# Apache2 web-server setting up on the OS Debian

[Apache2 web-server setting up on the OS Debian 1](#_Toc477171264)

[Objective 2](#_Toc477171265)

[Progress 2](#_Toc477171266)

[Web server 2](#_Toc477171267)

[1. The Apache 2 Web Server and PHP 5 installations 2](#_Toc477171268)

[2. MySQL installation 4](#_Toc477171269)

[3. The connection of modules 5](#_Toc477171270)

[Checking the result 6](#_Toc477171271)

[4. The phpMyAdmin installation 7](#_Toc477171272)

[5. The PHP configuration 9](#_Toc477171273)

[6. Configure the Web server and virtual hosts 10](#_Toc477171274)

[Changing the local host 11](#_Toc477171275)

[The creation of the main page of the site 12](#_Toc477171276)

[A new virtual host creation 13](#_Toc477171277)

[7. FTP access setting up 14](#_Toc477171278)

[8. Useful console commands 16](#_Toc477171279)

[9. Extra information 17](#_Toc477171280)

[10. Starting a virtual machine 17](#_Toc477171281)

## Objective

Setting up a web server on the operating system, consisting of free open source software - Debian. Learn the basic commands of the Debian system required to configure the system. In our case, we set up the environment to simulate a real host. At the end of the Internet technology course, your website developed during the semester should be located on your virtual machine.

## Progress

During the lab, the web server will be configured on a virtual machine with the Debian operating system, which was created using VMPlayer. The instructions for starting the virtual machine are given in Section 10. You must log in to the operating system under the *student* user, you must use the super root user account to execute commands that require special privileges. Passwords for accounts are listed in section 9.

Web server

A web server is a server that accepts HTTP requests from clients, usually web browsers, and gives them HTTP responses, usually along with an HTML page, image, file, media stream or other data.

A Web server refers to both the software that performs the functions of the web server, and directly the computer on which this software is running.

The client, which is usually a web browser, sends requests to the web server for resources indicated by URLs. Resources are HTML pages, images, files, media streams or other data that the client needs. In response, the web server sends the requested data to the client.

This exchange takes place via the HTTP protocol.

Before you start installing all the components of the web server, you need to understand the principle of the web server and how the components interact with each other.

Let's consider some sequence of actions:

1. The user enters the address of the web page in the address bar of the browser;
2. The browser splits the address into components (page name and web server address) and sends the page name to the web server;
3. The web server (Apache 2) accepts the request to receive the page;
4. Apache reads the file from the hard disk of the server;
5. If the requested file is a PHP script, and not just an HTML file, then it is transferred to another process, the PHP interpreter.
6. The PHP interpreter executes the PHP code that it found in the text received from the web server process. This code also includes accessing the MySQL database.
7. The PHP interpreter asks the process of the MySQL database to process access to the database.
8. The MySQL database process returns query results to the database.
9. The PHP interpreter terminates execution of the PHP code, adding data retrieved from the database, and returns the process to the web server.
10. The web server returns the result to the browser as HTML text.
11. The web browser forms the appearance of the web page on the screen of our computer in accordance with the received HTML-text.

# The Apache 2 Web Server and PHP 5 installations

Before you start installing the software, you need to log in as a student, by running a virtual machine.

Apt-get (Advanced Packaging Tool) – is a program for installing, updating and removing software packages in a command mode. The program is able to automatically install and configure programs from both precompiled packages and source codes. Packages are taken from Internet repositories, or they can be installed from existing media. The list of package sources is stored in the /etc/apt/sources.list file. For users who prefer a graphical interface to the command line, there is Synaptic - a graphical shell for apt-get, created based on GTK +.

Using the Utility

apt-get [parameters] command

apt-get [options] install | remove package1 [package2 ...]

apt-get [parameters] source package1 [package2 ...]

Before installing the web server, you need to update the packages, for this, the following commands must be executed in the terminal:

update - is used to synchronize the package description files with their source and get updated lists of packages. Before upgrading and dist-upgrade, always perform an update.

upgrade - is used to install the latest versions of installed system packages from the sources specified in /etc/apt/sources.list. In order for apt-get to learn about the existence of available new versions of packages, this command must be preceded by the update command.

sudo apt-get update

sudo apt-get upgrade

After the packages have been updated, you can proceed with installing the web server. As the web server will act Apache. Installation is performed using the install command, followed by the names of the packages that you want to install.

