



## 4. Web Server Administration

Miquel Àngel París i Peñaranda

Web Application Deployment

2nd C-VET Web Application Development



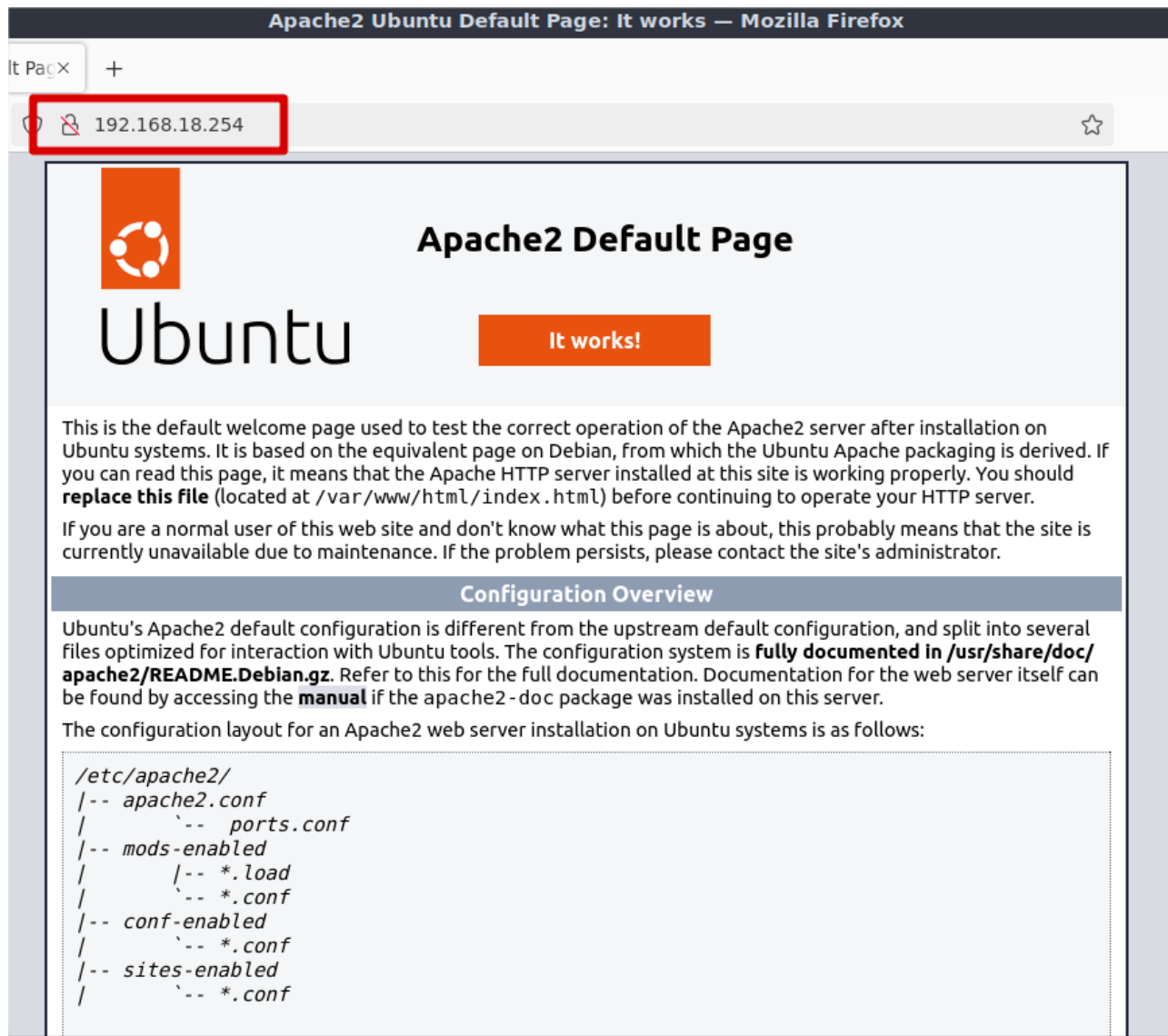
**Index**

Apache exercises .....3

## Apache exercises

### Step 1: Install the Apache2 Server

Verify that Apache2 is installed and running by opening a browser and navigating to <http://localhost>. Attach screenshots.



