**федеральное государственное автономное образовательное учреждение высшего образования**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**(ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ)**

**(Факультет информационных технологий)**

***(Институт Принтмедиа и информационных технологий)***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 5**

**Дисциплина:** Операционные системы

**Тема: Установка и настройка LAMP-сервера**

**Выполнил: студент группы 231-338**

Шаура Илья Максимович

(Фамилия И.О.)

**Дата, подпись** 02.04.2024  ***\_\_\_\_\_\_\_\_\_\_\_***

**Проверил: *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание)

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва2024**

Лабораторная работа № 5.

1) Установка LAMP-сервера

LAMP — это стек веб-приложений, включающий в себя операционную

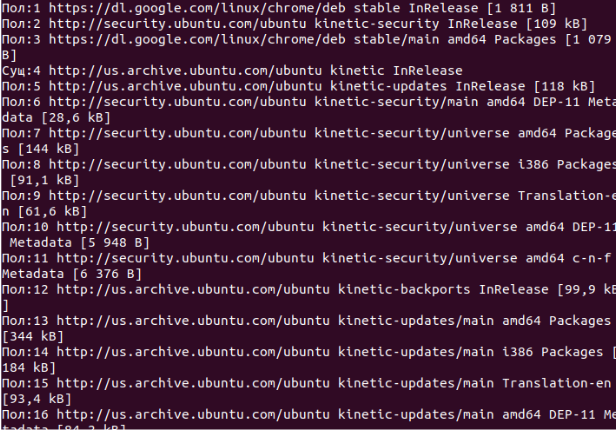
систему Linux, веб-сервер Apache, базу данных MySQL и язык

программирования PHP.

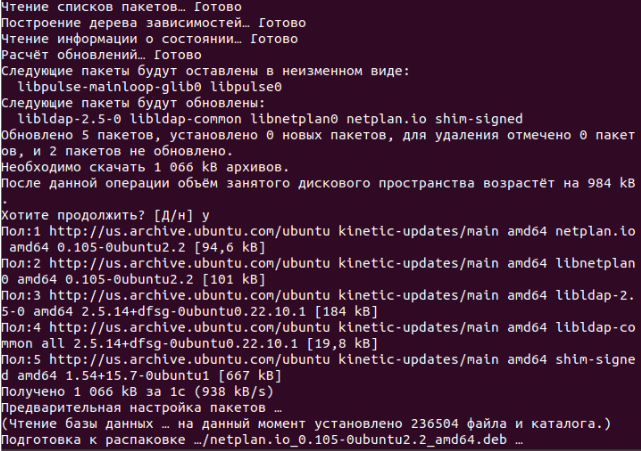
Обновление системы, необходимо для корректной установки компонентов

LAMP-server:

*sudo apt update*



*sudo apt upgrade*



2) Принципы работы и конфигурирования веб-сервера Apache

**Установка Apache**

Установим первый после операционной системы компонент LAMP — Apache.

Предварительно загрузим списки пакетов обновлений, для этого используется команда:

*sudo apt update*

Установим Apache:

*sudo apt install apache2*

**Настройка брандмауэра**

Брандмауэр (файервол) должен пропускать HTTP и HTTPS-трафик — то есть должны быть открыты порты 80 и 443.

Для начала нужно установить утилиту для управления правилами файервола UFW (Uncomplicated Firewall) и включить UFW:

*sudo apt install ufw*

*sudo ufw enable*

На сервере должны быть доступны профили приложений для Apache. Посмотрим список профилей:

*sudo ufw app list*

Вывод команды выглядит примерно таким образом:

*Available applications:*

*Apache*

*Apache Full*

*Apache Secure*

*OpenSSH*

Можем увидеть, что в выводе присутствует профиль Apache Full — именно он открывает нужные порты 80 и 443. Разрешим для него входящий трафик:

*sudo ufw allow &apos;Apache Full&apos;*

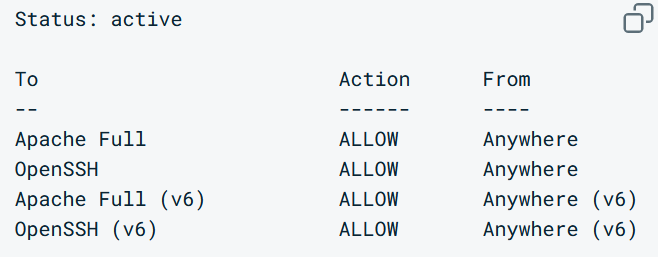
Чтобы в дальнейшем продолжать подключение к серверу по SSH, добавим также профиль OpenSSH:

*sudo ufw allow &apos;OpenSSH&apos;*

Проверим изменения:

*sudo ufw status*

В выводе команды проверим, что UFW активен, нужные профили включены, а, значит, разрешен HTTP и HTTPS-трафик:

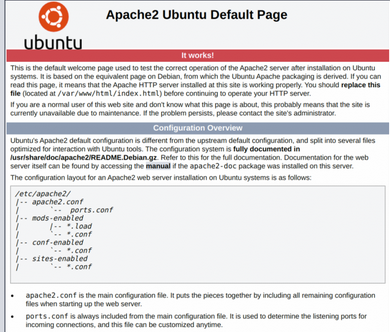


**Проверка работы Apache**

Проверим статус Apache, он должен быть активен (Active в выводе команды):

*sudo systemctl status apache2*

Чтобы проверить работу Apache, введем публичный IP-адрес сервера в браузерной строке. В нашем случае это подключенный ранее плавающий IP. Если Apache установлен, и к нему разрешен доступ через файервол, то откроется страница с информацией:



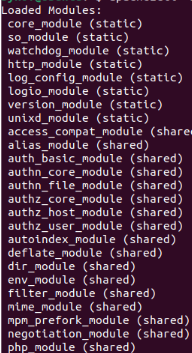
Система конфигурации Apache работает на текстовых файлах с прописанными настройками. Она подразделяется на три условных уровня, для каждого из которых имеется свой конфигурационный файл:

* Уровень конфигурации сервера (файл httpd.conf) – основной конфигурационный файл. Действие распространяется на весь механизм веб-сервера.
* Уровень каталога (файл .htaccess) – дополнительный конфигурационный файл. Его директивы охватывают только каталог, где расположен файл, а также вложенные подкаталоги.
* Уровень виртуального хоста (файл httpd.conf> или extra/httpd-vhosts.conf).
* Обычно конфигурационные файлы Apache находятся в папке «conf», а дополнительные конфигурационные файлы во вложенной в нее папке «extra». Внести изменения можно как через редактирование самого файла, так и через командную строку.

**Настройка модулей Apache**

Посмотреть все запущенные модули можно командой:

-apache2ctl –M



Включить модуль можно командой:

-sudo a2enmod имя\_модуля

А отключить:

sudo a2dismod имя\_модуля

После включения или отключения модулей нужно перезагрузить apache:

sudo systemctl restart apache2

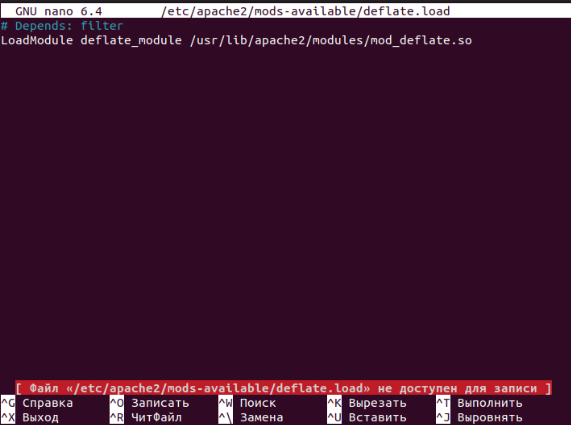
Во время выполнения одной из этих команд создается или удаляется