As a programming language, the PHP scripting language, which will be installed along with the web server, will be used to run the lab, and the connection between the web server and PHP is done using the libapache2-mod-php5 library.

The install key - installs one or more packages marked for installation. Each package is the name of the package, not the full file name that contains this package. All dependency packages will also be received and installed.

sudo apt-get install apache2 apache2-doc php5 libapache2-mod-php5 php-pear



Figure 1 – Apache web-server installation

Attention! When installing the software, you may need a disk with the Debian operating system. The instructions for mounting the disk are given in section 7 of the current document.

Any data management system (DBMS) can be used to store data: MySQL, Sqlite, Oracle, PostgreSQL, MSSQL. Communication of the DBMS with PHP is carried out by connecting additional modules.

# MySQL installation

The implementation of laboratory work will be carried out using the MySQL database. MySQL is a free database management system and is suitable for use in the development of small and medium-sized applications. To install MySQL, run the following command:

sudo apt-get install mysql-server mysql-client php5-mysql



Figure 2 – MySQL installation

****

Figure 3 – MySQL setting up

# The connection of modules

The PHP programming language allows you to extend the set of standard classes and functions by installing additional modules. The modules are connected in the php.ini configuration file. This file can be opened in any text editor and uncommented the directives for connecting these or other modules, so that they look like this:

extension = module\_name

It is recommended not to connect all the modules at once, but do this later, as necessary. This approach will save RAM and make the server work more stable.

This command allows you to automatically download additional modules that may be useful in the future:

sudo apt-get install php5-curl php5-gd php5-idn php5-imagick php5-ldap php5-imap php5-memcache php5-mhash php5-mysql php5-pspell php5-sqlite php5-tidy imagemagick php5-xcache php5-xdebug php5-xmlrpc php5-xsl

You can find out about these modules on the Internet, in the course of the Internet technologies, the modules presented above will not be used.

## Checking the result

After everything has been installed, you can check the result, you need to know the IP address of your web server with the ifconfig command, then open the web browser and specify a URL that consists of the IP address: http: //192.168.126.130. You will see the standard Apache web server welcome message.

****

Figure 4 – A web-server successful running

# The phpMyAdmin installation

PHPMyAdmin is an open source web application developed in PHP and providing full-value, including remote, work with MySQL databases via a browser.

To install the Web application, you must use the following command:

sudo apt-get install phpmyadmin

During the installation, you must specify a Web server that will be automatically configured to work with phpMyAdmin. In our case, such a web server is Apache2. To select, press the spacebar.



Figure 5 – the phpMyAdmin setting up

The PhpMyAdmin uses the database to store data, which is necessary for its work, for correct work. Thus, let phpMyAdmin automatically configure the database:



Figure 6 – the phpMyAdmin creates database for successful work

As already mentioned, the PhpMyAdmin is a web application, the work of which is provided by a web server.

Additionally! In detail about the setting up the phpMyAdmin you can read on the official website: <http://php-myadmin.ru/doc/config.html>.

Next, you must run a set of commands to start the Web server:

sudo /etc/init.d/apache2 restart // restarts apache2

sudo a2enmod php5 // starts php

sudo a2enmod rewrite // module activation mod\_rewrite is a rules-based mechanism (parser with regular expressions) that executes the conversion URL on the fly.

Additionally! To find out what mod\_rewrite module is used for, visit: <http://httpd.apache.org/docs/current/mod/mod_rewrite.html>.

This completes the installation of phpMyAdmin in order to access the web application. It is necessary in the browser to open the page <http://192.168.120.128/phpmyadmin/>.

With this client, we will get acquainted in more detail when designing a database.



Figure 7 – the phpMyAdmin authorization

# The PHP configuration

The next step in configuring the web server will be the PHP configuration. To begin with it is necessary to be defined, where our sites will be stored. All sites will be located in the /var/www directory. To ensure security, you need to change the owner and group of users who will have access to this folder. To change the owner, use the *chown* command followed by the -R switch to recursively change the owner in the attached files and folders. Next is the username and group.

sudo chown -R student: student /var/www

Make sure that the user and group have been changed, for this, execute the command:

ls -l

This command displays the contents of the current directory.



