Name : Adwait S Purao

UID: 2021300101

Batch: B2

Experiment no.: 6

Aim: To Create a Web Server using Apache2

Theory:

Web Server:

- A web server is software and hardware that uses HTTP (Hypertext Transfer Protocol) and other protocols to respond to client requests made over the World Wide Web. The main job of a web server is to display website content through storing, processing, and delivering webpages to users. Besides HTTP, web servers also support SMTP (Simple Mail Transfer Protocol) and FTP (File Transfer Protocol), used for email, file transfer and storage.
- Web server hardware is connected to the internet and allows data to be exchanged with other connected devices, while web server software controls how a user accesses hosted files. The web server process is an example of the client/server model. All computers that host websites must have web server software.
- Web servers are used in web hosting, or the hosting of data for websites and web-based applications -- or web applications.

How do web servers work?

- Web server software is accessed through the domain names of websites and ensures the delivery of the site's content to the requesting user.
- The software side is also comprised of several components, with at least an HTTP server. The HTTP server can understand HTTP and URLs.
- As hardware, a web server is a computer that stores web server software and other files related to a website, such as HTML documents, images, and JavaScript files.
- When a web browser, like Google Chrome or Firefox, needs a file that is hosted on a web server, the browser will request the file by HTTP. When the request is received by the web server, the HTTP server will accept the request, find the content, and send it back to the browser through HTTP.

- More specifically, when a browser requests a page from a web server, the process will follow a series of steps. First, a person will specify a URL in a web browser's address bar. The web browser will then obtain the IP address of the domain name -- either translating the URL through DNS (Domain Name System) or by searching in its cache. This will bring the browser to a web server. The browser will then request the specific file from the web server by an HTTP request. The web server will respond, sending the browser the requested page, again, through HTTP. If the requested page does not exist or if something goes wrong, the web server will respond with an error message. The browser will then be able to display the webpage.

Common and top web server software on the market
There are several common web servers available, some including:

- Apache HTTP Server
- Microsoft Internet Information Services (IIS)
- Nginx
- Lighttpd
- Sun Java System Web Server

Web Server Configuration:

- Web server configuration refers to the process of setting up and customizing a web server to meet the needs of a particular website or application. This involves configuring various settings, such as network protocols, security settings, server-side scripting languages, database connections, and caching rules.

- The configuration process is critical for ensuring that a web server is optimized for performance, security, and scalability. It typically involves modifying configuration files, such as the Apache httpd.conf file or the Nginx configuration file, to specify how the server should handle requests and respond to clients.

- Some common configuration tasks include:

o Setting up virtual hosts to handle multiple websites or domains
o Configuring SSL certificates for secure HTTPS connections
o Enabling server-side scripting languages like PHP or Python
o Configuring caching rules to improve performance
o Configuring access control rules to restrict access to sensitive resources
o Configuring error pages and logging settings for troubleshooting and monitoring
o Web server configuration can be a complex and time-consuming process, but it is essential for ensuring that a website or application is secure, reliable, and performant.

Apache:

- Apache is a free and open-source web server software that is widely used for serving web pages over the Internet.

- It was first released in 1995 and has since become one of the most popular web server software packages available.
- Apache is maintained by the Apache Software Foundation, a non-profit organization that supports the development of open-source software.

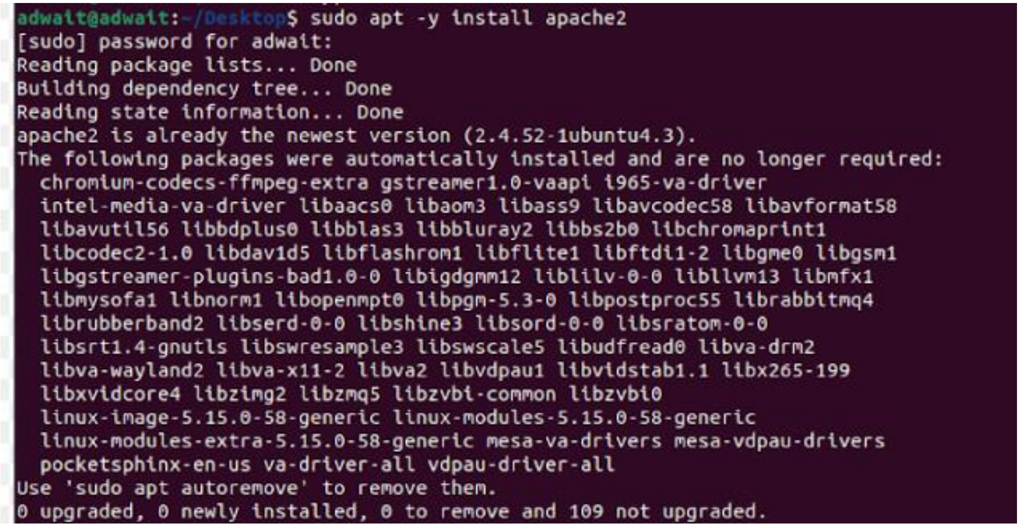
- Apache is known for its stability, security, and flexibility. It can run on a wide variety of operating systems, including Windows, Linux, and macOS. Apache supports a variety of programming languages, such as PHP, Python, and Perl, and can be extended with modules to add additional functionality.

- Apache uses a modular architecture, allowing administrators to enable or disable features as needed, which can improve performance and security. It is often used in conjunction with other open-source software packages, such as the MySQL database server and the PHP scripting language, to create a complete web development stack known as LAMP (Linux, Apache, MySQL, PHP).

