**Apache Web Server – Lecture 1/2 – 07/02/2022**

* Install:

$ sudo apt update

$ sudo apt install apache2

$ sudo service apache2 status

* Start:

$ sudo service apache2 start

* Get IP and Port:

IP: $ ip a

Text

Description automatically generated

Port: $ sudo ss -ltn



* Change Port:

$ sudo nano /etc/apache2/ports.conf

Text

Description automatically generated

$ sudo service apache2 restart

* Change User and Group:

Create User: $ sudo useradd -c "Apache2 User" -m -s /usr/sbin/nologin apache2

Change from Apache environment vars: $ sudo nano /etc/apache2/envvars

Text

Description automatically generated

Restart: $ sudo service apache2 restart

Text

Description automatically generated

Note: The new user has read permission on the web application.

* Default logs files:

Access log: /var/log/apache2/access.log

Error log: /var/log/apache2/error.log

* To print current MPM: apachectl -V 
* To disable a module: $ sudo a2dismod <module\_name> 🡪 $ sudo a2dismod mpm\_event

$ sudo service apache2 restart

* To enable a module: $ sudo a2enmod <module\_name> 🡪 $ sudo a2enmod mpm\_event

$ sudo service apache2 restart

* All modules available: /etc/apache2/mods-available/
* All enabled modules: /etc/apache2/mods-enabled 🡪 link from /etc/apache2/mods-available/

Text

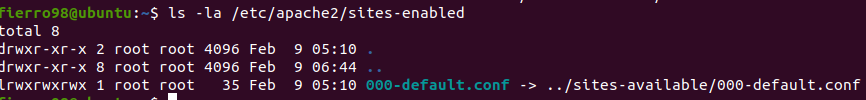
Description automatically generated

* Web server hosts web application (Website):

1. Hostname: Site URL
2. Document Root: Directory contains site web app files
3. URL+IP+PORT: Virtual Host

All configured sites: /etc/apache2/sites-available

All enabled sites (Must be configured 1st) : /etc/apache2/sites-enabled 🡪 link from /etc/apache2/sites-available



* Example:

$ sudo nano /etc/apache2/sites-available/000-default.conf

A picture containing graphical user interface

Description automatically generated

Default virtual host has DocumentRoot: /var/www/html

Calendar

Description automatically generated

To add new page: $ sudo nano /var/www/html/first.html

Text

Description automatically generated Graphical user interface

Description automatically generated

* Configurations:

Main: /etc/apache2/apache2.conf

* In apache2 there is configuration for directory called directive

<Directory DirectoryName>

</Directory>

$ sudo nano /etc/apache2/apache2.conf

Text

Description automatically generated

To apply certain configuration on any client request to this directory and its child.

If child has different directory directive, then the child config will be applied.

* /var/www/html 🡪have a config

Any directory under /var/www/html will have the same configuration.

* /var/www/html 🡪have a config1 & /var/www/html/site1 🡪have a config2

Any directory under /var/www/html will have the same configuration (config1)

but /var/www/html/site1 has a different configuration (config2).

* Can control who has permission to visit the directory content

Require: Specify who ca visit the directory

username/IP

grant/deny

* Example – To deny all users from access g1

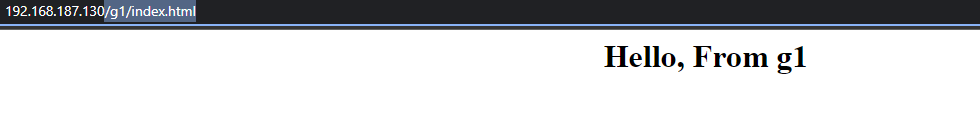
1. Create directory g1 in /var/www/html

$ sudo mkdir /var/www/html/g1

1. Create a html page

$ sudo nano /var/www/html/g1/index.html

Text

Description automatically generated 

1. Add <Directory> in config files

Create /etc/apache2/con-available/deny-g1.conf

$ sudo nano /etc/apache2/confs-available/deny-g1.conf

Text

Description automatically generated

<Directory /var/www/html/g1>

Require all denied

</Directory>

1. Enable configuration

$ sudo a2enconf deny-g1

1. Restart

$ sudo service apache2 restart

Graphical user interface, text, application, email

Description automatically generated

1. Error log: to see access denied

$ sudo tail /var/log/apache2/error.log

Text

Description automatically generated

1. Disable configuration

$ sudo a2disconf deny-g1

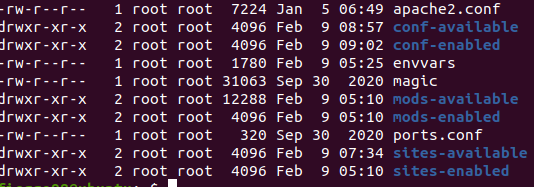
$ sudo service apache2 restart

* Summary:
* Apache conf file may include a variable, the variables are stored in

