Установка Apache HTTP Server в Ubuntu 22.04

Введение

Apache HTTP Server является наиболее широко используемым веб-сервером в мире. Он предоставляет множество мощных функций, включая динамически загружаемые модули, надежную поддержку мультимедиа и обширную интеграцию с другим популярным программным обеспечением.

В этом руководстве мы проследим, как установить Apache HTTP Server на сервер Ubuntu 22.04.

Подключимся к нашему серверу через программу PuTTY, введем логин указанный при установке нашего сервера и пароль.

```
🚜 alisa@linux: ~
                                                                               X
                                                                         login as: alisa
  alisa@192.168.1.145's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-97-generic x86 64)
 * Documentation: https://help.ubuntu.com
  Management:
                  https://landscape.canonical.com
 * Support:
                 https://ubuntu.com/advantage
 System information as of Wed Feb 28 01:24:43 PM UTC 2024
 System load: 0.0
                                 Temperature:
                                                           41.0 C
               7.1% of 97.87GB
                                                           195
 Usage of /:
                                  Processes:
 Memory usage: 1%
                                 Users logged in:
  Swap usage:
                                 IPv4 address for enp6s0: 192.168.1.145
Expanded Security Maintenance for Applications is not enabled.
59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
alisa@linux:~$
```

Предварительные условия

Прежде чем приступить к работе с этим руководством, на вашем сервере должен быть настроен обычный пользователь без полномочий root с привилегиями sudo. Кроме того, вам необходимо включить базовый брандмауэр для блокировки несущественных портов. Вы можете узнать, как настроить учетную запись обычного пользователя и настроить брандмауэр для вашего сервера, следуя нашему руководству по начальной настройке сервера для Ubuntu 20.04.

Если у вас есть доступная учетная запись, для начала войдите в систему как пользователь без полномочий root.

Установка Apache

Apache доступен в репозиториях программного обеспечения Ubuntu по умолчанию, что позволяет установить его с помощью обычных инструментов управления пакетами.

Начнем с обновления локального индекса пакетов, чтобы отразить последние изменения исходного кода:

sudo apt update

```
alisa@linux:~$ sudo apt update
[sudo] password for alisa:
```

Введем повторно пароль и дождемся окончания обновления индекса пакетов.

```
alisa@linux:~$ sudo apt update
[sudo] password for alisa:
Hit:1 http://ru.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ru.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ru.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://ru.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
58 packages can be upgraded. Run 'apt list --upgradable' to see them.
alisa@linux:~$
```

Установим непосредственно сам пакет Apache HTTP Server

```
sudo apt install apache2
```

```
alisa@linux:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 apache2-bin apache2-data apache2-utils bzip2 libaprl libaprutill
  libaprutill-dbd-sqlite3 libaprutill-ldap liblua5.3-0 mailcap mime-support
  ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
 bzip2-doc
The following NEW packages will be installed:
 apache2 apache2-bin apache2-data apache2-utils bzip2 libaprl libaprutill
 libaprutill-dbd-sqlite3 libaprutill-ldap liblua5.3-0 mailcap mime-support
  ssl-cert
 upgraded, 13 newly installed, 0 to remove and 58 not upgraded.
Need to get 2,139 kB of archives.
After this operation, 8,518 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

После подтверждения установки apt install

```
Do you want to continue? [Y/n] **Y**
```

будет установлен Apache и все необходимые зависимости.

Настройка брандмауэра

Перед тестированием Apache необходимо изменить настройки брандмауэра, чтобы разрешить внешний доступ к веб-портам по умолчанию. Предполагая, что вы следовали инструкциям, указанным в предварительных требованиях, у вас должен быть настроен брандмауэр UFW, ограничивающий доступ к вашему серверу.

Во время установки Apache регистрируется в UFW, чтобы предоставить несколько профилей приложений, которые можно использовать для включения или отключения доступа к Apache через брандмауэр.

Перечислите **ufw** профили приложений, набрав:

sudo ufw app list

```
alisa@linux:~$ sudo ufw app list
Available applications:
Apache
Apache Full
Apache Secure
OpenSSH
alisa@linux:~$
```

Как видно из результатов, для Apache доступны три профиля:

Арасhe: этот профиль открывает только порт 80 (обычный незашифрованный веб-

трафик).

- Apache Full: этот профиль открывает как порт 80 (обычный незашифрованный вебтрафик), так и порт 443 (зашифрованный трафик TLS/SSL).
- Apache Secure : этот профиль открывает только порт 443 (трафик с шифрованием TLS/SSL).

Рекомендуется включить наиболее ограничительный профиль, который по-прежнему будет разрешать настроенный вами трафик. Поскольку в этом руководстве мы еще не настроили SSL для нашего сервера, нам нужно будет разрешить трафик только через порт 80:

sudo ufw allow 'Apache'