- Apache is licensed under the Apache License, which allows it to be used and distributed freely, including for commercial purposes. It has a large and active community of users and developers who contribute to its ongoing development and support.

Screenshots:

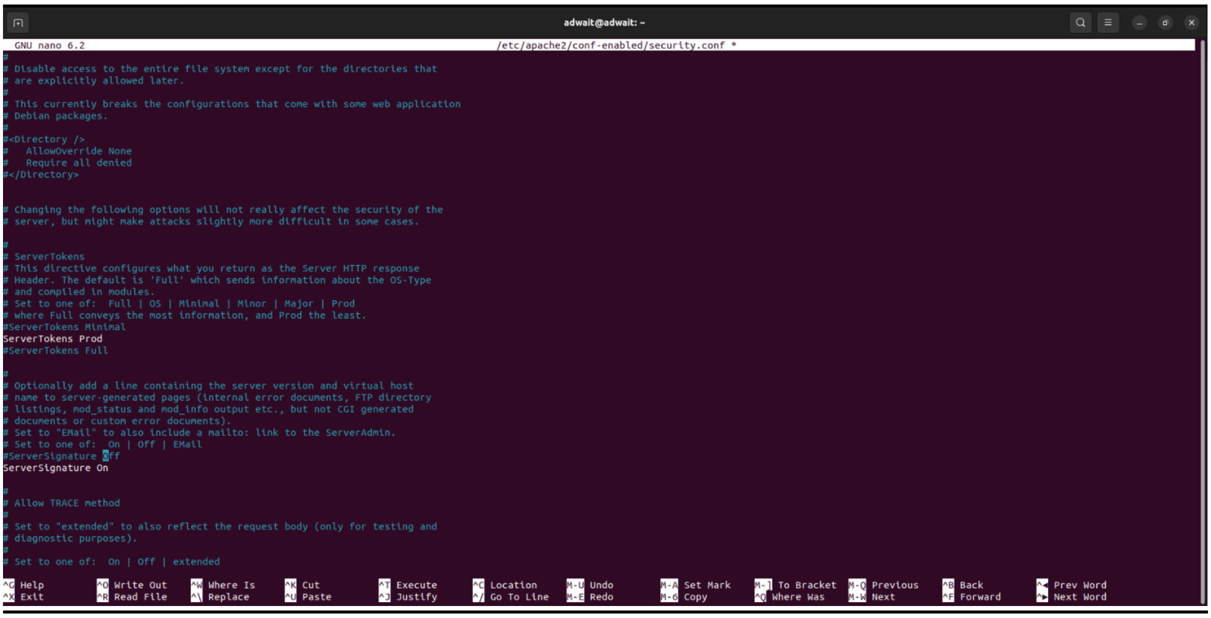
1. Install Apache:



```
adwait@adwait:~/Desktop$ sudo apt -y install apache2
[sudo] password for adwait:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.52-1ubuntu4.3).
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver
  intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58
  libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0 libchromaprint1
  libcodec2-1.0 libdav1d5 libflashrom1 libflite1 libftdi1-2 libgme0 libgsm1
  libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libllvm13 libmfx1
  libmysofa1 libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55 librabbitmq4
  librubberband2 libserd-0-0 libshine3 libsord-0-0 libsratom-0-0
  libsrt1.4-gnutls libswresample3 libswscale5 libudfread0 libva-drm2
  libva-wayland2 libva-x11-2 libva2 libvdpau1 libvidstab1.1 libx265-199
  libxvidcore4 libzimg2 libzmq5 libzvbi-common libzvbi0
  linux-image-5.15.0-58-generic linux-modules-5.15.0-58-generic
  linux-modules-extra-5.15.0-58-generic mesa-va-drivers mesa-vdpau-drivers
  pocketsphinx-en-us va-driver-all vdpau-driver-all
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 109 not upgraded.
```