Figure 8 – a directory listing with rights access

Now the very configuration of PHP, the configuration file with the name php.ini is stored in the directory /etc/php5/apache2/. You need to make the following changes to this file:

max\_execution\_time with 30 changes to 90

max\_input\_time change from 60 to 180

post\_max\_size change from 8M to 200M

upload\_max\_filesize change from 2M to 50M

max\_file\_uploads change from 50 to 150

# Configure the Web server and virtual hosts

Using virtual hosts allows you to create multiple Web sites on a single Web server.

There are two main methods for implementing access to websites:

• by name, when all websites share a common IP address. According to the HTTP/1.1 protocol, the web browser, when requested to the web server, specifies the domain name of the website in the Host field of the current request, and the web server uses it to properly execute the request, and copies this name to the [HTTP\_HOST] cell of the superglobal array $ \_SERVER.

• by IP address, where each website has its own IP address, and the web server has several physical or virtual network interfaces.

In our case, the first method of access to sites will be used for laboratory work. Since we have only one IP address. The most convenient way to host virtual hosts is to place them in the user's home directory.

## Changing the local host

Each site that will be located on the web server, you need to create a directory with the mkdir command. The first site has a local name localhost. A good tone when creating a virtual host, inside the folder with the host name in our case localhost, create a www folder in which the script files of the site will be located.

cd /var/www/

mkdir localhost

cd /var/www/localhost

mkdir www

The configuration of each virtual host is stored in the directory: /etc/apache2/sites-available/. Let's make the site that is accessible at http://localhost, physically located in the directory /var/www/localhost/www.

To do this, you need to make changes to the configuration file of the standard virtual host, which is responsible for the site with the domain name localhost. Red color indicates the place of change. Thus, we specified that the root directory for this host will be located /var/www/localhost/www.

sudo nano /etc/apache2/sites-available/default

*<VirtualHost \*: 80>*

 *# The ServerName directive sets the request scheme, hostname and port t $*

 *# The server uses to identify itself. This is used when creating*

 *# Redirection URLs. In the context of virtual hosts, the ServerName*

 *# Specifies what hostname must appear in the request's Host: header to*

 *# Match this virtual host. For the default virtual host (this file) this*

 *# Value is not decisive as it is used.*

 *# However, you must set it for any further virtual host explicitly.*

 *#ServerName www.example.com*

 *ServerAdmin webmaster@localhost*

 *DocumentRoot /var/www/localhost/www*

 *# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,*

 *# error, crit, alert, emerg.*

 *# It is also possible to configure the loglevel for particular*

 *# modules, e.g.*

 *# LogLevel info ssl: warn*

 *ErrorLog $ {APACHE\_LOG\_DIR} /error.log*

 *CustomLog $ {APACHE\_LOG\_DIR} /access.log combined*

 *# For most configuration files from conf-available/, which are*

 *# enabled or disabled at a global level, it is possible to*

 *# include a line for only one particular virtual host. For example the*

 *# the following line enables the CGI configuration for this host only*

 *# after it has been globally disabled with "a2disconf".*

 *# Include conf-available/serve-cgi-bin.conf*

*</ VirtualHost>*

*# Vim: syntax = apache ts = 4 sw = 4 sts = 4 sr noet*

After the changes have been made, you must restart the apache web-server.

## The creation of the main page of the site

Now we turn to the development of the main page of the site, it will consist of only one line and display the current server configuration. Open the non-existing file named index.php in the editor, it will automatically be created after saving

nano /var/www/localhost/www/index.php

In the opened file we will write a script in which the function phpinfo () is called:

<? php phpinfo (); ?>

All scripts developed in the php language are placed in the <? php ?> Tags.

To view the result of the script, open the page <http://192.168.120.128/>. Physically, the file is located in the /var/www/localhost/ www directory.