### Step 2: Check the Version

Use the command `apache2 -v` to check the installed version. Attach a screenshot.

```
root@server:/home/user# apache2 -v
Server version: Apache/2.4.52 (Ubuntu)
Server built:   2024-07-17T18:57:26
```

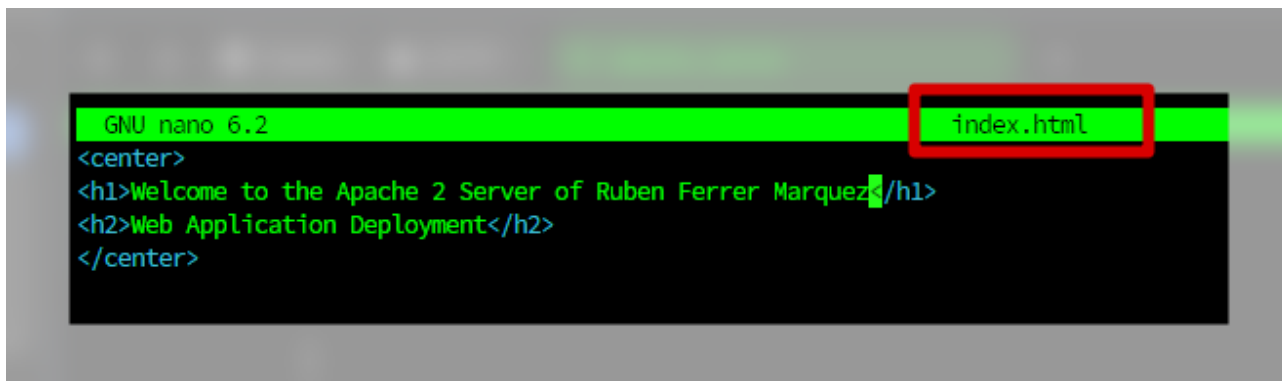
### Step 3: Create a Welcome Page

Replace the default welcome page with a custom one:

- Navigate to `/var/www/html/`.
- Rename the existing `index.html` file.
- Create a new `index.html` file with the following content:

```
<center>
<h1>Welcome to the Apache 2 Server of [Your Name]</h1>
<h2>Web Application Deployment</h2>
</center>
```

Replace `[Your Name]` with your own name. Refresh the browser, and the new page should appear. Attach screenshots.