1. BASH Shell 2- Apache environment vars: /etc/apache2/envars

* Different directories under /etc/apache2

$ ls -la /etc/apache2/



1. conf-available

Existing configuration files, end with .conf

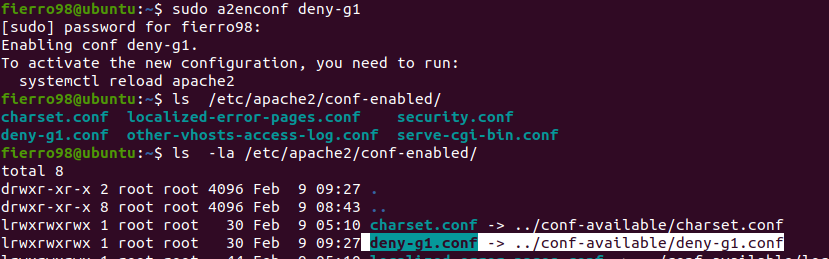
The configuration in this directory may be activated or maybe not.

1. conf-enabled 🡪 link from conf-available

The activated configuration, which must reside in conf-available.

Enable configuration: a- $ sudo a2enconf <cong-name> [Create link from conf-available]

b- $ sudo service apache2 restart



Disable configuration: a- $ sudo a2disconf <cong-name> [Remove link from conf-available]

b- $ sudo service apache2 restart

Text

Description automatically generated

1. mods-available

Existing installed modules, end with .conf

The modules in this directory may be activated or maybe not.

* + - MPM (Multi-Processing Modules)

A screenshot of a computer

Description automatically generated with medium confidence

1. mods-enabled 🡪 link from mods-available

The activated Modules, which must reside in mods-available.

Disable a module: a- $ sudo a2dismod <module\_name> [Create link from mods-available]

b- $ sudo service apache2 restart

Enable a module: a- $ sudo a2enmod <module\_name> [Remove link from mods-available]

b- $ sudo service apache2 restart

1. sites-available

The configured virtual host.

The virtual host may be activated or maybe not.

1. sites-enabled 🡪 link from sites -available

The activated virtual hosts which must be in sites -available.

Disable a site: a- $ sudo a2dissite <site\_name> [Create link from sites-available]

b- $ sudo service apache2 restart

Enable a site: a- $ sudo a2ensite <site\_name> [Remove link from sites-available]

b- $ sudo service apache2 restart

Text

Description automatically generated

* .htaccess:

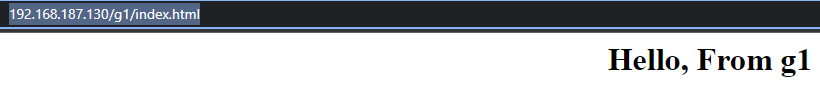
There is a file called .htaccess, Add the file to the directory include all configuration

.htaccess can be used: [Access list – Authentication – Default pages and modules – Mod rewrite].

$ sudo nano /var/www/html/g1/.htaccess



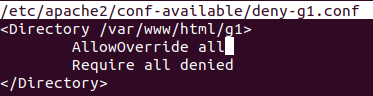
Require all denied



Nothing happens

If a directive from .htaccess conflicts with main server configuration, will return to   
AllowOverride.

$ sudo nano /etc/apache2/con-available/deny-g1.conf



Text

Description automatically generated

<Directory /var/www/html/g1>

AllowOverride all

Require all denied // Remove it because it is already on .htaccess

</Directory>

Text

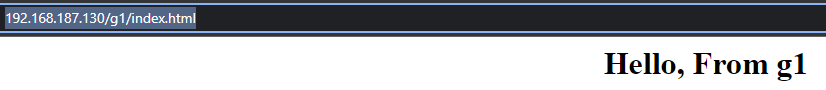
Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated



**Apache Web Server – Lecture 2/2 – 09/02/2022**

* **Apache authentication:**

$ cd /var/www/html/g1

$ sudo nano .htaccess

|  |  |
| --- | --- |
| Graphical user interface, text  Description automatically generated | AuthType Basic  AuthName "g1 Area"  AuthUserFile /etc/.htpasswd  Require valid-user |

To add a username for the 1st time

$ sudo htpasswd -c /etc/.htpasswd ahmed

To add a username after that

$ sudo htpasswd /etc/.htpasswd mahmoud

Text

Description automatically generated Shape

Description automatically generated 

* Open other location with js:

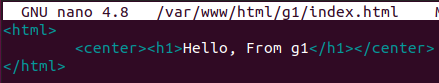
$ sudo nano /var/www/html/g1/index2.html

|  |  |
| --- | --- |
|  | <html>  <center><h1>Hello, From g1 - Page 2</h1></center>  </html> |

$ sudo nano /var/www/html/g1/index.html

|  |  |
| --- | --- |
|  | <script>  location.href="/g1/index2.html";  </script>  <html>  <center><h1>Hello, From g1</h1></center>  </html> |

🡪

\* Remove <script>…..</script> for mod\_rewrite 

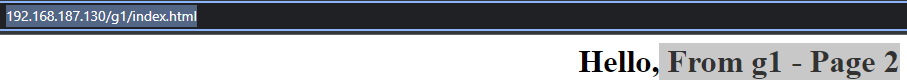
* mod\_rewrite
* RewriteEngine on: turn the engine on, so the rules would take effects
* RewriteEngine off: turn the engine off, so the rules would not take effects. Can be
* configured in the .conf files or in .htaccess.
* RewriteRule: used to perform the URL rewrite operations.