## A new virtual host creation

To create another virtual host, you must perform the above actions. But for each new host, you need to create your own file with a unique name (describing the domain name, for example, labs) and the settings in the /etc/apache2/sites-available/ directory. You can copy the default settings file and make changes to it:

sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/labs.conf

As already mentioned above, the website files should be placed in the user's home directory, who will upload the files to the server (in our case it's the student user). In order to go to the home directory of the current user (in your case it's the student user), you need to use the cd command.

Register in the settings, then the domain name, by which it will be accessible from the browser.

*...*

*ServerAdmin webmaster @ localhost*

 *DocumentRoot /home/student/labs/www # do not forget to create directories*

*ServerName labs.loc*

*...*

*<Directory /home/student/labs/www>*

 *...*

 *Require all granted # you must grant full access to the directory*

 *...*

*</ Directory>*

*...*

Important! You must grant access rights for all users so that the web server can access the /home/student/labs/ www website using the command:

sudo chmod 777 / home / student / labs -R

Creating a configuration file is not enough for the virtual host to work, you need to activate it with the command:

sudo a2ensite labs

Then you must restart the Apache2.

Now we need to map the IP address created by us to the virtual host so that the browser knows where to look for the site requested by the user. To do this, you need to edit the hosts file. Hosts - a text file containing a database of domain names and used when translating them into network addresses of nodes. The request to this file takes precedence before it accesses DNS servers. Unlike DNS, the contents of the file are controlled by the computer administrator.

Copy the file C:\Windows\System32\drivers\etc\ host, to the desktop of the computer, edit the copied file, adding the following entry:

192.168.126.130 labs.loc

Then replace the original file with a modifiable one. Attention! Access to this file can be obtained only by the administrator, ask the teacher to help.



Figure 9 – The hosts file editing

Save, close, and restart the Apache web-server. After the virtual host configuration is complete, provide a proof of a successful start.

# FTP access setting up

In the previous 6 paragraphs, a full-fledged web server was configured, but what is the web server without remote file management? The fastest and easiest way is to configure the ftp server. FTP (File Transfer Protocol) is a standard protocol intended for transferring files over TCP networks (for example, the Internet). There is a large number of ftp-servers. In our labs we will configure the vsftpd web server.

apt-get install vsftpd

In fact, the installation process does not end with the execution of one command, but for you to do the lab work it will be enough.

Next, you need to make changes to the configuration file:

sudo nano /etc/vsftpd.conf

Ensure that the write\_enable directive is set to YES. Otherwise, you will not be able to copy the files to the server. Do not forget to restart the FTP service.

****

Figure 10 – FTP server installation process

You can use Windows Explorer as a client. The authentication process is shown below.

****

Figure 11 – FTP server authentication window

# Useful console commands

sudo /etc/init.d/apache2 reload – apache web-server reload

sudo — super user (root) actions

sudo -i – runs shell with super user (root) rights

nano /path/to/file — opens file in console editor nano

gedit /path/to/file — opens file in открыть файл в текстовом редакторе Gedit

cp /path/to/file1 /path/to/file2 — copying a file1 to file2

ln -s /path/to/file /path/to/new\_link — symbol link

wget http://site.dom/archive.tar.gz — download archive .tar.gz

aptitude install 'package' — install package ‘packgage’

apt-get install 'package' — install package ‘packgage’

aptitude -t lenny-backports install 'package' — install package ‘packgage’ from lenny-backports repository

cd — change directory

mkdir — make directory

rm — delete directory

mv — rename

tar — operations with archives

chmod — change file access

chown — change file owner

sudo a2ensite virtual\_host\_name – activate virtual host

sudo a2dissite virtual\_host\_name – deactivate virtual host

# Extra information

**User to use the system:**

User Name: student

Password: student

**Computer administrator:**

Username: root

Password: root

# Starting a virtual machine

To start a virtual machine, click on the shortcut (figure 13) located on the desktop or in the program menu.

****

Figure 13 – VMWare Player shortcut

Next, you need to click "Open a Virtual Machine" and select the file containing the Debian 6 virtual machine 64-bit.vmx located on the E: \\!InternetTechnology\ItVM drive of the current machine in the dialog box. If the file does not exist in the specified directory, you need to copy the virtual machine from <https://yadi.sk/d/_9-gnFh4tXqAu>.