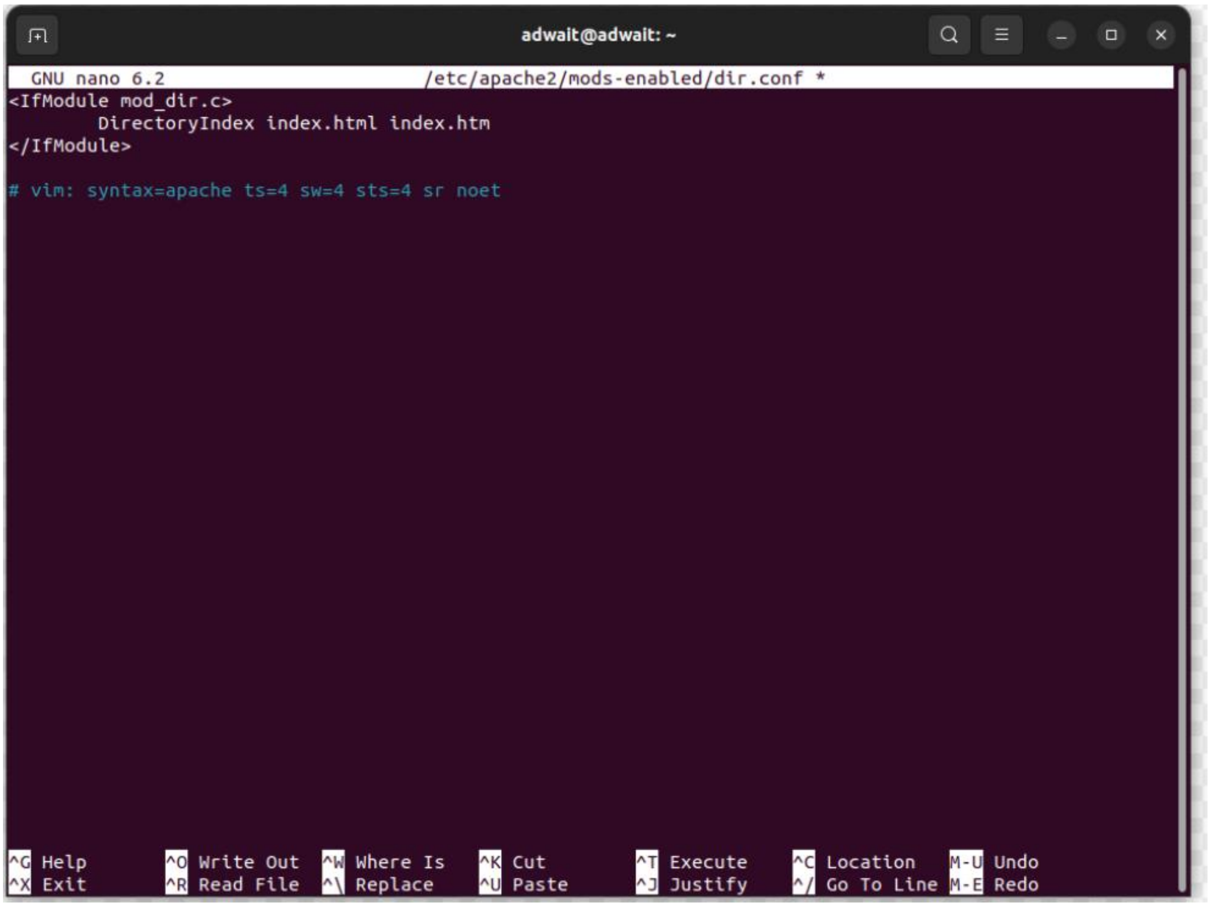
2. Configuring Apache:

- Changing Server Tokens from OS to Prod



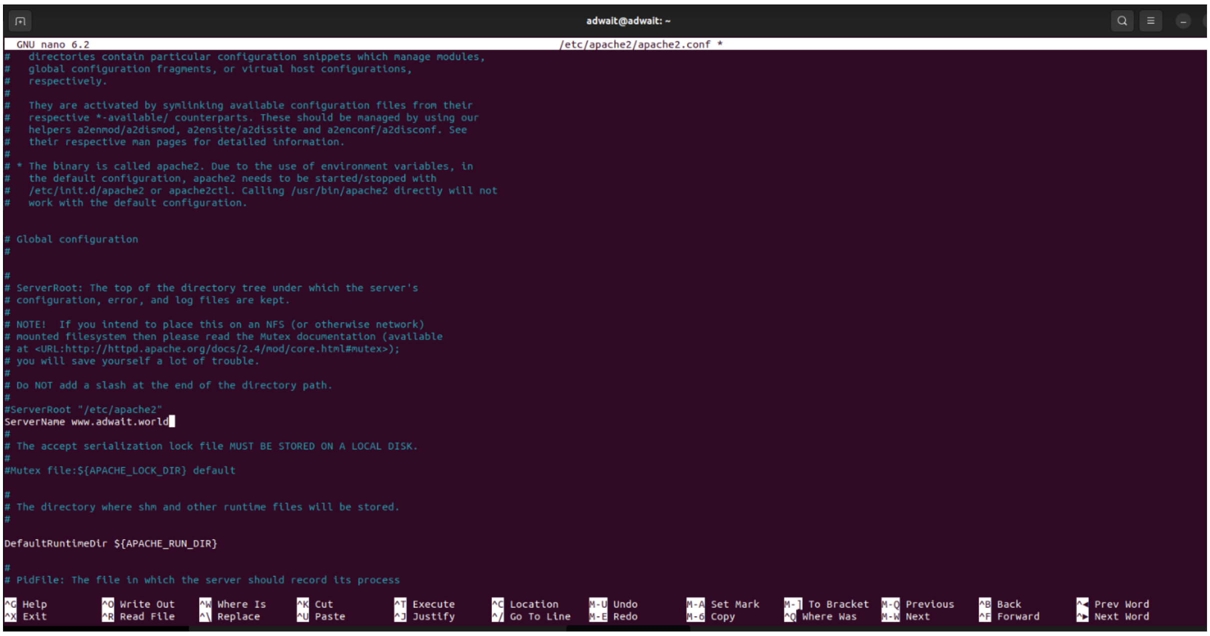
```
#
# Disable access to the entire file system except for the directories that
# are explicitly allowed later.
#
# This currently breaks the configurations that come with some web application
# Debian packages.
#
#<Directory />
#   AllowOverride None
#   Require all denied
#</Directory>


# Changing the following options will not really affect the security of the
# server, but might make attacks slightly more difficult in some cases.

#
# ServerTokens
# This directive configures what you return as the Server HTTP response
# Header. The default is 'Full' which sends information about the OS-Type
# and compiled in modules.
# Set to one of:  Full | OS | Minimal | Minor | Major | Prod
# where Full conveys the most information, and Prod the least.
#ServerTokens Minimal
ServerTokens Prod
#ServerTokens Full

#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (internal error documents, FTP directory
# listings, mod_status and mod_info output etc., but not CGI generated
# documents or custom error documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of:  On | Off | EMail
#ServerSignature Off
ServerSignature On

#
# Allow TRACE method
#
# Set to "extended" to also reflect the request body (only for testing and
# diagnostic purposes).
#
# Set to one of:  On | Off | extended
```