Enable mod\_rewrite: $ sudo a2enmod rewrite

$ sudo service apache2 restart

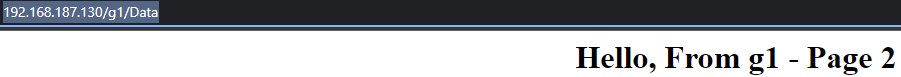
$ sudo nano /var/www/html/g1/.htaccess

|  |  |
| --- | --- |
|  | RewriteEngine on  RewriteRule index.html$ /g1/index2.html |



* Uses RegEx. ([NC] no case write it at the end of RewriteRule line for case insensitive]

🡪 RewriteRule ^data/{0,1}$ /g1/index2.html [NC]



* RewriteRule /?data$ /get1.html

🡪 Will rewrite the URL to get the file get1.html if the URL path ends with /data.

* RewriteRule /?data$ /get1.html [NC]

🡪 Will rewrite the URL to get the file get1.html if the URL path ends with /data   
with any case.

* RewriteRule ^/?get/([a-zA-Z\_]+)/([0-9]+)/?$ get1.php?fname=$1&fage=$2

🡪 Wil convert the directory-based URL to Query String. To Rewrite (2) -> (1)

(1)http://www.site/g1/test.html?name=ahmed&age=30

(2)http://www.site/g1/get/ahmed/30

* RewriteCond: used to construct conditions control the URL rewrite operations.

RewriteCond %{QUERY\_STRING} "noha "

RewriteRule .? http://%{HTTP\_HOST}/noha.html? [R]

RewriteCond, will test the QUERY\_STRING if contains the pattern noha

If returns true, the next RerwriteRule will be executed.

RewriteRule will replace all the url with the new string and terminates the   
URL path with ? To remove the QUERY\_STRING.

To bind them with or, use the flag [OR]

* Virtual Hosts:

The term VirtualHost refers to that run multiple web sites on the same web server.

* For site mahmoudkamal.com

1. Create a file /etc/apache2/sites-available/mahmoudkamal.com.conf

$ sudo nano /etc/apache2/sites-available/mahmoudkamal.com.conf

|  |  |
| --- | --- |
| Text  Description automatically generated | <VirtualHost \*:80>  ServerName mahmoudkamal.com  ServerAlias www.mahmoudkamal.com  DocumentRoot /var/websites/mahmoudkamal.com  </VirtualHost>  <Directory /var/websites/mahmoudkamal.com >  AllowOverride all  Require all granted  </Directory> |

1. Create Directory /var/websites/mahmoudkamal.com

$ sudo mkdir -p /var/websites/mahmoudkamal.com

1. Create index.html inside this Directory

$ sudo nano /var/websites/mahmoudkamal.com/index.html

|  |  |
| --- | --- |
| Text  Description automatically generated | <html>  <body>  <center><h1>Welcome to Mahmoud Kamal website</h1></center>  </body>  </html> |

1. Enable site then reload apache2

$ sudo a2ensite mahmoudkamal.com

$ sudo service apache2 reload

Text

Description automatically generated

1. Get IP

$ ip a

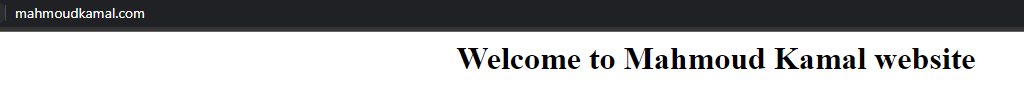
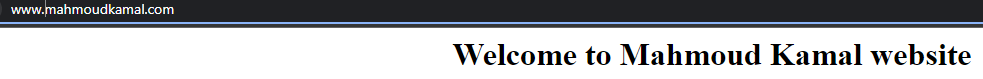
Text

Description automatically generated

1. Open notepad on Windows as administrator

Open Hosts files 🡪 C:\Windows\System32\drivers\etc\hosts

|  |  |
| --- | --- |
|  | 192.168.187.130 mahmoudkamal.com  192.168.187.130 www.mahmoudkamal.com |



* To configure apache for SSL

1-Generate key pairs (public, private)

