

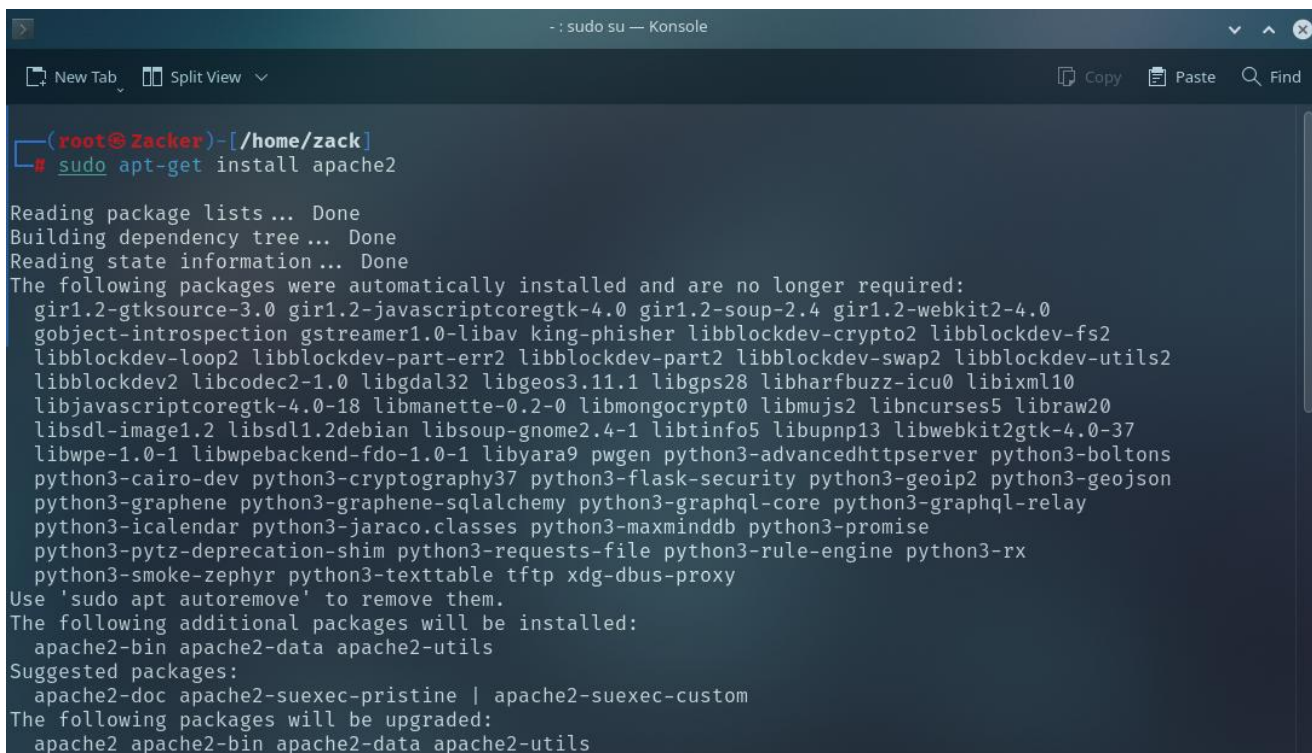
INFRASTRUCTURE SERVICE

ADMINISTRATION AND SECURITY

Installing web server in Kali Linux:

- 1) We install Apache with the following command:

```
sudo apt-get install apache2
```



```
- : sudo su — Konsole

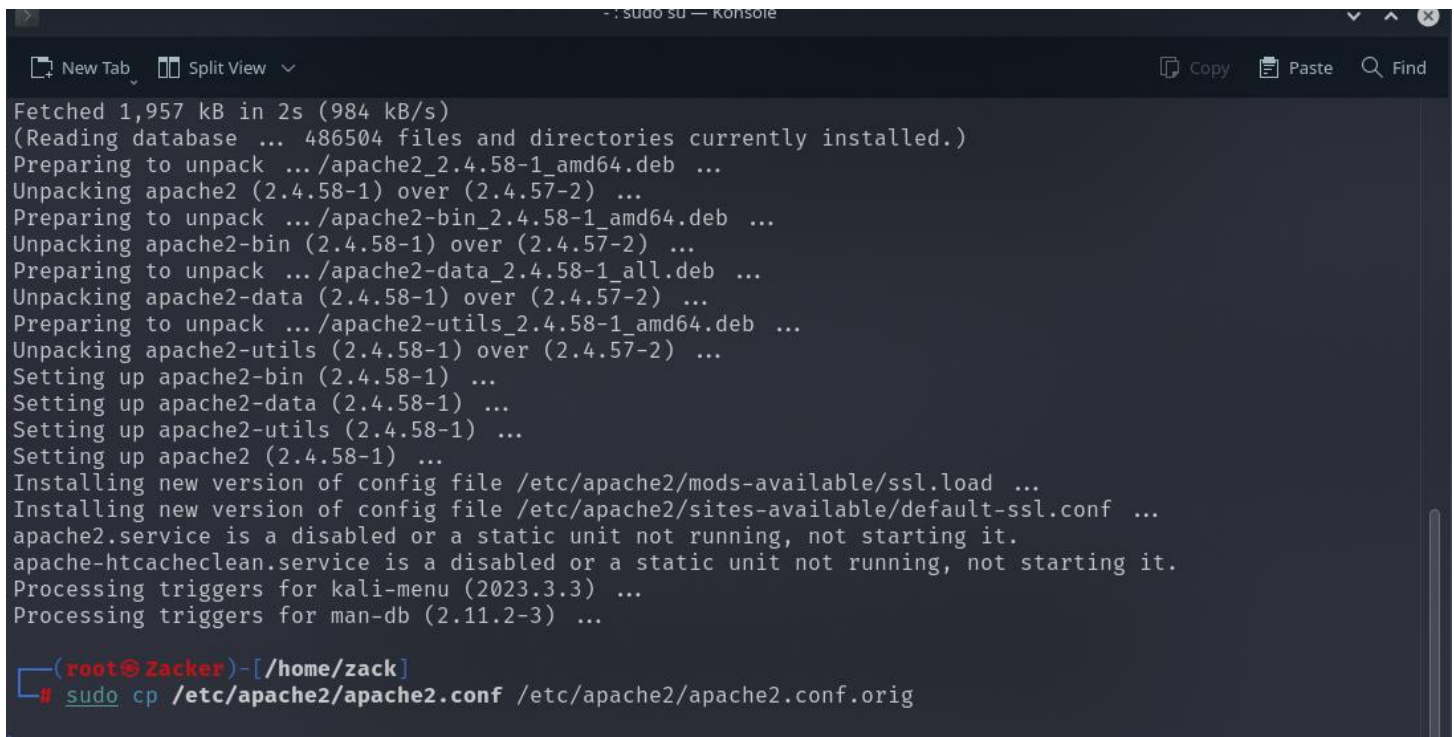
New Tab Split View Copy Paste Find

(root@Zacker)-[/home/zack]
# sudo apt-get install apache2

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
 gir1.2-gtksource-3.0 gir1.2-javascriptcoregtk-4.0 gir1.2-soup-2.4 gir1.2-webkit2-4.0
 gobject-introspection gstreamer1.0-libav king-phisher libblockdev-crypto2 libblockdev-fs2
 libblockdev-loop2 libblockdev-part-err2 libblockdev-part2 libblockdev-swap2 libblockdev-utils2
 libblockdev2 libcodec2-1.0 libgdal32 libgeos3.11.1 libgps28 libharfbuzz-icu0 libixml10
 libjavascriptcoregtk-4.0-18 libmanette-0.2-0 libmongocrypt0 libmujs2 libncurses5 libraw20
 libsdl-image1.2 libsdl1.2debian libsoup-gnome2.4-1 libtinfo5 libupnp13 libwebkit2gtk-4.0-37
 libwpe-1.0-1 libwpebackend-fdo-1.0-1 libyara9 pwgen python3-advancedhttpserver python3-boltons
 python3-cairo-dev python3-cryptography37 python3-flask-security python3-geoip2 python3-geojson
 python3-graphene python3-graphene-sqlalchemy python3-graphql-core python3-graphql-relay
 python3-icalendar python3-jaraco.classes python3-maxminddb python3-promise
 python3-pytz-deprecation-shim python3-requests-file python3-rule-engine python3-rx
 python3-smoke-zephyr python3-texttable tftp xdg-dbus-proxy
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
 apache2-bin apache2-data apache2-utils
Suggested packages:
 apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following packages will be upgraded:
 apache2 apache2-bin apache2-data apache2-utils
```