- add file name that it can access only with directory's name



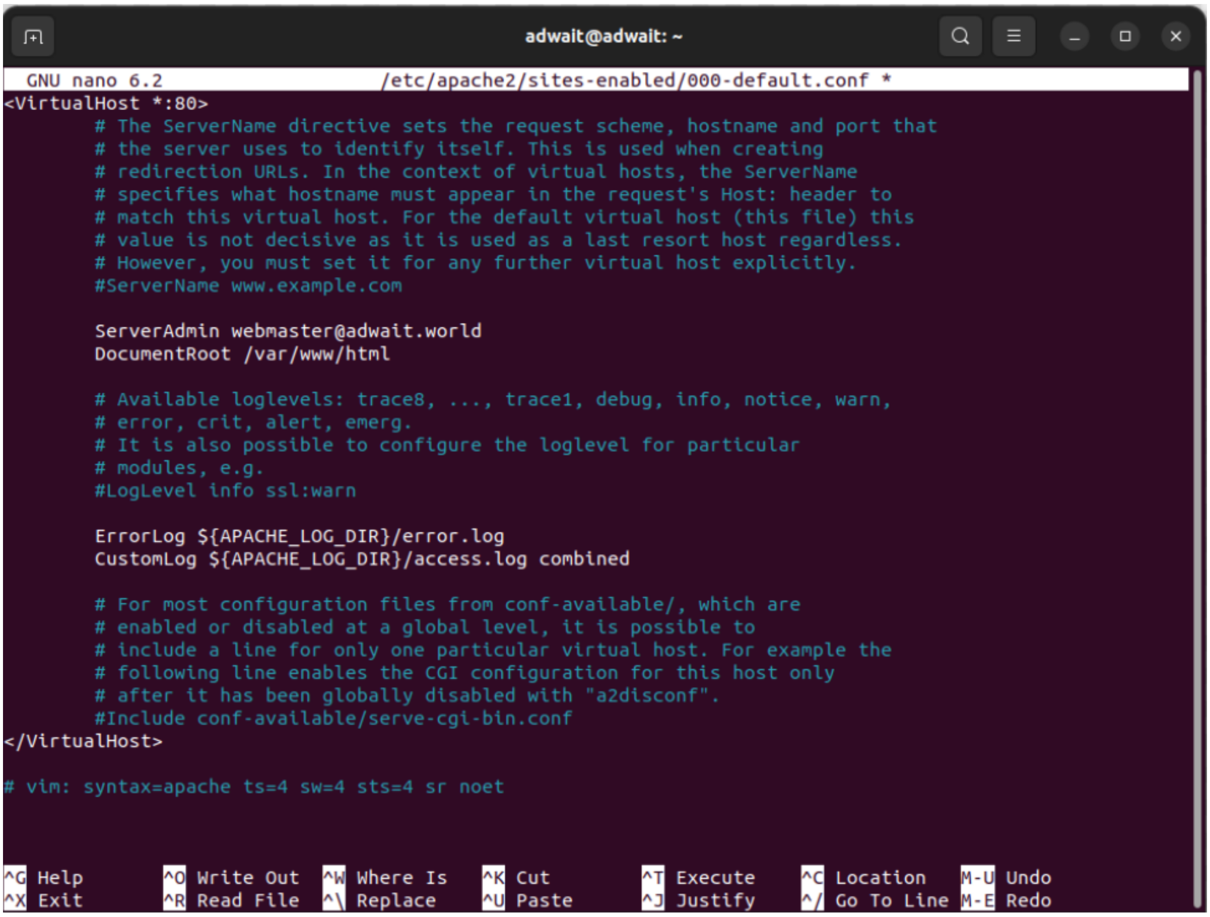
```
<IfModule mod_dir.c>
        DirectoryIndex index.html index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

- add to specify server name



```
  GNU nano 6.2                                          /etc/apache2/apache2.conf *
#   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.


# Global configuration
#


#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE!  If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the Mutex documentation (available
# at <URL:http://httpd.apache.org/docs/2.4/mod/core.html#mutex>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
#ServerRoot "/etc/apache2"
ServerName www.adwait.world
#
# The accept serialization lock file MUST BE STORED ON A LOCAL DISK.
#
#Mutex file:${APACHE_LOCK_DIR} default