* + Generate private key

$ openssl genrsa -out mykey.priv 2048

* + Generate public key

$ openssl rsa -in mykey.priv -pubout > mykey.pub

* + Secure private key

$ chmod o-r mykey.priv

2-Generate CSR

$ openssl req -new -key mykey.priv -out mycsr.csr

Text

Description automatically generated

3-Pay for the certificate or use self-signed certificate

$ openssl x509 -req -days 365 -in mycsr.csr -signkey mykey.priv -sha256 -out mycert.crt

4-Configure apache2 for SSL

* + Enable apache for ssl

$ sudo a2enmod ssl

* + Configure the SSL virtualhost /etc/apache2/sites-available/default-ssl.conf

Change certifictes paths SSLCertificateFile,SSLCertificateKeyFile

$ sudo cp mycert.crt /etc/ssl/certs

$ sudo cp mykey.priv /etc/ssl/private

$ sudo nano /etc/apache2/sites-available/default-ssl.conf

Text

Description automatically generated

* + Enable SSL Site

$ sudo a2ensite default-ssl

* + Restart

$ sudo service apache2 restart

A screenshot of a computer

Description automatically generated with medium confidence Graphical user interface, text, application, email

Description automatically generated

* Rewrite rule to modify url for http connections to https using .htaccess file [Lab1]

$ sudo nano /var/www/html/g1/.htaccess

|  |  |
| --- | --- |
|  | RewriteEngine On  RewriteCond %{HTTPS} !=on  RewriteRule ^(.\*)$ https://%{HTTP\_HOST}%{REQUEST\_URI} [L,R=301,NE] |

🡪

* Setup php with apache2

1. PHP as interpreter: Prefork MPM

Web server: fork php interpreter, and pass to it the php requested file

1. Replace mpm\_event into mpm\_prefork

$ sudo a2dismod mpm\_event

$ sudo a2enmod mpm\_prefork

1. Install php, php-apache-module, php extensions

$ sudo apt install php7.4 php7.4-mysql

1. Restart apache

$ sudo service apache2 restart

1. Test

create a file "info.php" under document root

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

|  |  |
| --- | --- |
|  | <?php  phpinfo();  ?> |

Graphical user interface

Description automatically generated with medium confidence

1. To remove:

$ sudo service apache2 stop

$ sudo apt remove php\*

$ sudo apt purge php\*

$ sudo a2dismod mpm\_prefork

$ sudo a2enmod mpm\_event

$ sudo service apache2 restart

Text

Description automatically generated

1. PHP as fastCGI (Preferred): Event MPM/Worker MPM

Service from php called php-fpm.

Require a config in apache2 to use php-frpm.

\* Apache2 must be configured with event\_mpm.

1. Install php, php-apache-module, php-fpm, php extensions

$ sudo apt install php7.4 php7.4-mysql php7.4-fpm

1. Configure apache2

$ sudo a2enmod proxy\_fcgi setenvif

$ sudo a2enconf php7.4-fpm

1. Change from Apache environment vars:

$ sudo nano /etc/apache2/envvars

|  |  |
| --- | --- |
|  | export APACHE\_RUN\_USER=apache2 🡪 export APACHE\_RUN\_USER=www-data  export APACHE\_RUN\_GROUP=apache2 🡪 export APACHE\_RUN\_GROUP= www-data |

1. Restart FPM and apache

$ sudo service php7.4-fpm restart

$ sudo service apache2 restart

Text

Description automatically generated

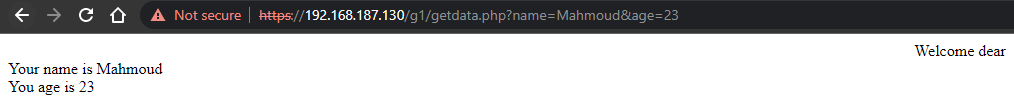
1. Test



create a file "getdata.php" under document root

$ sudo nano /var/www/html/g1/getdata.php

|  |  |
| --- | --- |
|  | <html>  <title>Welcome to my PHP page</title>  <body>  <?php  echo "<center>Welcome dear</center>";  $NAME=$\_GET["name"];  $AGE=$\_GET["age"];  echo "Your name is $NAME<br>";  echo "You age is $AGE<br>";  ?>  </body>  </html> |

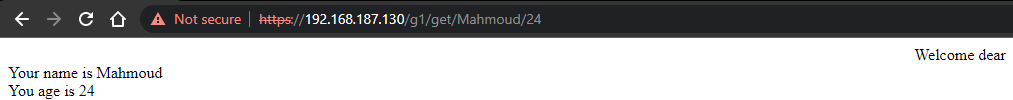


1. Write url module that convert from directory URL to querystring.

http://192.168.187.130/g1/getdata.php?name=Mahmoud&age=23 🡪 http://192.168.187.130/g1/get/Mahmoud/23

$ sudo nano /var/www/html/g1/.htaccess

|  |  |
| --- | --- |
|  | RewriteRule ^get/([a-zA-Z\_]+)/([0-9]+)/{0,1}$ /g1/getdata.php?name=$1&age=$2 |



* To setup wordpress:

1. Create a database on mariadb (mysql)

$ sudo mysql -u root -p

mysql> create database mywordpress;

mysql> create user wpadmin@localhost identified by 'Password123#@!';

mysql> grant all privileges on mywordpress.\* to wpadmin@localhost;

mysql> flush privileges;

mysql> exit;

* + - Name: mywordpress
    - User: wpadmin
    - Password: Password123#@!