```
alisa@linux:~$ sudo ufw allow 'Apache'
Rules updated
Rules updated (v6)
alisa@linux:~$
```

Правила обновлены и вы можете проверить изменение, набрав:

```
sudo ufw status
```

В результате, при включенном брандмауэре, будет предоставлен список разрешенного HTTPтрафика с уведомлением запись «Status: active» и мы смело переходим к главе **Проверка** вашего веб-сервера

```
alisa@linux:~$ sudo ufw status

Status: active

To Action From
-- -----
Apache ALLOW Anywhere
Apache (v6) ALLOW Anywhere (v6)

alisa@linux:~$
```

В случае же отключенного брандмауэра, мы получим запись «Status: inactive»

```
alisa@linux:~$ sudo ufw status
Status: inactive
```

Включение брандмауэра UFW в Ubuntu

Как только мы обнаружим, что брандмауэр не активен, проверим, добавлены ли к нему какиелибо правила. Эта команда будет работать, даже если брандмауэр неактивен.

sudo ufw show added

```
alisa@linux:~$ sudo ufw status
Status: inactive
alisa@linux:~$ sudo ufw show added
Added user rules (see 'ufw status' for running firewall):
ufw allow Apache
```

Проверим правила и убедимся, что если мы включим брандмауэр, с нашим сервером все будет в порядке. А затем включим брандмауэр с подтверждением наших действий клавишей **Y**.

sudo ufw enable

```
alisa@linux:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

Снова проверим статус брандмауэра UFW ранее использованной командой:

sudo ufw status

```
alisa@linux:~$ sudo ufw status
Status: active

To Action From
-- -----
Apache ALLOW Anywhere
Apache (v6) ALLOW Anywhere (v6)
alisa@linux:~$
```

Брандмауэр успешно активирован.

Проверка вашего веб-сервера

Наш веб-сервер уже должен быть установлен и запущен. Проверим **systemd** систему инициализации, чтобы убедиться, что служба работает, набрав:

sudo systemctl status apache2

```
alisa@linux:~$ sudo systemctl status apache2
[sudo] password for alisa:
 apache2.service - The Apache HTTP Server
    Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
    Active: active (running) since Wed 2024-02-28 13:40:18 UTC; 54min ago
      Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 2751 (apache2)
     Tasks: 55 (limit: 19032)
    Memory: 5.5M
       CPU: 204ms
    CGroup: /system.slice/apache2.service
              -2751 /usr/sbin/apache2 -k start
              -2752 /usr/sbin/apache2 -k start
              -2753 /usr/sbin/apache2 -k start
Feb 28 13:40:18 linux systemd[1]: Starting The Apache HTTP Server...
Feb 28 13:40:18 linux apachectl[2750]: AH00558: apache2: Could not reliably det>
Feb 28 13:40:18 linux systemd[l]: Started The Apache HTTP Server.
```

Как подтверждает этот вывод, служба запущена успешно. Однако лучший способ проверить это — запросить страницу у Apache.

Вы можете получить доступ к целевой странице Apache по умолчанию, чтобы убедиться, что

программное обеспечение работает правильно через ваш IP-адрес. Если вы не знаете IP-адрес вашего сервера, вы можете получить его несколькими способами из командной строки.

Попробуйте ввести это в командной строке вашего сервера:

hostname -I

```
alisa@linux:~$ hostname -I
192.168.1.145
alisa@linux:~$
```

Другой вариант — использовать инструмент Icanhazip, который должен предоставить вам ваш общедоступный IP-адрес, прочитанный из другого места в Интернете:

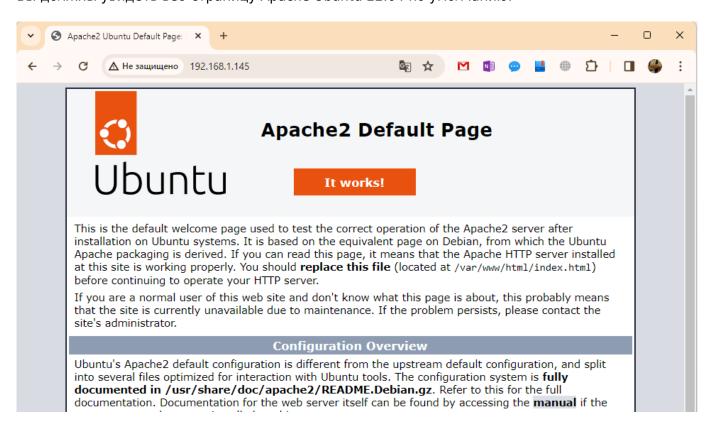
curl -4 icanhazip.com

```
alisa@linux:~$ hostname -I
192.168.1.145
alisa@linux:~$ curl -4 icanhazip.com
78.38.30.192
alisa@linux:~$
```