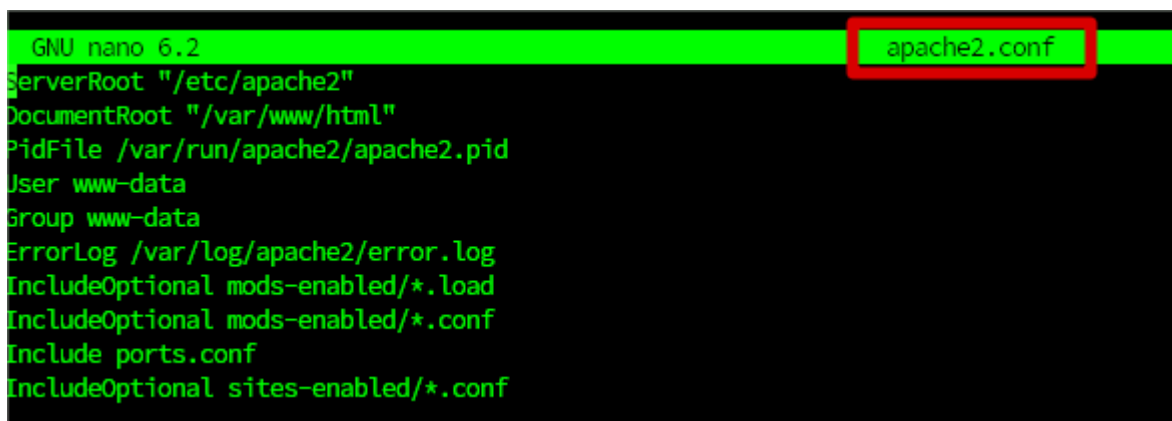


#### Step 4: Update Apache2 Configuration File

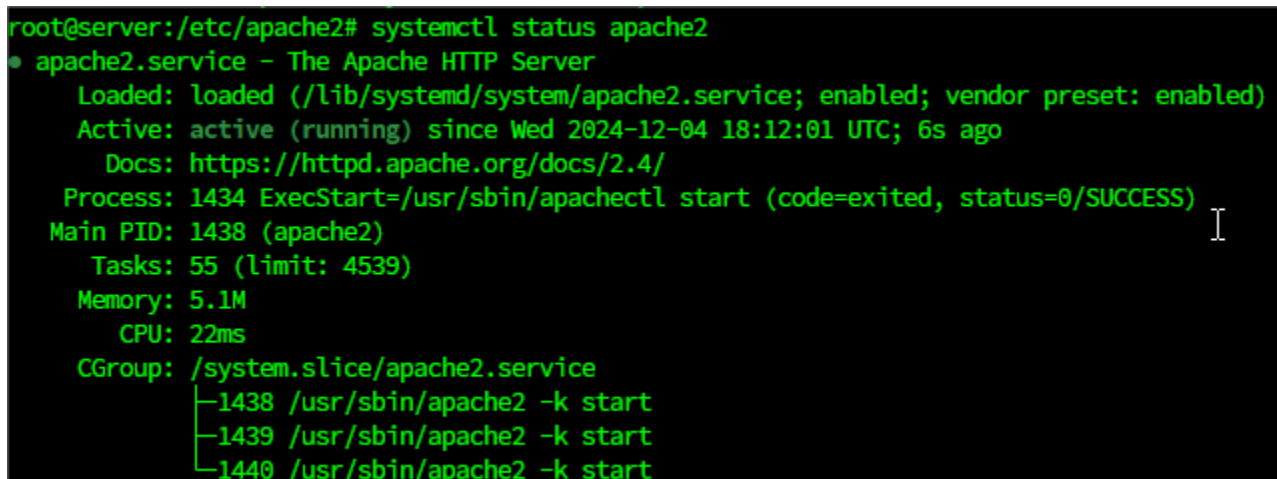
- Make a backup of `/etc/apache2/apache2.conf`.
- Add the following lines to the configuration file:

```
# Basic Configuration File (/etc/apache2/apache2.conf)
ServerRoot "/etc/apache2"
DocumentRoot "/var/www/html"
PidFile /var/run/apache2/apache2.pid
User www-data
Group www-data
ErrorLog /var/log/apache2/error.log
IncludeOptional mods-enabled/*.load
IncludeOptional mods-enabled/*.conf
Include ports.conf
IncludeOptional sites-enabled/*.conf
```

Restart Apache2 and verify its functionality. Attach screenshots.



```
GNU nano 6.2 apache2.conf
ServerRoot "/etc/apache2"
DocumentRoot "/var/www/html"
PidFile /var/run/apache2/apache2.pid
User www-data
Group www-data
ErrorLog /var/log/apache2/error.log
IncludeOptional mods-enabled/*.load
IncludeOptional mods-enabled/*.conf
Include ports.conf
IncludeOptional sites-enabled/*.conf
```



```
root@server:/etc/apache2# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-12-04 18:12:01 UTC; 6s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 1434 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 1438 (apache2)
    Tasks: 55 (limit: 4539)
   Memory: 5.1M
      CPU: 22ms
   CGroup: /system.slice/apache2.service
           └─1438 /usr/sbin/apache2 -k start
             └─1439 /usr/sbin/apache2 -k start
               └─1440 /usr/sbin/apache2 -k start
```

## Unit 4. Web Server Administration



### Step 5: Change the Server's Access Name

Change the access name of the server to `www.eihsa.es` using either DNS configuration or by editing the `/etc/hosts` file. Test that the server responds to the new name. Attach screenshots.