1. Download wordpress

$ cd /var/www/html

$ sudo wget https://wordpress.org/latest.tar.gz

$ sudo tar zxf latest.tar.gz



* + - * Ip/wordpress

Graphical user interface, text, application

Description automatically generated

$ cd /var/www/html/wordpress/

$ sudo nano wp-config.php

|  |  |
| --- | --- |
|  |  |

1. Wordpress

|  |  |
| --- | --- |
| Graphical user interface, text, application, email  Description automatically generated | Username: admin  Password: 5Atqaa(8Bf@\*06@sm#  192.168.187.130/wordpress/wp-admin/ |

A screenshot of a computer

Description automatically generated with medium confidence

* To setup phpmyadmin
  + You have to install MySQL, apache and php.

$ sudo apt install phpmyadmin php-mbstring

$ sudo phpenmod mbstring

$ sudo service php7.4-fpm restart

$ sudo service apache2 restart

* + To solve the root login for phpmyadmin

Connect to mysql with root

mysql> update user set plugin='' where User='root';

mysql> flush privileges;

* Web application test:
  + - Expected number of users ?. For example: 1000 users
    - Load test: How system behave on max load.
    - Stress test: How system behave on the high extreme. (DoS attack)
    - ASF developed java app called Apache-JMeter. <https://jmeter.apache.org/>

Apache-JMeter [Windows]:

1. Download: <https://jmeter.apache.org/download_jmeter.cgi>

<https://dlcdn.apache.org//jmeter/binaries/apache-jmeter-5.4.3.zip>

1. Extract apache-jmeter-5.4.3.zip
2. CMD: D:\apache-jmeter-5.4.3\bin>jmeter.bat

A screenshot of a computer

Description automatically generated with medium confidence

1. Add >> Threads (Users) >> Thread Group

A screenshot of a computer

Description automatically generated with medium confidence

1. Set Thread Group Name, Comments and Thread properties

Graphical user interface, text, application, email

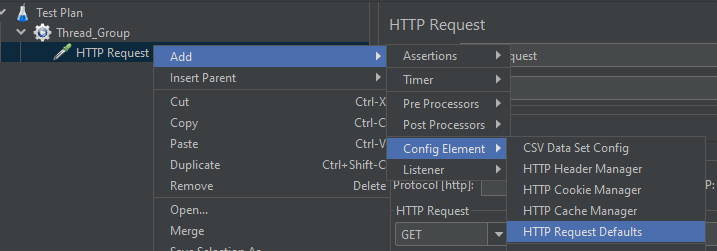
Description automatically generated

1. Add >> Sampler >> HTTP Request

A screenshot of a computer

Description automatically generated

1. Add >> Config Element >> HTTP Request Defaults



1. Write IP and Path

Graphical user interface, text, application

Description automatically generated

1. Add >> Listener >> Summery Report

A screenshot of a computer

Description automatically generated with medium confidence

1. Add >> Listener >> View Results Tree

A screenshot of a computer

Description automatically generated with medium confidence

1. Save

Graphical user interface, text, application

Description automatically generated

Recording Tests: <https://jmeter.apache.org/usermanual/jmeter_proxy_step_by_step.html>

* + CMD
    - Don't use GUI mode for load testing! only for Test creation and Test debugging.
    - For load testing, use CLI Mode (was NON GUI):

jmeter -n -t [jmx file] -l [results file] -e -o [Path to web report folder]

& increase Java Heap to meet your test requirements:

Modify current env variable HEAP="-Xms1g -Xmx1g -XX:MaxMetaspaceSize=256m" in the jmeter batch file

* Check: <https://jmeter.apache.org/usermanual/best-practices.html>

**Apache Web Server – Lab – 09/02/2022**

1-Rewrite rule to modify url for http connections to https using .htaccess file

\* Page 10 and scroll up for step by step in task 1 [Page 1 – Page 10]

$ sudo nano /var/www/html/g1/.htaccess

|  |  |
| --- | --- |
| Graphical user interface, text  Description automatically generated | RewriteEngine On  RewriteCond %{HTTPS} !=on  RewriteRule ^(.\*)$ https://%{HTTP\_HOST}%{REQUEST\_URI} [L,R=301,NE] |

🡪

2-Using Virtual host configuration

1-Setup laravel framework (iti.org)

[<https://www.hostinger.com/tutorials/how-to-install-laravel-on-ubuntu-18-04-with-apache-and-php/>]

[<https://www.howtoforge.com/tutorial/install-laravel-on-ubuntu-for-apache/>] [<https://laravel.com/docs/8.x/installation>]