Когда у вас есть IP-адрес вашего сервера, введите его в адресную строку браузера (у меня это 192.168.1.145):

http://192.168.1.145

Вы должны увидеть веб-страницу Apache Ubuntu 22.04 по умолчанию:



Эта страница указывает на то, что Apache работает правильно. Он также включает некоторую

базовую информацию о важных файлах Apache и расположении каталогов.

Управление процессом Apache

Теперь, когда ваш веб-сервер настроен и работает, давайте рассмотрим некоторые основные команды управления с использованием **systemctl**.

Чтобы остановить ваш веб-сервер, введите:

sudo systemctl stop apache2

Чтобы запустить веб-сервер, когда он остановлен, введите:

sudo systemctl start apache2

Чтобы перезапустить веб-сервер, когда это необходимо, введите:

sudo systemctl restart apache2

Если вы просто вносите изменения в конфигурацию, Apache часто может перезагрузиться, не разрывая соединения. Для этого используйте эту команду:

sudo systemctl reload apache2

По умолчанию Apache настроен на автоматический запуск при загрузке сервера. Если это не то, что вам нужно, отключите это поведение, набрав:

sudo systemctl disable apache2

Чтобы снова включить запуск службы при загрузке, введите:

sudo systemctl enable apache2

Теперь Apache должен запускаться автоматически при повторной загрузке сервера.

Настройка виртуальных хостов (рекомендуется)

При использовании веб-сервера Apache вы можете использовать виртуальные хосты (аналогично серверным блокам в Nginx) для инкапсуляции деталей конфигурации и размещения более одного домена на одном сервере. Мы создадим домен под названием your_domain, но вам следует заменить его своим собственным доменным именем.

В Apache в Ubuntu 22.04 по умолчанию включен один серверный блок, который настроен для обслуживания документов из /var/www/html каталога. Хотя это хорошо работает для одного сайта, это может стать громоздким, если вы размещаете несколько сайтов. Вместо изменения /var/www/html, давайте создадим структуру каталогов /var/www для сайта your_domain, оставив ее /var/www/html в качестве каталога по умолчанию, который будет обслуживаться,

если запрос клиента не соответствует никаким другим сайтам.

Создайте каталог для your_domain следующим образом:

```
sudo mkdir /var/www/your_domain
```

Затем назначьте владельца каталога с помощью **\$USER** переменной среды:

```
sudo chown -R $USER:$USER /var/www/your_domain
```

Разрешения ваших веб-корней должны быть правильными, если вы не изменили значение umask, которое устанавливает разрешения для файлов по умолчанию. Чтобы убедиться, что ваши разрешения верны и разрешить владельцу читать, записывать и выполнять файлы, одновременно предоставляя разрешения только на чтение и выполнение группам и другим лицам, вы можете ввести следующую команду:

sudo chmod -R 755 /var/www/your_domain

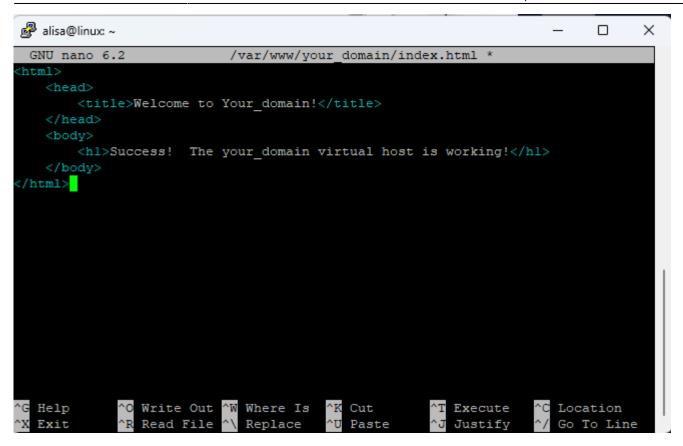
/var/www/				
Имя	Размер	Изменено	Права	Владел
t		28.02.2024 13:40:02	rwxr-xr-x	root
tml html		28.02.2024 13:40:14	rwxr-xr-x	root
your_domain_		28.02.2024 15:04:16	rwxr-xr-x	alisa