- 2) Then, make a copy of the Apache configuration file before editing. For example:

```
sudo cp /etc/apache2/apache2.conf  
/etc/apache2/apache2.conf.orig
```



```
-: sudo su — Konsole  
New Tab Split View  
Copy Paste Find  
Fetched 1,957 kB in 2s (984 kB/s)  
(Reading database ... 486504 files and directories currently installed.)  
Preparing to unpack .../apache2_2.4.58-1_amd64.deb ...  
Unpacking apache2 (2.4.58-1) over (2.4.57-2) ...  
Preparing to unpack .../apache2-bin_2.4.58-1_amd64.deb ...  
Unpacking apache2-bin (2.4.58-1) over (2.4.57-2) ...  
Preparing to unpack .../apache2-data_2.4.58-1_all.deb ...  
Unpacking apache2-data (2.4.58-1) over (2.4.57-2) ...  
Preparing to unpack .../apache2-utils_2.4.58-1_amd64.deb ...  
Unpacking apache2-utils (2.4.58-1) over (2.4.57-2) ...  
Setting up apache2-bin (2.4.58-1) ...  
Setting up apache2-data (2.4.58-1) ...  
Setting up apache2-utils (2.4.58-1) ...  
Setting up apache2 (2.4.58-1) ...  
Installing new version of config file /etc/apache2/mods-available/ssl.load ...  
Installing new version of config file /etc/apache2/sites-available/default-ssl.conf ...  
apache2.service is a disabled or a static unit not running, not starting it.  
apache-htcacheclean.service is a disabled or a static unit not running, not starting it.  
Processing triggers for kali-menu (2023.3.3) ...  
Processing triggers for man-db (2.11.2-3) ...  
(root@Zacker)-[/home/zack]  
# sudo cp /etc/apache2/apache2.conf /etc/apache2/apache2.conf.orig
```

- 3) Now for modifying the file:

```
/etc/httpd/conf/httpd.conf
```

- 1) Edit the Apache configuration file using the text editor nano:

```
sudo nano /etc/apache2/apache2.conf
```
- 2) Locate the lines with the directives you want to modify:
 - a) Change the ServerName directive:

```
ServerName www.looneytunes.com:80
```
 - b) Change the ServerAdmin directive:

```
ServerAdmin root@looneytunes.com
```
 - c) Change the ServerSignature directive:

```
ServerSignature EMail
```

```
af723-2143-4d02-9065-74d5d8d199e1
- : sudo su — Konsole

[1/2] /etc/apache2/sites-available/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
ServerName www.looneytunes.com:80
ServerAdmin root@looneytunes.com
ServerSignature EMail

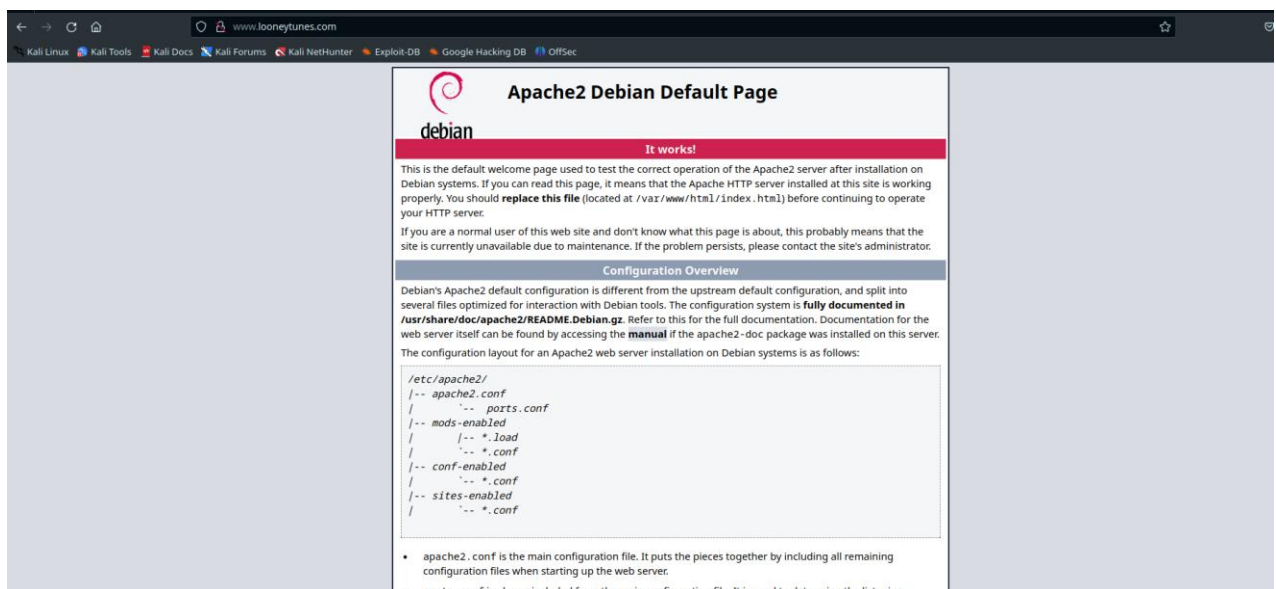
# Available loglevels: trace8, ..., tracel, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
[ 000-default.conf -- 29 lines ]

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

- 3) Save the changes by pressing Ctrl + O, then press Enter to confirm the filename, and exit the text editor by pressing Ctrl + X.
- 4) Start the Apache server using the following command:
sudo service apache2 start
- 5) Go to /etc/hosts
- 6) add www.looneytunes.com in the host file as shown in the screenshot below.