* For site iti.org without Laravel [This Steps isn’t necessary for lab start from next page to setup laravel]

1. Create a file /etc/apache2/sites-available/iti.org.conf

$ sudo nano /etc/apache2/sites-available/iti.org.conf

|  |  |
| --- | --- |
|  | <VirtualHost \*:80>  ServerName iti.org  DocumentRoot /var/websites/iti.org  </VirtualHost>  <Directory /var/websites/iti.org >  AllowOverride all  Require all granted  </Directory> |

1. Create Directory /var/websites/iti.org

$ sudo mkdir -p /var/websites/iti.org

1. Create index.html inside this Directory

$ sudo nano /var/websites/iti.org/index.html

|  |  |
| --- | --- |
|  | <html>  <body>  <center><h1>Welcome to ITI.org</h1></center>  </body>  </html> |

1. Enable site then reload apache2

$ sudo a2ensite iti.org

$ sudo service apache2 reload

Text

Description automatically generated

1. Get IP

$ ip a

Text

Description automatically generated

1. Open notepad on Windows as administrator

Open Hosts files 🡪 C:\Windows\System32\drivers\etc\hosts

|  |  |
| --- | --- |
|  | * + - 1. iti.org |



* To setup laravel:

1. Install Apache Web Server & Install and Configure PHP 7.4

$ sudo apt update

$ sudo apt install apache2

$ sudo apt install libapache2-mod-php php php-common php-xml php-gd php-opcache php-mbstring php-tokenizer php-json php-bcmath php-zip unzip

$ sudo service apache2 restart

1. Install Composer PHP Packages Management

$ sudo apt install curl

$ curl -sS https://getcomposer.org/installer | php

$ sudo mv composer.phar /usr/local/bin/composer

$ composer --version

Text

Description automatically generated

1. Install Laravel 8.x on Ubuntu 20.04

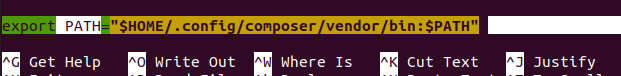
\* This section will install the Laravel web framework with the non-root.

$ composer global require laravel/installer

$ nano ~/.bashrc

Paste the following configuration to the end of the line.

export PATH="$HOME/.config/composer/vendor/bin:$PATH"



$ source ~/.bashrc

$ echo $PATH

$ laravel new blog

\* To make it accessible for the Apache webserver to access your Laravel project directory, change the project directory's group owner to the group 'www-data' and make the 'storage' directory writable.

$ sudo chgrp -R www-data /home/$USERNAME/blog



Graphical user interface

Description automatically generated with medium confidence$ sudo chmod -R 775 /home/$USERNAME/blog/storage

1. Install Laravel Via Composer create-project

$ cd /var/websites/

$ sudo composer create-project --prefer-dist laravel/laravel blog

$ sudo chown -R www-data:www-data /var/websites/blog

$ sudo chmod -R 775 /var/websites/blog/storage

$ cd /var/websites/blog

$ sudo composer install

$ sudo chmod 775 /var/websites/blog/public/favicon.ico

1. Setup Apache for Laravel Project

$ cd /etc/apache2/sites-available/

$ sudo nano laravel.conf

|  |  |
| --- | --- |
|  | GNU nano 4.8 /etc/apache2/sites-available/laravel.conf Modified  <VirtualHost \*:80>  ServerName iti2.org  ServerAlias www.iti2.org  DocumentRoot /var/websites/blog/public  <Directory /var/websites/blog/public>  Options Indexes MultiViews  AllowOverride all  Require all granted  </Directory>  ErrorLog ${APACHE\_LOG\_DIR}/error.log  CustomLog ${APACHE\_LOG\_DIR}/access.log combined  </VirtualHost> |

$ sudo a2enmod rewrite

$ sudo a2dissite iti.org.conf

$ sudo a2ensite laravel.conf

$ sudo service apache2 restart

$ sudo service apache2 reload

1. Open notepad on Windows as administrator

Open Hosts files 🡪 C:\Windows\System32\drivers\etc\hosts

|  |  |
| --- | --- |
|  | 192.168.187.130 www.iti2.org  192.168.187.130 iti2.org |

2-Using Virtual host configuration

2-Setup wordpress (wp.iti.org)

* For site wp.iti.org

1. Create a file /etc/apache2/sites-available/wp.iti.org.conf

$ sudo nano /etc/apache2/sites-available/wp.iti.org.conf

|  |  |
| --- | --- |
|  | <VirtualHost \*:80>  ServerName wp.iti.org  DocumentRoot /var/websites/wp.iti.org  </VirtualHost>  <Directory /var/websites/wp.iti.org >  AllowOverride all  Require all granted  </Directory> |