символическая ссылка на файл модуля с расширением load в директории

mods-available. Можем посмотреть содержимое этого файла, там только одна

строка. Например:

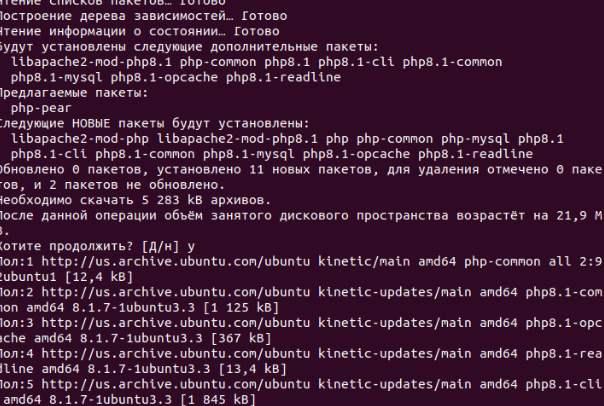
nano /etc/apache2/mods-available/deflate.load



3) Конфигурация PHP

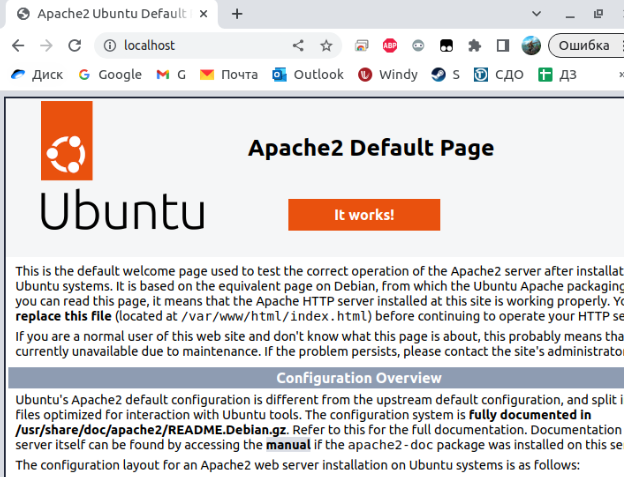
**Установка PHP**

*sudo apt install php libapache2-mod-php php-mysql*



Для проверки наберем в адресной строке браузера IP-адрес сервера, куда мы

устанавливали LAMP, в качестве примера выбран localhost:



То, что при открытии мы увидели страницу, означает что веб-север apache

успешно запущен и работает.

**Проверка, что PHP работает**

*sudo nano /var/www/html/info.php*

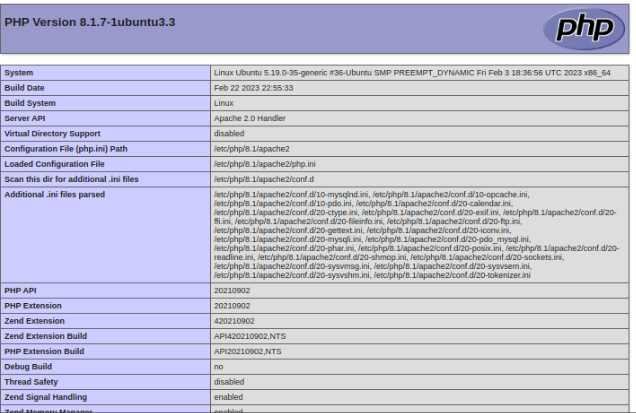
Откроется файл в текстовом редакторе. Напишите в нем следующее:

*<?php phpinfo(); ?>*

Сохраняем файл и закрываем редактор. Теперь откроем веб-браузер и

введите в адресной строке IP-адрес сервера, за которым следует /info.php:

<http://localhost/info.php>



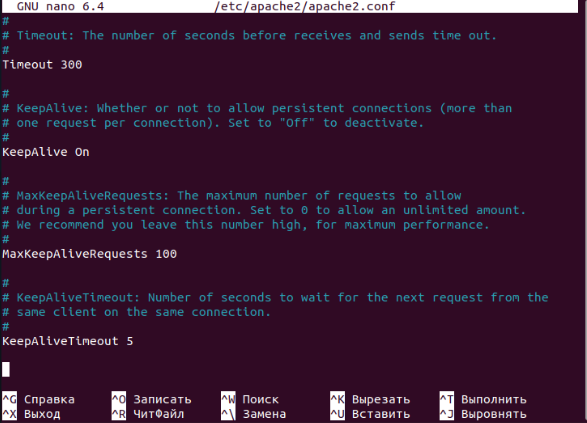
Так как мы видим информацию о PHP, значит PHP работает правильно.

Принципы работы и конфигурирования веб-сервера Apache

Сначала давайте рассмотрим главный файл конфигурации, для его открытия

используем команду:

*nano /etc/apache2/apache2.conf*



* Timeout - указывает как долго сервер будет пытаться продолжить прерванную передачу или прием данных.
* KeepAlive On - параметр, позволяет передавать несколько файлов, за одно соединение, например, не только саму html страницу, но и картинки и css файлы.
* MaxKeepAliveRequests 100 - максимальное количество запросов за одно соединение.
* KeepAliveTimeout 5 - таймаут соединения, обычно для загрузки страницы достаточно 5-10 секунд, так что больше ставить не нужно, но и рвать соединение раньше, чем загрузились все данные тоже не нужно.
* User, Group - пользователь и группа, от имени которых будет работать программа.
* HostnameLookups - записывать в логи вместо ip адресов доменные имена.
* LogLevel - уровень логирования ошибок. По умолчанию используется warn, но, чтобы логи заполнялись медленнее достаточно включить error
* Include - все директивы include отвечают за подключение рассмотренных выше конфигурационных файлов.

Параметр значение

</Directory>

Здесь доступны такие основные опции:

* AllowOverride - указывает нужно ли читать .htaccess файлы из этой директории, это такие же файлы настроек и таким же синтаксисом.
* All - разрешать все, None - не читать эти файлы.
* DocumentRoot - устанавливает из какой папки нужно брать документы для отображенияа пользователю
* Options - указывает какие особенности веб-сервера нужно разрешить в этой папке. Например, All - разрешить все, FollowSymLinks - переходить по символическим ссылкам, Indexes - отображать содержимое каталога если нет файла индекса.
* Require - устанавливает, какие пользователи имеют доступ к этому каталогу. Require all denied - всем запретить, Require all granted – всем разрешить. можно использовать вместо all директиву user или group чтобы явно указать пользователя.
* Order - позволяет управлять доступом к директории. Принимает два значения Allow,Deny - разрешить для всех, кроме указанных или Deny, Allow - запретить для всех, кроме указанных.

4) Принципы работы с виртуальными хостами

Виртуальный хост ― это сущность, которая позволяет работать множеству сайтов на одном сервере. Хосты совместно и максимально эффективно используют доступные ресурсы сервера.

Файлы сайтов хранятся на серверах в отдельных директориях. На одном сервере может быть размещено множество сайтов. Чтобы сервер понимал, какую директорию использовать, когда пользователь переходит на сайт, нужно связать домен и директорию между собой. За это отвечают виртуальные хосты. Они связывают доменное имя сайта с директорией, в которой находятся нужные файлы.

Настройка хостов при помощи связи домена с директорией называется Name-based (на основании имени). Есть и другие способы настройки виртуальных хостов. Например, Port-based (на основании порта) и IP-based (на основании IP-адреса).

Мы рассмотрим вариант настройки виртуального хоста на основании домена и покажем, как это сделать в Apache на сервере с Ubuntu 18.04 и 20.04.

**Настройка виртуальных хостов Apache**

Если на вашем сервере нет Apache, установите его. Для этого:

Обновите список пакетов:

*sudo apt update*

Установите Apache:

*sudo apt install apache2*

Готово, этого будет достаточно, чтобы настроить виртуальный хост и протестировать его работу при помощи HTML.