#
# The directory where shm and other runtime files will be stored.
#

DefaultRuntimeDir ${APACHE_RUN_DIR}

#
# PidFile: The file in which the server should record its process

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo    M-A Set Mark  M-] To Bracket M-Q Previous  ^B Back      ^◂ Prev Word
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line M-E Redo   M-6 Copy      ^Q Where Was   M-W Next      ^F Forward   ^▸ Next Word
```

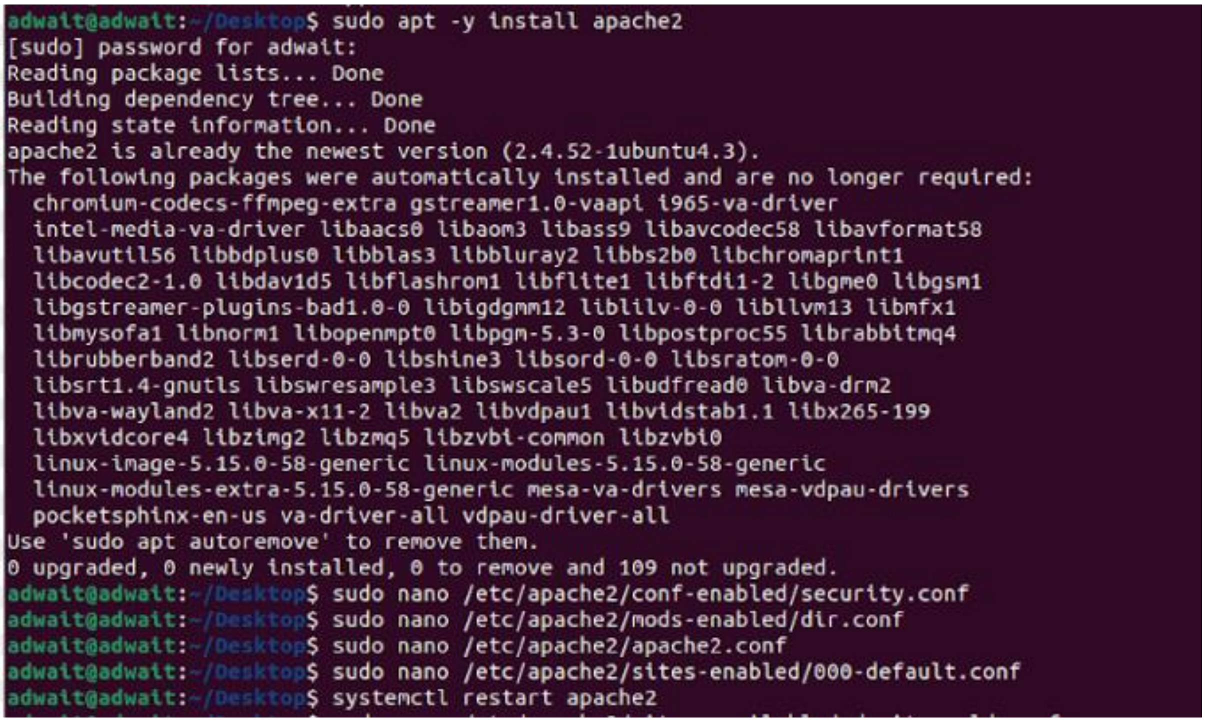
- change to webmaster's email



```
  GNU nano 6.2                  /etc/apache2/sites-enabled/000-default.conf *
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # 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 as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@adwait.world
        DocumentRoot /var/www/html

        # 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
        # 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



^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo
```

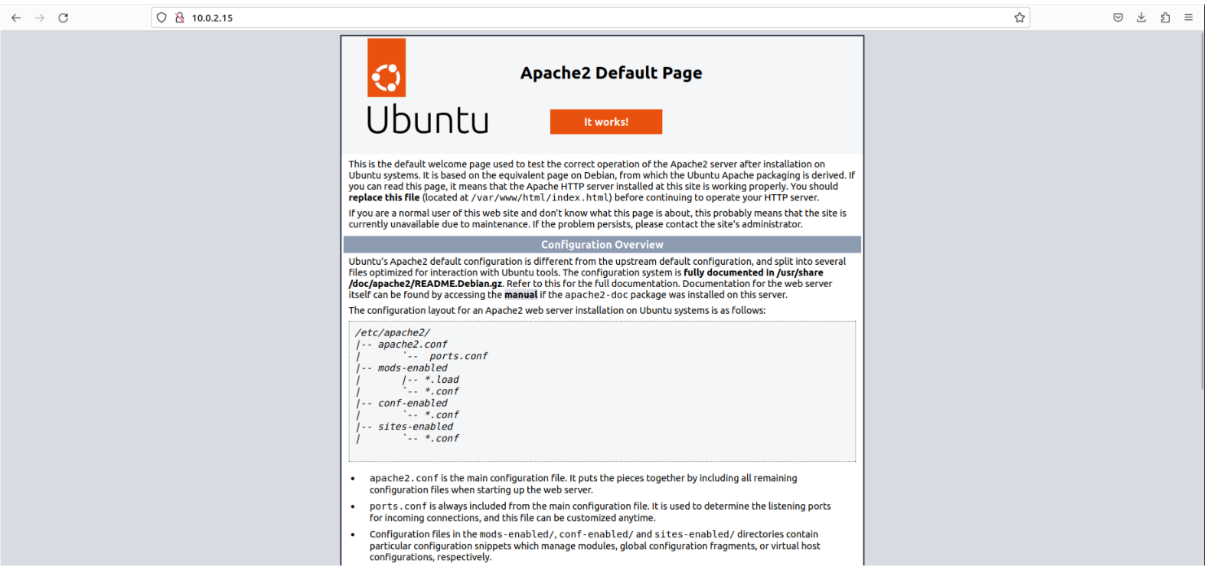
- Restart the Apache Server

Commands Done till this step