```
- : sudo su — Konsole
New Tab Split View Copy Paste Find
(root@Zacker)~/home/zack
# sudo nano /etc/apache2/sites-available/*
(root@Zacker)~/home/zack
# sudo a2ensite 000-default.conf
Site 000-default already enabled

● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; preset: disabled)
   Active: active (running) since Thu 2023-11-02 20:18:41 IST; 7s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 4272 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 4275 (apache2)
    Tasks: 6 (limit: 4603)
   Memory: 25.3M
      CPU: 58ms
   CGroup: /system.slice/apache2.service
           └─4275 /usr/sbin/apache2 -k start
             └─4277 /usr/sbin/apache2 -k start
               └─4278 /usr/sbin/apache2 -k start
                 └─4279 /usr/sbin/apache2 -k start
                   └─4280 /usr/sbin/apache2 -k start
                     └─4281 /usr/sbin/apache2 -k start

Nov 02 20:18:41 Zacker systemd[1]: Starting apache2.service - The Apache HTTP Server ...
```

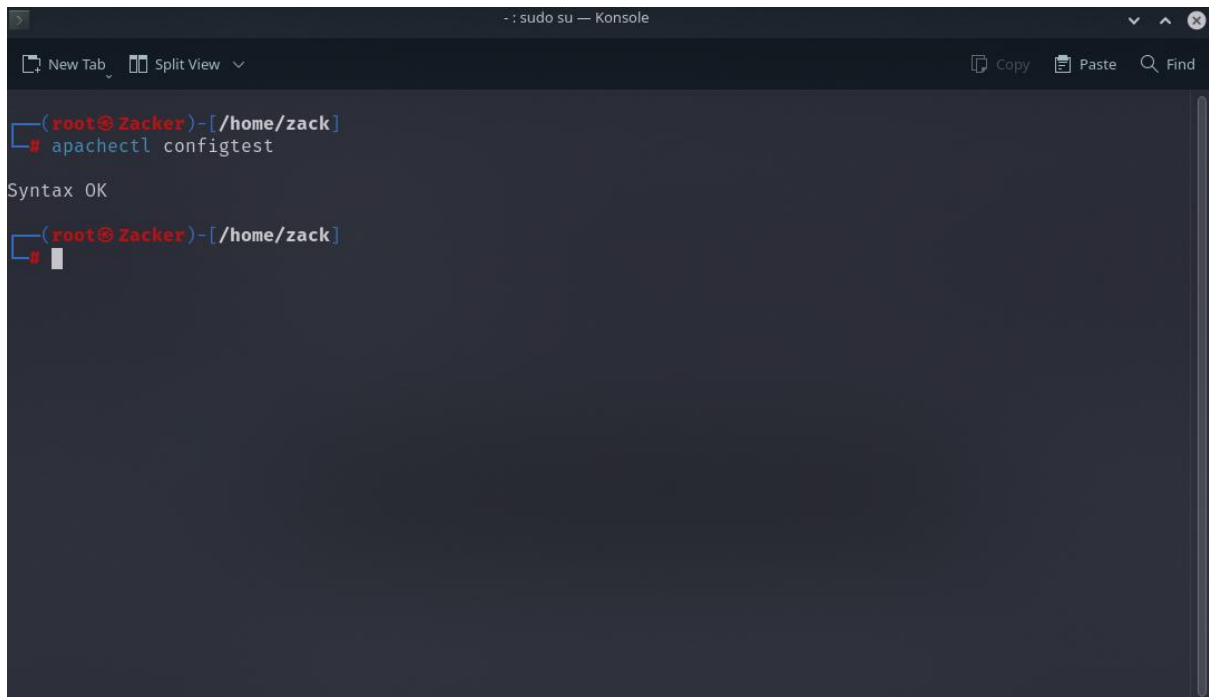
- 7) After starting the server, your modified configurations will take effect. Make sure to verify the status of the Apache service to ensure it is running correctly:
- ```
sudo service apache2 status
```

## TEST APACHE WEB SERVER

In this section, you will test the Apache web server configuration file that you just created.

Test Apache configuration with:

```
apachectl connfigtest
```

A terminal window titled '- : sudo su — Konsole' with a dark background. The prompt is '(root@Zacker)-[/home/zack]'. The command 'apachectl configtest' has been entered and executed, resulting in the output 'Syntax OK'. The prompt is now '(root@Zacker)-[/home/zack]#'. The terminal window has a menu bar with 'New Tab', 'Split View', 'Copy', 'Paste', and 'Find' options.

```
(root@Zacker)-[/home/zack]
apachectl configtest

Syntax OK

(root@Zacker)-[/home/zack]
#
```

## START WEB SERVER

Start the web server with:

```
service httpd start
```

You can make sure that the Apache web server starts at boot with:

```
chkconfig httpd on
```

## CREATE A WEBPAGE

Create a simple web page

In the following folder:

```
/var/www/html
```

Create a new file called:

```
index.html
```

In the file create the following HTML:

```
<!DOCTYPE html>

<html>

<head>

 <title>Welcome to Our Website</title>

</head>

<body>

 <div style="text-align:center; padding: 100px;">

 <h1>Welcome to Our Website</h1>

 <p>We are excited to have you here!</p>

 </div>

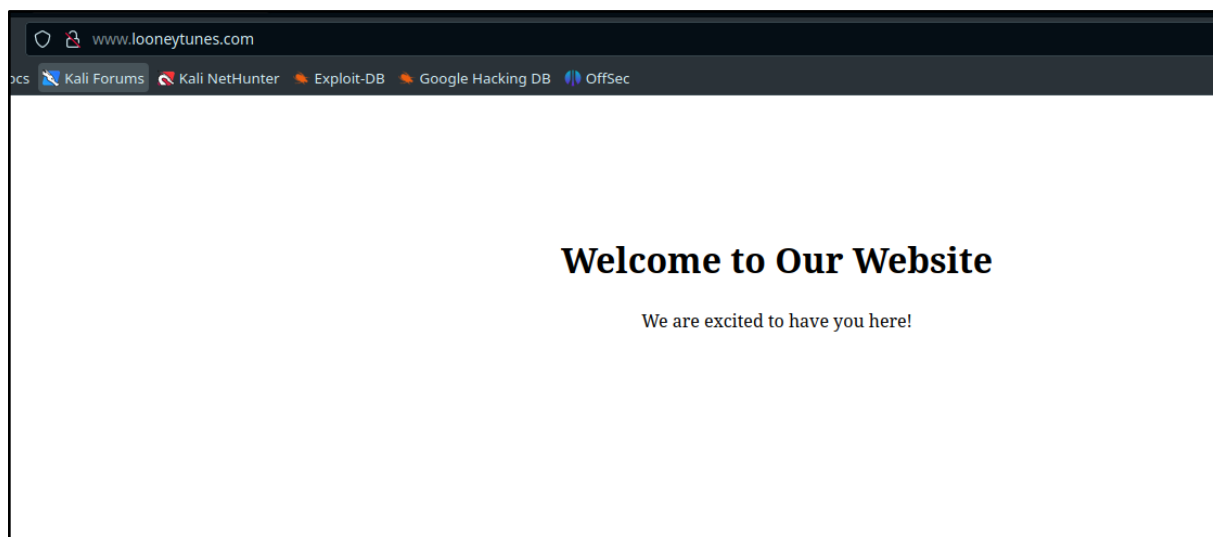
</body>

</html>
```

## TESTING APACHE WEB SERVER WITH A CLIENT:

Test the web server from a client. On your Windows client, start a web browser and go to:

<http://www.looneytunes.com>



## WEB SERVER AUTHENTICATION

In this section, you will create a webpage that requires "Basic Authentication"

Create a new folder:

```
/var/www/html/userauth
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
 <title>Welcome to Our Website</title>
```