1. Create Directory /var/websites/wp.iti.org

$ sudo mkdir -p /var/websites/wp.iti.org

1. Create index.html inside this Directory

$ sudo nano /var/websites/wp.iti.org/index.html

|  |  |
| --- | --- |
|  | <html>  <body>  <center><h1>Welcome to Wp ITI</h1></center>  </body>  </html> |

1. Enable site then reload apache2

$ sudo a2ensite wp.iti.org

$ sudo service apache2 reload

Text

Description automatically generated

1. Get IP

$ ip a

Text

Description automatically generated

1. Open notepad on Windows as administrator

Open Hosts files 🡪 C:\Windows\System32\drivers\etc\hosts

|  |  |
| --- | --- |
|  | * + - 1. wp.iti.org |



* To setup wordpress:

1. Create a database on mariadb (mysql)

$ sudo mysql -u root -p

mysql> create database mywordpress;

mysql> create user wpadmin@localhost identified by 'Password123#@!';

mysql> grant all privileges on mywordpress.\* to wpadmin@localhost;

mysql> flush privileges;

mysql> exit;

* + - Name: mywordpress
    - User: wpadmin
    - Password: Password123#@!

1. Download wordpress

$ cd /var/websites/wp.iti.org

$ sudo wget https://wordpress.org/latest.tar.gz

$ sudo tar zxf latest.tar.gz

Text

Description automatically generated

* + - * Ip/wordpress

Graphical user interface, text, application

Description automatically generated

$ cd /var/websites/wp.iti.org/wordpress/

$ sudo nano wp-config.php

|  |  |
| --- | --- |
|  |  |

1. Wordpress

|  |  |
| --- | --- |
| Graphical user interface, text, application, email  Description automatically generated | Username: admin  Password: 5Atqaa(8Bf@\*06@sm#  192.168.187.130/wordpress/wp-admin/ |

* Copy wordpress files and directories to /var/websites/wb.iti.org & delete index.html

$ cd /var/websites/wp.iti.org/wordpress/

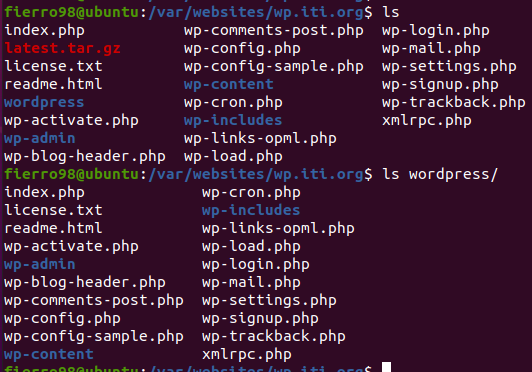
$ sudo cp -r \* ../

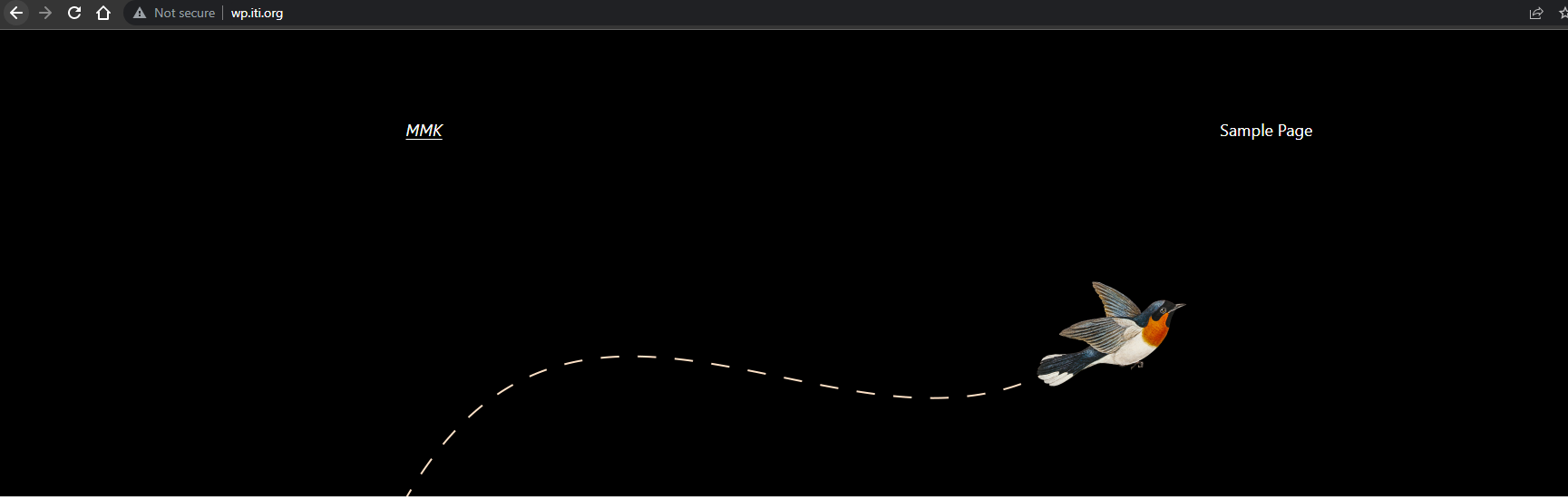
$ cd ..