```
adwait@adwait:~/Desktop$ sudo apt -y install apache2
[sudo] password for adwait:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.52-1ubuntu4.3).
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver
  intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58
  libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0 libchromaprint1
  libcodec2-1.0 libdav1d5 libflashrom1 libflite1 libftdi1-2 libgme0 libgsm1
  libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libllvm13 libmfx1
  libmysofa1 libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55 librabbitmq4
  librubberband2 libserd-0-0 libshine3 libsord-0-0 libsratom-0-0
  libsrt1.4-gnutls libswresample3 libswscale5 libudfread0 libva-drm2
  libva-wayland2 libva-x11-2 libva2 libvdpau1 libvidstab1.1 libx265-199
  libxvidcore4 libzimg2 libzmq5 libzvbi-common libzvbi0
  linux-image-5.15.0-58-generic linux-modules-5.15.0-58-generic
  linux-modules-extra-5.15.0-58-generic mesa-va-drivers mesa-vdpau-drivers
  pocketsphinx-en-us va-driver-all vdpau-driver-all
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 109 not upgraded.
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/conf-enabled/security.conf
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/mods-enabled/dir.conf
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/apache2.conf
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/sites-enabled/000-default.conf
adwait@adwait:~/Desktop$ systemctl restart apache2
```

Output:



Apache2 Default Page

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:

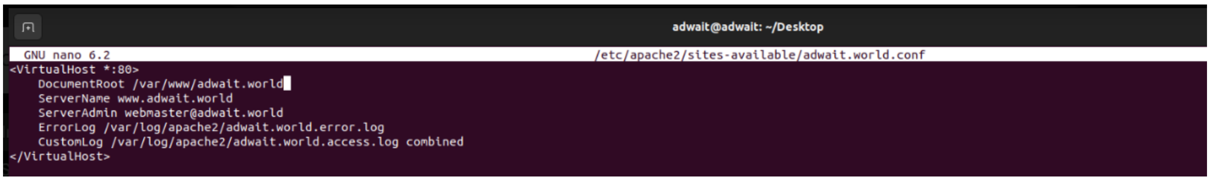
```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `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.

3. Configuring Virtual Host: Configure Virtual Hostings to use multiple domain names.

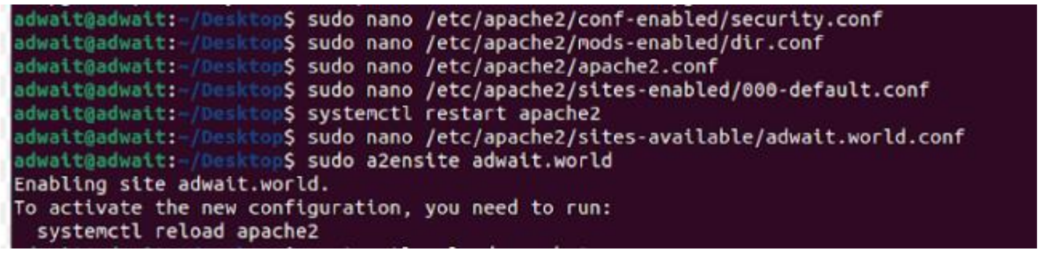
For example, Add a new Host setting that domain name is [virtual.host], document root is [/var/www/virtual.host].

- create new settings for new domain



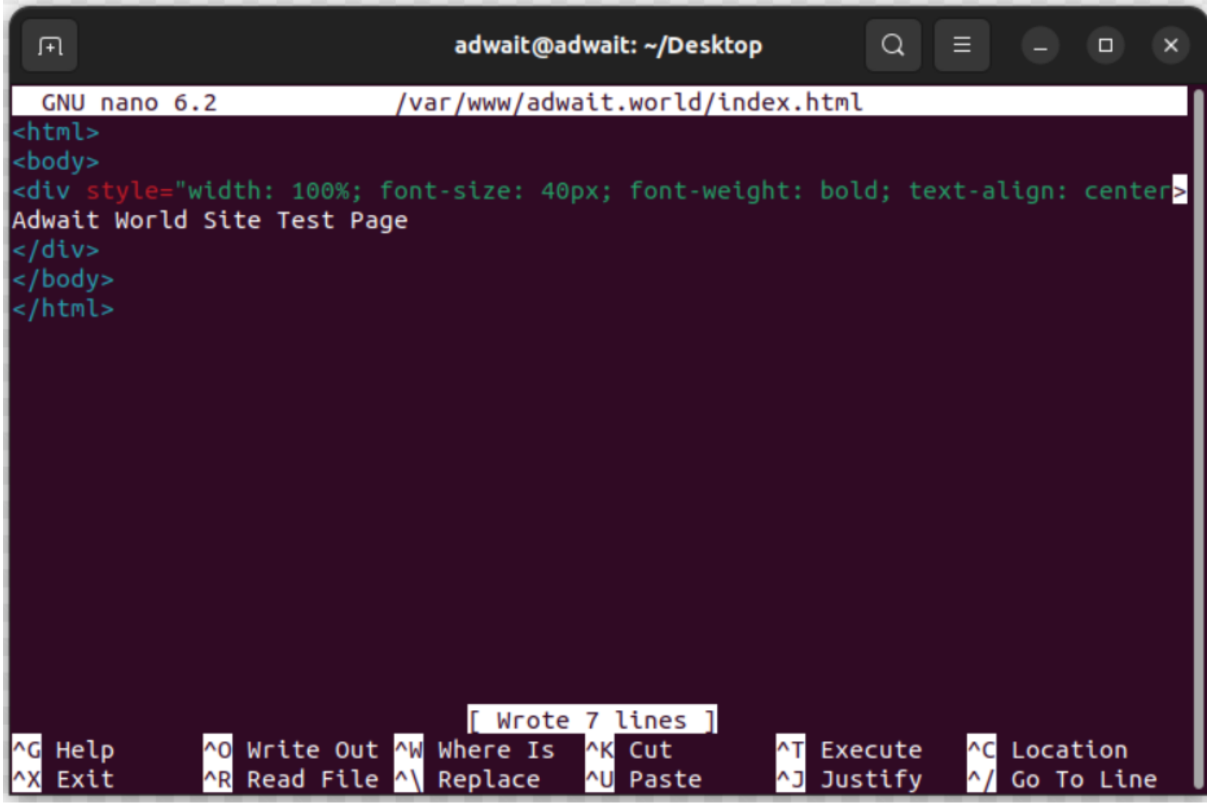
```
  GNU nano 6.2                                  /etc/apache2/sites-available/adwait.world.conf
<VirtualHost *:80>
    DocumentRoot /var/www/adwait.world
    ServerName www.adwait.world
    ServerAdmin webmaster@adwait.world
    ErrorLog /var/log/apache2/adwait.world.error.log
    CustomLog /var/log/apache2/adwait.world.access.log combined
</VirtualHost>
```