## Step 6: Enable User Web Spaces

Enable Apache's module that allows users to host their web pages in personal directories:

- Run: `$ sudo a2enmod userdir`

```
root@server:/etc/apache2# a2enmod userdir
Module userdir already enabled
root@server:/etc/apache2# systemctl restart apache2
root@server:/etc/apache2#
```

- Each user can create a `public_html` folder in their home directory (e.g., `/home/username/public_html`).

```
GNU nano 6.2 mods-enabled/userdir.conf
<IfModule mod_userdir.c>
    UserDir public_html
    UserDir disabled root

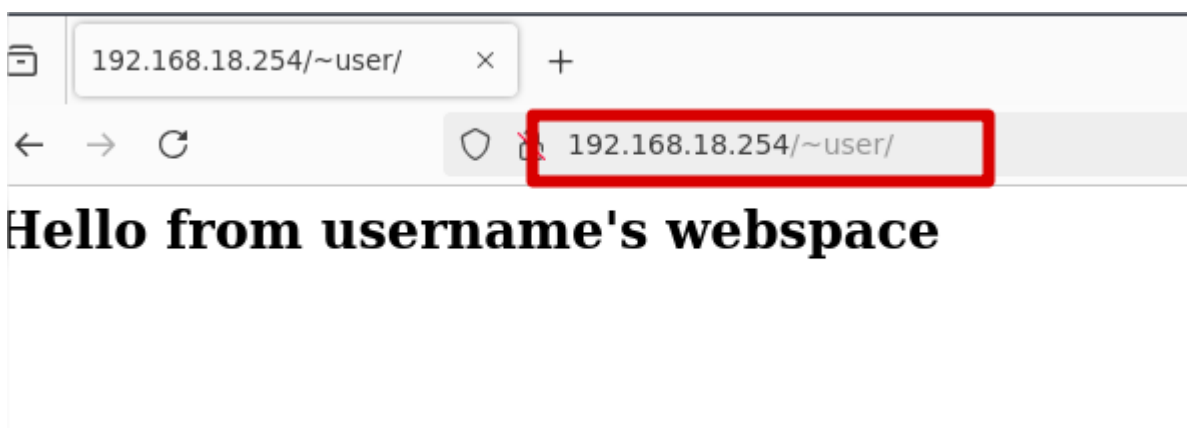
    <Directory /home/*/public_html>
        AllowOverride FileInfo AuthConfig Limit
        Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
        Require all granted
    </Directory>
</IfModule>

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

Set appropriate permissions:

- Directory: `chmod 755`
- Files: `chmod 644`

Users can access their pages via URLs like `http://server_address/~username`. Restart Apache2 and test. Attach screenshots.





### Step 7: Define Virtual Hosts by Name

- Create a new configuration file in `/etc/apache2/sites-available/` for the new host.

```
root@server:/etc/apache2/sites-available# ls
000-default.conf 001-eihsa.conf default-ssl.conf
root@server:/etc/apache2/sites-available#
```

- Set up the directory and create a web page for the host.

```
GNU nano 6.2 001-eihsa.conf
<VirtualHost www.eihsa.es>
    # 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.eihsa.es

    ServerAdmin webmaster@eihsa.es
    DocumentRoot /var/www/html/eihsa

    # 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 /tmp/eihsa_ERROR.log
    #CustomLog ${APACHE_LOG_DIR}/access.log combined
    TransferLog /tmp/eihsa_ACCESS.log
    # 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
```

- Enable the site using `sudo a2ensite`.

```
root@server:/etc/apache2/sites-available# a2ensite 001-eihsa.conf
Enabling site 001-eihsa.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@server:/etc/apache2/sites-available# systemctl reload apache2
root@server:/etc/apache2/sites-available#
```

- Restart Apache2 and verify functionality. Attach screenshots.

