



LAMP

ЧТО ЭТО? И ДЛЯ ЧЕГО?

Что такое LAMP?

- ▶ стек LAMP - это набор программного обеспечения с открытым исходным кодом, который обычно устанавливается на сервер для отображения динамических веб-сайтов и веб-приложений. Эта аббревиатура обозначает операционную систему Linux с установленным веб-сервером Apache. Данные сайта хранятся в базе данных MySQL, динамический контент обрабатывается с помощью PHP.

АРАСНЕ

- ▶ Веб-сервер Apache в настоящее время является одним из самых популярных веб-серверов в мире. Он хорошо документирован и используется значительную часть времени с момента создания сети Интернет, что делает его прекрасным выбором для хостинга веб-сайта.

Установим Apache используя менеджер пакетов Ubuntu apt:

- ▶ `sudo apt update`
- ▶ `sudo apt install apache2`

Настройка файрвола для разрешения веб-трафика

- ▶ `sudo ufw`
- ▶ `sudo ufw app info "Apache Full"`

Проверка

Проверить результат установки можно набрав в вашем веб-браузере публичный IP адрес вашего сервера:

`http://IP_адрес_вашего_сервера`



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in [/usr/share/doc/apache2/README.Debian.gz](#)**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

Установка MySQL

- ▶ Теперь, когда наш веб-сервер установлен и запущен, пора установить MySQL. MySQL это система управления базами данных. Она организует и обеспечит доступ к базам данных, в которых ваш сайт может хранить информацию.

После завершения установки нам потребуется выполнить некоторые дополнительные команды, чтобы наше окружение MySQL было настроено безопасным образом. Введите следующую команду:

- ▶ `sudo mysql_secure_installation`

Установка PHP

- ▶ PHP это компонент, который будет обрабатывать код для отображения динамического контента. Он может запускать скрипты, подключаться к нашим базам данных MySQL для получения информации и передавать обработанный контент в наш веб-сервер для отображения.

Мы можем вновь воспользоваться менеджером пакетов apt для установки компонентов. Мы также добавим некоторые вспомогательные пакеты, чтобы код на PHP мог работать с нашим сервером Apache, а также обращаться к базе данных MySQL:

- ▶ `sudo apt install php libapache2-mod-php php-mysql`

Мы также можем проверить статус сервиса apache2 используя systemctl:

- ▶ `sudo systemctl restart apache2`
- ▶ `sudo systemctl status apache2`

Тестирование работы PHP на вашем веб-сервере

Чтобы проверить, что наша система сконфигурирована должным образом, мы можем создать простой *PHP* скрипт. Назовём этот скрипт `info.php`. Чтобы *Apache* нашел и обработал файл корректно, он должен быть сохранен в специальной директории, которая называется "*web root*".

В *Ubuntu 18.04* данная директория находится по адресу `/var/www/html/`. Мы можем создать там файл введя следующую команду:

- `sudo nano /var/www/html/info.php`

Проверка

Откроется пустой файл. Введём в файл следующий текст, который является корректным PHP кодом:

```
info.php
```

```
<?php phpinfo(); ?>
```

После внесения изменений сохраните и закройте файл.

Теперь мы можем проверить, может ли веб-сервер корректно отображать контент, сгенерированный PHP скриптом. Для проверки нам просто нужно открыть данную страницу в веб-браузере. Вам снова потребуется публичный IP-адрес сервера.

Откроем этот адрес:

```
http://IP_адрес_вашего_сервера/info.php
```