- Giving new configuration



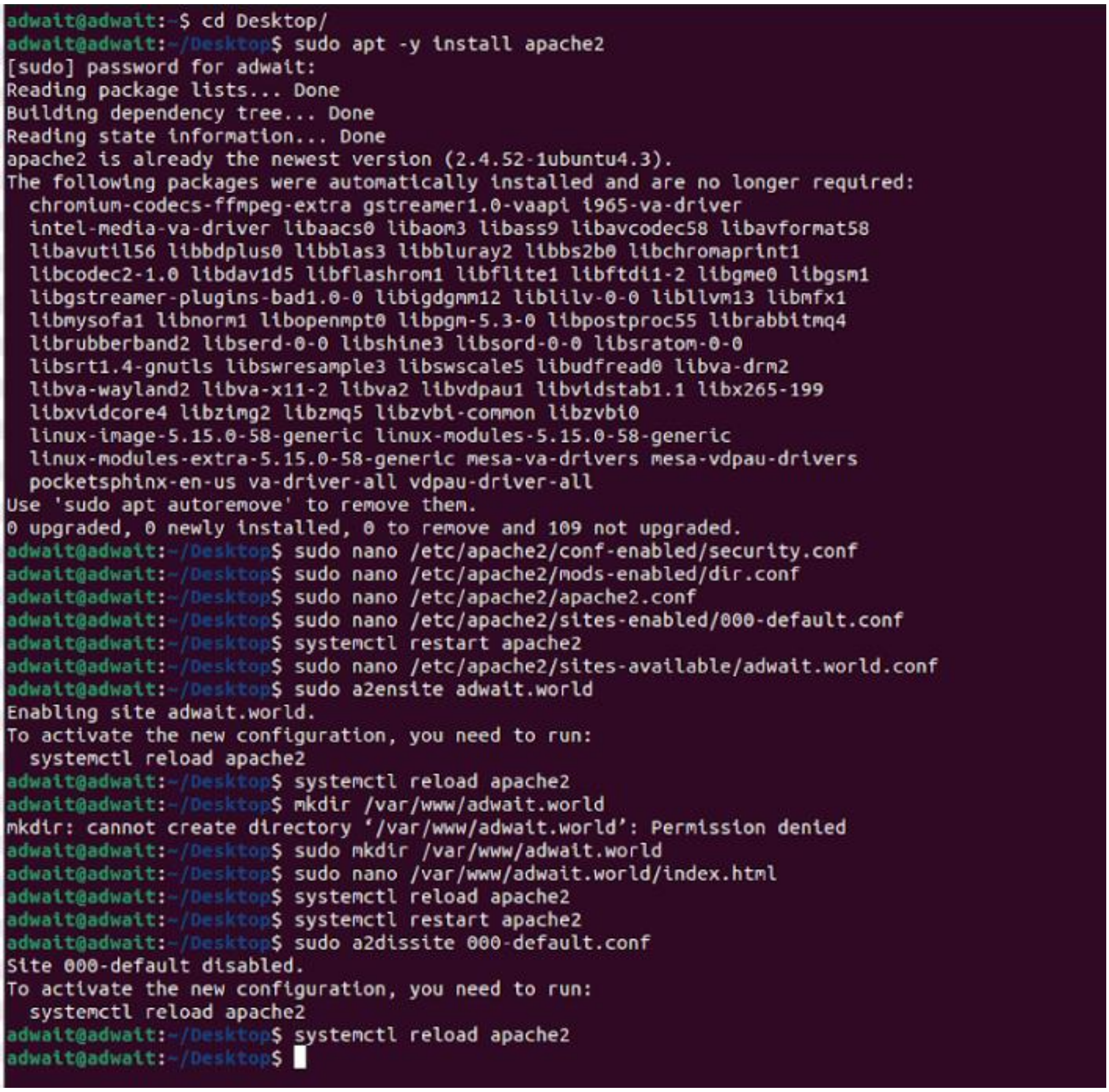
```
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/conf-enabled/security.conf
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/mods-enabled/dir.conf
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/apache2.conf
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/sites-enabled/000-default.conf
adwait@adwait:~/Desktop$ systemctl restart apache2
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/sites-available/adwait.world.conf
adwait@adwait:~/Desktop$ sudo a2ensite adwait.world
Enabling site adwait.world.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

4. Create a test page and access to it from any client computer with web browser. That is OK if following page is shown.



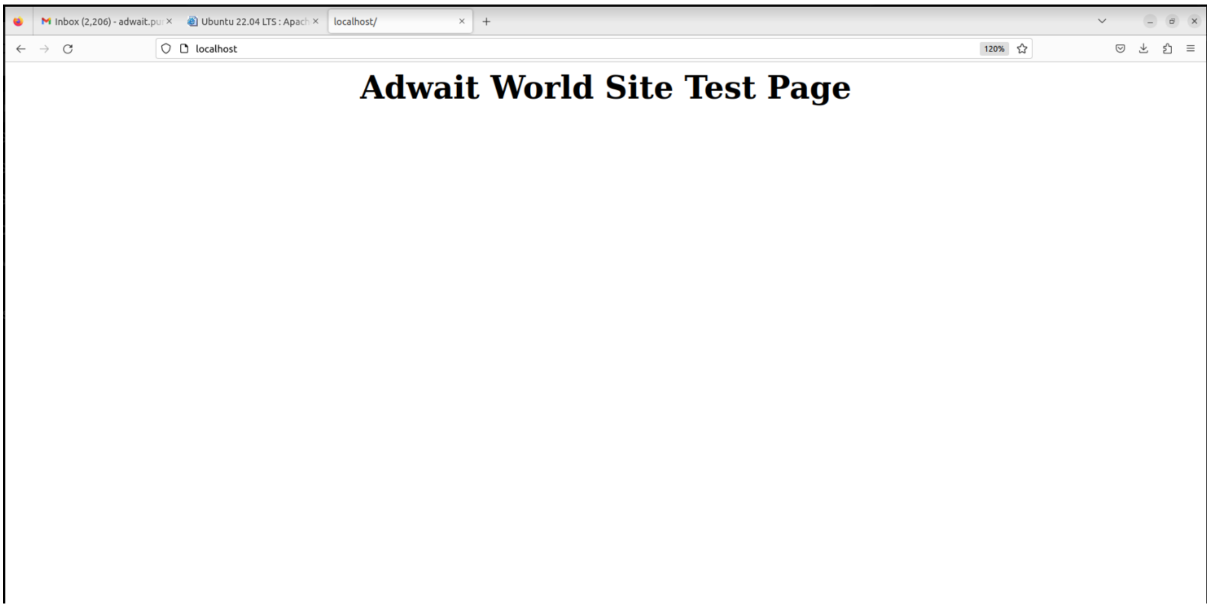
```
  GNU nano 6.2              /var/www/adwait.world/index.html
<html>
<body>
<div style="width: 100%; font-size: 40px; font-weight: bold; text-align: center>
Adwait World Site Test Page
</div>
</body>
</html>