```
root@server:/var/www/html# systemctl restart apache2
root@server:/var/www/html#
```



## Step 8: Define Virtual Hosts by IP

Configure a virtual host accessible via IP address. Test and attach screenshots.

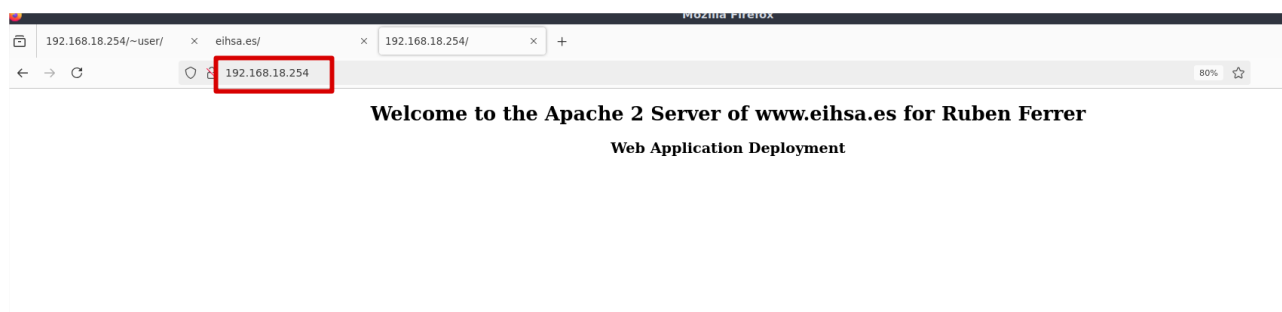
```
GNU nano 6.2                                001-eihsa.conf
VirtualHost 192.168.18.100>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# 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.eihsa.es

ServerAdmin webmaster@eihsa.es
DocumentRoot /var/www/html/eihsa

# 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 /tmp/eihsa_ERROR.log
#CustomLog ${APACHE_LOG_DIR}/access.log combined
TransferLog /tmp/eihsa_ACCESS.log
# 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
```



## Step 9: Define Virtual Hosts by IP and Port

Set up a virtual host accessible via a specific IP and port. Test and attach screenshots.

```
GNU nano 6.2 ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80
Listen 1111
<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

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

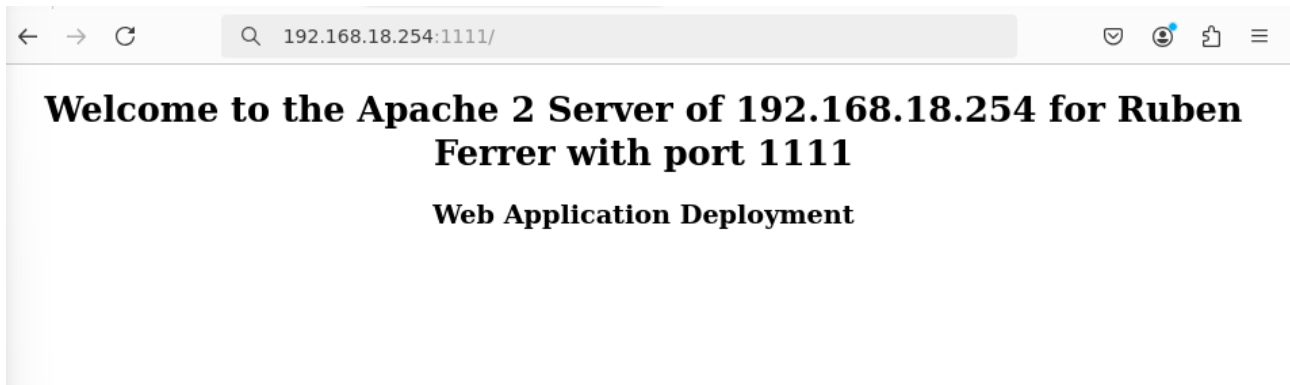
```
GNU nano 6.2 001-eihsa.conf
VirtualHost 192.168.18.100:1111
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what 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.eihsa.es

    ServerAdmin webmaster@eihsa.es
    DocumentRoot /var/www/html/eihsa

    # 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 /tmp/eihsa_ERROR.log
    #CustomLog ${APACHE_LOG_DIR}/access.log combined
    TransferLog /tmp/eihsa_ACCESS.log
    # 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
```



## Step 10: Basic HTTP Authentication and Protected Directories

Add authentication to a directory on the virtual server using `mod_auth_basic`.

Steps to Implement:

1. Create a User

Use the `htpasswd` command to create a `.htpasswd` file:

```
sudo su
htpasswd -c /etc/apache2/passwd/.htpasswd username
root@server:/etc/apache2# mkdir passwd
root@server:/etc/apache2# cd passwd/
root@server:/etc/apache2/passwd# htpasswd -c /etc/apache2/passwd/.htpasswd ruben
New password:
Re-type new password:
Adding password for user ruben
root@server:/etc/apache2/passwd#
```

Set permissions on `.htpasswd` to `chmod 644`.

