-bin`.

Перейдите в каталог скриптов:

```
cd /usr/lib/cgi-bin
```

От имени администратора (`sudo`) создайте в нём файл скрипта `first.py`. Сделайте его исполняемым:

```
sudo chmod +x /usr/lib/cgi-bin/first.py
```

Проверьте права командой `ls` с ключом `-l`:

```
ls -l
```

Если имя файла `first.py` подсвечено зеленым цветом значит файл стал исполняемым.

Note: {Тут нужны подробности про права на файлы и каталоги в Linux}

В файл напишите следующее:

```
#!/usr/bin/python3

print ("Content-type: text/html")
print ("")
print ("")
print ("My first Web Python")
```

Сохраните.

Первая строка скрипта указывает на программу, которая будет выполнять данный скрипт. Далее 3 строки с командой print выдают веб-серверу информацию о том, что скрипт возвращает html-страницу.

Теперь откройте в браузере страницу bsk23-01-1.sibgu-itlab.data-pool.ru/cgi-bin/first.py (замените цифру в имени сайта на вашу)

В окне браузера вы должны увидеть текст «My first Web Python».

From:
<https://sibgu-itlab-wiki.data-pool.ru/> - **SIBGU-ITLAB-WIKI**

Permanent link:
https://sibgu-itlab-wiki.data-pool.ru/zadanie_4._zapuskaem_svoj_veb-server?rev=1711167063

Last update: **2024/03/23 04:11**