```
</head>
```

```
<body>
```

```
 <div style="text-align:center; padding: 100px;">
```

```
 <h1>Welcome Authenticated user!</h1>
```

```
 </div>
```

```
</body>
```

```
</html>
```

**Modify your file at:**

`/var/www/html/index.html`

**to add a link to**

`/var/www/html/userauth/index.html`

**Modified index.html:**

```
<!DOCTYPE html>

<html>

<head>

 <title>Welcome to Our Website</title>

</head>

<body>

 <div style="text-align:center; padding: 100px;">

 <h1>Welcome to Our Website</h1>

 <p>We are excited to have you here!</p>

 Click here to go to
User Authentication

 </div>

</body>

</html>
```



Now you will need to configure "htaccess"

Recall that in the folder:

```
/etc/httpd/conf.d
```

Any file in that folder ending in \*.conf will be loaded as an Apache configuration.

What that means is you can simply create a small configuration file that will be added to main configuration file upon httpd startup.

In our situation, we want to add some special considerations to the folder

```
/var/www/html/userauth
```

Create a new configuration file:

```
/etc/lighttpd/userauth.conf
```

```
<Directory "/var/www/html/userauth">
 Options Indexes FollowSymLinks
 AllowOverride All
 Require all granted
</Directory>
```

In the folder:

```
/var/www/html/userauth
```

make a new file called

```
.htaccess
```

And add the following lines to the .htaccess file:

```
AuthName "Member's Only!"
```

```
AuthUserFile /var/www/html/userauth/.htpasswd
```

```
AuthType Basic
```

```
Require valid-user
```

Now create a user and password combination for htaccess with htpasswd.

In this section, you will create a new .htpasswd file that contains just the userid and encrypted passwd.

You can add users with the htpasswd command.

For example:

```
htpasswd -c /var/www/html/userauth/.htpasswd test
```

Set the password to

```
P@ssw0rd
```

After you have created the htaccess and htpasswd files, httpd needs to be restarted.

You can test if the configuration is working by restarting httpd.

Be sure to restart your client web browser also.

Attempt to browse to the "Authenticated Users Only" link. It should ask for credentials, and you should see something similar:

Enter the following credentials:

Username: test

Password: P@ssw0rd

## WEB SERVER - LOG FILES

Web server log files are an invaluable tool for both diagnosing issues and for creating statistics on web site and web page client use.

Review the Apache log files at:

```
/var/log/httpd/access_log
```

```
/var/log/httpd/error_log
```

### **Q: What is / describe the directive: Listen**

Description: Defines the IP addresses and ports that the Apache server listens to for incoming connections.

**Q: What is / describe the directive: ServerName**

Description: Sets the hostname and port that the server uses to identify itself. It is often used when the server serves multiple hostnames.

**Q: What is / describe the directive: ServerAdmin**

Description: Sets the email address of the server administrator. This address appears in various error messages sent by the server.

**Q: What is / describe the directive: ServerSignature**

Description: Determines whether the server includes a footer on server-generated documents, containing the server name and version number.

**Q: What is / describe the directive: DirectoryIndex**

Description: Sets the file that the server will serve if a directory is requested by the client. It serves the specified file when no specific file is requested.

**Q: What is / describe the directive: DocumentRoot**

Description: Specifies the directory that forms the main document tree visible from the web. It is the directory from which Apache serves files by default.

**Q: What is / describe the directive: Include conf.d/\*.conf**

Description: Includes all configuration files with the ".conf" extension in the "conf.d" directory. This directive allows for the modular organization of configuration files.

**Q: What other ways can you check to see if the web server is running?**

- Running a command to check the service status, such as `systemctl status apache2` or `service apache2 status`.
- Checking the server's status through the Apache web server's status page.

**Q: What web page comes up?**

This depends on the specific content of the `index.html` file in the `DocumentRoot` directory. It could be a default Apache page or a custom web page if one has been created.

**Q: What does the directive "AllowOverride All" mean?**

This directive allows the use of the .htaccess file to override global configuration settings for the specific directory and its subdirectories.

**Q: What is the purpose of an .htaccess file?**

An .htaccess file is used to configure directory-specific settings for an Apache web server. It allows for decentralized management of configuration for a specific directory and its subdirectories.

**Q: How can you restart httpd?**

You can restart the httpd service using the command `systemctl restart apache2` or `service apache2 restart`, depending on your system's configuration.

**Q: What page is now displayed?**

This depends on the configuration and content of the index.html file in the DocumentRoot directory. If the configuration has been set up properly, it should display the custom welcome page or the default Apache page.