2. Restrict Access to a Private Directory

Create the directory `/var/www/html/virtual/private_directory` and add an `index.html` file. Add the following configuration to the relevant site file:

```
<Directory "/var/www/html/virtual/private_directory">
    AuthType Basic
    AuthName "Private Directory"
    AuthUserFile /etc/apache2/passwd/.htpasswd
    Require valid-user
</Directory>
```

```
GNU nano 6.2 /var/www/html/virtual/private_directory/index.html
<h1>archivo de private_directory</h1>
```

```
GNU nano 6.2 private_site.conf
<VirtualHost www.priveihsa.es>

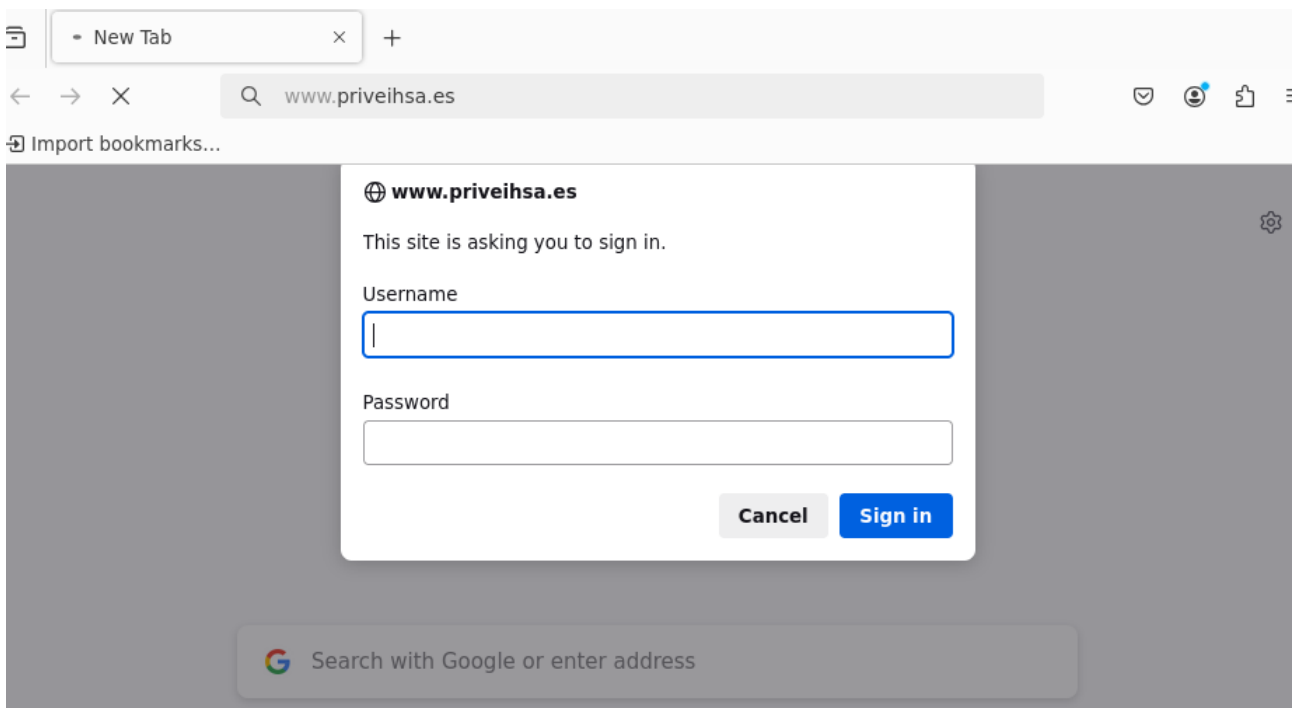
    ServerName www.priveihsa.es

    ServerAdmin webmaster@priveihsa.es
    DocumentRoot /var/www/html/virtual/private_directory/

    ErrorLog /tmp/www_ERROR.log

    TransferLog /tmp/www_ACCESS.log


    <Directory "/var/www/html/virtual/private_directory">
        AuthType Basic
        AuthName "Private Directory"
        AuthUserFile /etc/apache2/passwd/.htpasswd
        Require valid-user
    </Directory>
</VirtualHost>
```





Restart Apache2 and test access via a browser.