Затем создайте образец index.html страницы, используя nano или ваш любимый редактор:

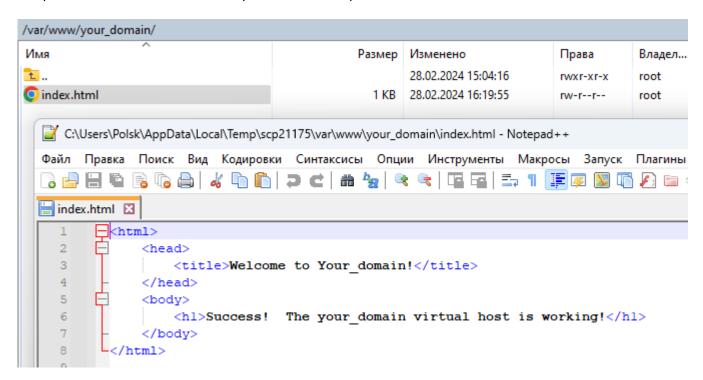
```
sudo nano /var/www/your_domain/index.html
```

Внутри добавьте следующий образец HTML:

index.html



Coxpaните CTRL+O→INTER и закройте CTRL+X файл, когда закончите.



Чтобы Apache мог обслуживать этот контент, необходимо создать файл виртуального хоста с правильными директивами. Вместо того, чтобы изменять файл конфигурации по умолчанию, расположенный по /etc/apache2/sites-available/000-default.conf адресу, давайте создадим новый по адресу:/etc/apache2/sites-available/your_domain.conf

```
sudo nano /etc/apache2/sites-available/your domain.conf
```

Вставьте следующий блок конфигурации, который аналогичен блоку по умолчанию, но

обновлен для нашего нового каталога и имени домена:

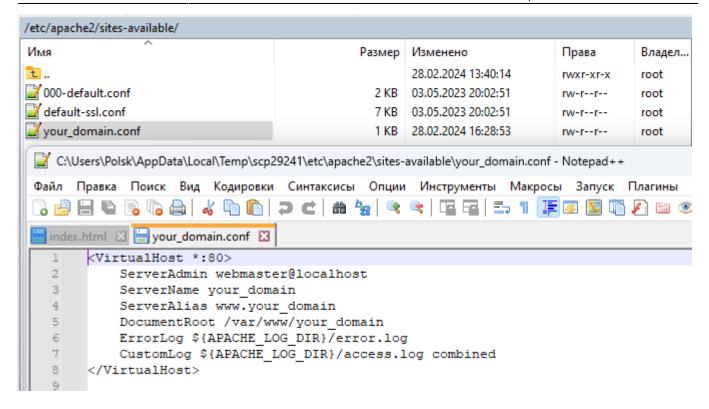
your_domain.conf

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName your_domain
    ServerAlias www.your_domain
    DocumentRoot /var/www/your_domain
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Обратите внимание, что мы обновили **DocumentRoot** наш новый каталог и **ServerAdmin** адрес электронной почты, к которому может получить доступ администратор сайта your_domain . Мы также добавили две директивы: **ServerName**, которая устанавливает базовый домен, который должен соответствовать этому определению виртуального хоста, и **ServerAlias**, которая определяет дополнительные имена, которые должны совпадать, как если бы они были базовым именем.

```
alisa@linux: ~
                                                                          ×
 GNU nano 6.2
                   /etc/apache2/sites-available/your domain.conf
VirtualHost *:80>
   ServerAdmin webmaster@localhost
   ServerName your domain
   ServerAlias www.your domain
   DocumentRoot /var/www/your domain
   ErrorLog ${APACHE LOG DIR}/error.log
   CustomLog ${APACHE LOG DIR}/access.log combined
/VirtualHost>
               Write Out
                            Where Is
                                          Cut
                                                       Execute
                                                                    Location
               Read File
                            Replace
```

Сохраните CTRL+O→INTER и закройте CTRL+X файл, когда закончите.