                              [ Wrote 7 lines ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

Commands run till now:



```
adwait@adwait:~$ cd Desktop/
adwait@adwait:~/Desktop$ sudo apt -y install apache2
[sudo] password for adwait:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.52-1ubuntu4.3).
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver
  intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58
  libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0 libchromaprint1
  libcodec2-1.0 libdav1d5 libflashrom1 libflite1 libftdi1-2 libgme0 libgsm1
  libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libllvm13 libmfx1
  libmysofa1 libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55 librabbitmq4
  librubberband2 libserd-0-0 libshine3 libsord-0-0 libsratom-0-0
  libsrt1.4-gnutls libswresample3 libswscale5 libudfread0 libva-drm2
  libva-wayland2 libva-x11-2 libva2 libvdpau1 libvidstab1.1 libx265-199
  libxvidcore4 libzimg2 libzmq5 libzvbi-common libzvbi0
  linux-image-5.15.0-58-generic linux-modules-5.15.0-58-generic
  linux-modules-extra-5.15.0-58-generic mesa-va-drivers mesa-vdpau-drivers
  pocketsphinx-en-us va-driver-all vdpau-driver-all
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 109 not upgraded.
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/conf-enabled/security.conf
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/mods-enabled/dir.conf
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/apache2.conf
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/sites-enabled/000-default.conf
adwait@adwait:~/Desktop$ systemctl restart apache2
adwait@adwait:~/Desktop$ sudo nano /etc/apache2/sites-available/adwait.world.conf
adwait@adwait:~/Desktop$ sudo a2ensite adwait.world
Enabling site adwait.world.
To activate the new configuration, you need to run:
  systemctl reload apache2
adwait@adwait:~/Desktop$ systemctl reload apache2
adwait@adwait:~/Desktop$ mkdir /var/www/adwait.world
mkdir: cannot create directory ‘/var/www/adwait.world’: Permission denied
adwait@adwait:~/Desktop$ sudo mkdir /var/www/adwait.world
adwait@adwait:~/Desktop$ sudo nano /var/www/adwait.world/index.html
adwait@adwait:~/Desktop$ systemctl reload apache2
adwait@adwait:~/Desktop$ systemctl restart apache2
adwait@adwait:~/Desktop$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
adwait@adwait:~/Desktop$ systemctl reload apache2
adwait@adwait:~/Desktop$ 
```

Output:



Adwait World Site Test Page

Conclusion:

Through this experiment, we gained knowledge on configuring Apache servers, including the creation of our own server and its operation in localhost. We discovered that Apache is a dependable, cost-free, and straightforward approach to establishing a web server. Additionally, we familiarized ourselves with a range of new commands such as a2dissite, systemctl, and a2enssite. By utilizing the systemctl command, we were able to commence, discontinue, reload, and restart our Apache server. Ultimately, our accomplishment was being able to access our custom web page on the web server.