|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Shubham Golwal | | |
| **UID no.** | 20203000**15** | **Class:** | Comps A Batch |
| **Experiment No.** | 5 | | |

|  |  |
| --- | --- |
| **AIM:** | To setup a Apache Web server on a linux machine |
| **THEORY:** | **Whats is Apache Web Application Architecture?**  Apache is the web server that processes requests and serves web assets and content via HTTP. MySQL is the database that stores all your information in an easily queried format. PHP is the programming language that works with apache to help create dynamic web content.  Firewalls help protect the web application from both external threats and internal vulnerabilities depending on where the firewalls are configured. Load Balancers help distribute traffic across the web servers which handle the HTTP(S) requests (this is where Apache comes in) and application servers (servers that handle the functionality and workload of the web app.)  **Web Server Landscape**  The internet is comprised of many different technologies and not all of them are the same. While Apache is arguably one of the most popular web servers out there on the net, there are many other players and the landscape is always changing. Back in the late 90s and early 2000s, Apache’s dominance was very strong, serving over 50% of the internet's active websites. Microsoft's IIS (Internet Information Services) was also an option but not nearly as popular.  **Why Apache Web Servers?**  Apache is considered open source software, which means the original source code is freely available for viewing and collaboration. Being open source has made Apache very popular with developers who have built and configured their own modules to apply specific functionality and improve on its core features. Apache has been around since 1995 and is responsible as a core technology that helped spur the initial growth of the internet in its infancy.  **Features of Apache Web Server**   * Handling of static files * Loadable dynamic modules * Auto-indexing * .htaccess * Compatible with IPv6 * Supports HTTP/2 * FTP connections * Gzip compression and decompression * Bandwidth throttling * Perl, PHP, Lua scripts * Load balancing * Session tracking * URL rewriting * Geolocation based on IP address   APACHE CONFIGURATION FILES IN /ETC/APACHE2/ #  charset.conv  Specifies which character sets to use for different languages. Do not edit this file.  conf.d/\*.conf  Configuration files added by other modules. These configuration files can be included into your virtual host configuration where needed. See vhosts.d/vhost.template for examples. By doing so, you can provide different module sets for different virtual hosts.  default-server.conf  Global configuration for all virtual hosts with reasonable defaults. Instead of changing the values, overwrite them with a virtual host configuration.  errors.conf  Defines how Apache responds to errors. To customize these messages for all virtual hosts, edit this file. Otherwise overwrite these directives in your virtual host configurations.  listen.conf  Binds Apache to specific IP addresses and ports. Name-based virtual hosting is also configured here. For details, see Section 31.2.2.1.1, “Name-Based Virtual Hosts”.  magic  Data for the mime\_magic module that helps Apache automatically determine the MIME type of an unknown file. Do not change this file.  mime.types  MIME types known by the system (this actually is a link to /etc/mime.types). Do not edit this file. If you need to add MIME types not listed here, add them to mod\_mime-defaults.conf.  mod\_\*.conf  Configuration files for the modules that are installed by default. Refer to Section 31.4, “Installing, Activating, and Configuring Modules” for details. Note that configuration files for optional modules reside in the directory conf.d.  server-tuning.conf  Contains configuration directives for the different MPMs (see Section 31.4.4, “Multiprocessing Modules”) as well as general configuration options that control Apache's performance. Properly test your Web server when making changes here.  ssl-global.conf and ssl.\*  Global SSL configuration and SSL certificate data. Refer to Section 31.6, “Setting Up a Secure Web Server with SSL” for details.  sysconfig.d/\*.conf  Configuration files automatically generated from /etc/sysconfig/apache2. Do not change any of these files—edit /etc/sysconfig/apache2 instead. Do not put other configuration files in this directory.  uid.conf  Specifies under which user and group ID Apache runs. Do not change this file. |
| **EXPERIMENT** | |
| **PROCEDURE:** | **SETTING UP THE WEB SERVER:**  The Apache2 web server is available in Ubuntu Linux. To install Apache2:   1. At a terminal prompt enter the following command:   *root@www:~# sudo apt -y install apache2*       1. Configure Apache2:   *root@www:~# sudo gedit /etc/apache2/conf-enabled/security.conf* # line 25: change the ServerTokens from OS to Prod  *ServerTokens Prod*         1. Enter the following command on the terminal:   *root@www:~# sudo gedit /etc/apache2/mods-enabled/dir.conf*    # line 2: add file name that it can access only with directory's name  DirectoryIndex index.html index.htm     1. Enter the following command on the terminal:   *root@www:~# sudo gedit /etc/apache2/apache2.conf*  # line 70: add to specify server name  ServerName [www.srv.world](http://www.srv.world/)        5)Enter the following command on the terminal:  *root@www:~# sudo gedit /etc/apache2/sites-enabled/000-default.conf*    # line 11: change to webmaster's email  ServerAdmin webmaster@srv.world         1. Enter the following command on the terminal:   *root@www:~# sudo systemctl restart apache2*             1. Access to [http://(your server's hostname or IP address)/] with web browser.        1. Entering the ip address of the webpage(one can even enter the url):            1. The webpage gets loaded: |
| **RESULT:** This experiment taught me how to set up Apache web server and how to load a website using Inet address. Learnt how the Apache’s web services and its features. | |