Давайте активируем файл с помощью a2ensite инструмента:

```
sudo a2ensite your domain.conf
```

```
alisa@linux:~$ sudo a2ensite your_domain.conf
Enabling site your_domain.
To activate the new configuration, you need to run:
   systemctl reload apache2
alisa@linux:~$
```

Отключите сайт по умолчанию, определенный в 000-default.conf:

```
sudo a2dissite 000-default.conf
```

```
alisa@linux:~$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
alisa@linux:~$
```

Далее проверим наличие ошибок конфигурации:

```
sudo apache2ctl configtest
```

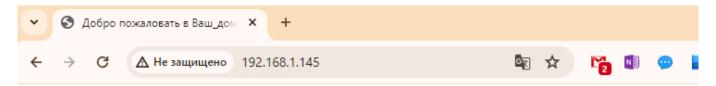
Вы должны получить следующий вывод:

```
alisa@linux:~$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified doma
in name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress th
is message
Syntax OK
alisa@linux:~$
```

Перезапустите Apache, чтобы изменения вступили в силу:

sudo systemctl restart apache2

Теперь Apache должен обслуживать ваше доменное имя. Вы можете проверить это, перейдя к , где вы должны увидеть что-то вроде этого: http://your_domain (http://192.168.1.145)



Успех! Виртуальный хост your_domain работает!

Знакомство с важными файлами и каталогами Apache

Теперь, когда вы знаете, как управлять самой службой Apache, вам следует потратить несколько минут на ознакомление с несколькими важными каталогами и файлами.

Содержание

• /var/www/html: Фактический веб-контент, который по умолчанию состоит только из страницы Apache по умолчанию, которую вы видели ранее, обслуживается из каталога /var/www/html. Это можно изменить, изменив файлы конфигурации Apache.

Конфигурация сервера

- /etc/apache2: каталог конфигурации Apache. Здесь находятся все файлы конфигурации Apache.
- /etc/apache2/apache2.conf: основной файл конфигурации Apache. Это можно изменить, чтобы внести изменения в глобальную конфигурацию Apache. Этот файл отвечает за загрузку многих других файлов в каталоге конфигурации.
- /etc/apache2/ports.conf: этот файл определяет порты, которые будет прослушивать Apache. По умолчанию Apache прослушивает порт 80 и дополнительно прослушивает порт 443, когда включен модуль, обеспечивающий возможности SSL.
- /etc/apache2/sites-available/: каталог, в котором могут храниться виртуальные хосты для каждого сайта. Арасhe не будет использовать файлы конфигурации, находящиеся в этом каталоге, если они не связаны с этим sites-enabled каталогом. Обычно вся конфигурация блоков сервера выполняется в этом каталоге, а затем включается путем связывания с другим каталогом с помощью команды a2ensite.
- /etc/apache2/sites-enabled/: каталог, в котором хранятся включенные виртуальные хосты для каждого сайта. Обычно они создаются путем ссылки на файлы конфигурации, находящиеся в sites-available каталоге с расширением a2ensite. Apache считывает файлы конфигурации и ссылки, найденные в этом каталоге, при запуске или перезагрузке для компиляции полной конфигурации.
- /etc/apache2/conf-available/, /etc/apache2/conf-enabled/: Эти каталоги имеют ту же связь, что и каталоги sites-available и sites-enabled, но используются для хранения

фрагментов конфигурации, которые не принадлежат виртуальному хосту. Файлы в **confavailable** каталоге можно включить с помощью **a2enconf** команды и отключить с помощью **a2disconf** команды.

• /etc/apache2/mods-available/, /etc/apache2/mods-enabled/: Эти каталоги содержат доступные и включенные модули соответственно. Файлы, заканчивающиеся на, .load содержат фрагменты для загрузки определенных модулей, а файлы, заканчивающиеся на, .conf содержат конфигурацию этих модулей. Модули можно включать и отключать с помощью команды a2enmod и a2dismod.

Журналы сервера

- /var/log/apache2/access.log: по умолчанию каждый запрос к вашему веб-серверу записывается в этот файл журнала, если Apache не настроен на иное.
- /var/log/apache2/error.log: По умолчанию все ошибки записываются в этот файл. Директива LogLevel в конфигурации Apache определяет, насколько подробно будут содержаться журналы ошибок.

Заключение

Теперь, когда у вас установлен веб-сервер, у вас есть множество вариантов типа контента, который вы можете обслуживать, и технологий, которые вы можете использовать для создания более богатого опыта.

Ссылки и Примечания

- Исправьте «sudo ufw status Status: inactive» в Ubuntu
- Первоначальная настройка сервера с Ubuntu 20.04
- Как установить веб-сервер Apache в Ubuntu 20.04

From

https://wwoss.ru/ - worldwide open-source software

Permanent link:

https://wwoss.ru/doku.php?id=software:linux server:ubuntu server install apache&rev=1709283744

Last update: 2024/03/01 12:02