$ sudo rm index.htm

\* Delete wordpress Directory inside /var/websites/wb.iti.org

$ sudo rm -r wordpress



http://wp.iti.org/Text

Description automatically generated with low confidence

3-Using jmeter, create a recorded plan for your wordpress site.

* Update the site with form for test

$ cd /var/websites/wp.iti.org/

$ sudo nano index.html

|  |
| --- |
| Text  Description automatically generated  <html>  <head>  <title>Applicatio to demo JMeter</title>  </head>  <body>  <table width=100%>  <tr>  <center><h1>To demo the operation of Apache-JMeter from - ASF -</h1></center>  </tr>  <tr>  <form name="loginform" action="mainmenu.php" method="post">  <table width=100%>  <tr>  <td width=30%>Username</td>  <td width=70%><input name="username"></td>  </tr>  <tr>  <td width=30%>Password</td>  <td width=70%><input name="password" type= password></td>  </tr>  <tr>  <td width=30%><input type=submit></td>  <td width=70%><input type=reset></td>  </tr>  </table>  </form>  </tr>  </table>  </body>  </html> |

$ sudo nano mainmenu.php

|  |  |
| --- | --- |
|  | <html>  <title>Main Menu</title>  <body>  <?php  $USER=$\_POST["username"];  $PASS=$\_POST["password"];  If($USER == "fierro" && $PASS == "password") {  echo "$USER is authenticated";  } else {  echo "$USER is access denied";  }  ?>  <br>  <a href="index.html">Back to login</a>  </body>  </html> |

* Apache-JMeter [Windows]:

1. Download: <https://jmeter.apache.org/download_jmeter.cgi>

<https://dlcdn.apache.org//jmeter/binaries/apache-jmeter-5.4.3.zip>

1. Extract apache-jmeter-5.4.3.zip
2. CMD: D:\apache-jmeter-5.4.3\bin>jmeter.bat

A screenshot of a computer

Description automatically generated with medium confidence

1. Add >> Threads (Users) >> Thread Group

A screenshot of a computer

Description automatically generated with medium confidence

1. Set Thread Group Name, Comments and Thread properties

Graphical user interface, text, application, email

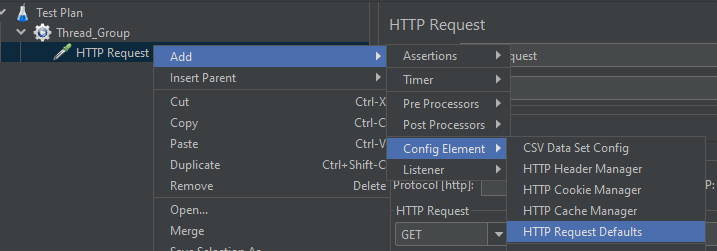
Description automatically generated

1. Add >> Sampler >> HTTP Request

A screenshot of a computer

Description automatically generated

1. Add >> Config Element >> HTTP Request Defaults



1. Write IP and Path

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

1. Add >> Listener >> Summery Report

A screenshot of a computer

Description automatically generated with medium confidence

1. Add >> Listener >> View Results Tree

A screenshot of a computer

Description automatically generated with medium confidence

1. Save

Graphical user interface, text, application

Description automatically generated

1. Results

Graphical user interface, application

Description automatically generated Text

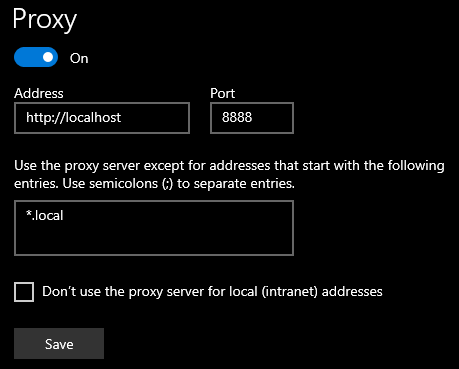
Description automatically generated

* Recording test

Recording Tests: <https://jmeter.apache.org/usermanual/jmeter_proxy_step_by_step.html>

1. Graphical user interface, application

   Description automatically generated
2. Graphical user interface, application

   Description automatically generated
3. 
4. Graphical user interface

   Description automatically generated with low confidence
5. Text

   Description automatically generated
6. Save

Graphical user interface, application

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated

* CMD

CMD: D:\apache-jmeter-5.4.3\bin> jmeter -n -t Recording.jmx -l results.txt -e -o report

Graphical user interface, text

Description automatically generated

Graphical user interface, application, table

Description automatically generated

\*bin folder uploaded on GitHub

<https://github.com/MahmoudFierro98/ITI_OpenSourceApplicationDevelopment/tree/main/Apache_Web_Server>