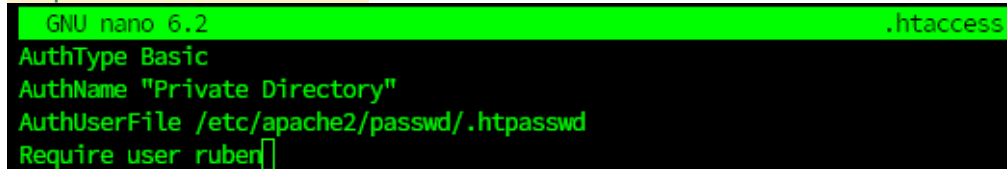


## Unit 4. Web Server Administration

### 3. Using .htaccess for Authentication

Create a .htaccess file in the directory:

```
AuthType Basic
AuthName "Private Directory"
AuthUserFile /etc/apache2/passwd/.htpasswd
Require user username
```



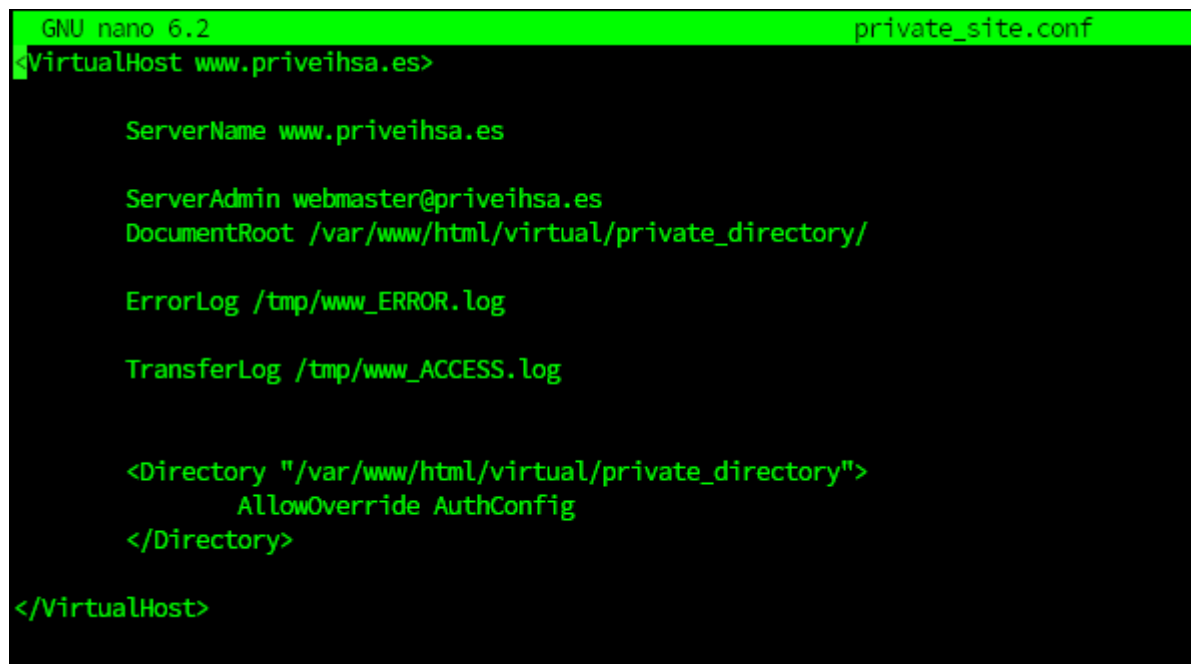
A screenshot of a terminal window with a green title bar that reads "GNU nano 6.2" and ".htaccess". The terminal shows the following configuration for basic authentication:

```
AuthType Basic
AuthName "Private Directory"
AuthUserFile /etc/apache2/passwd/.htpasswd
Require user ruben
```

Enable .htaccess in the site configuration by adding:

```
<Directory "/var/www/html/virtual/private_directory">
    AllowOverride AuthConfig
</Directory>
```

Test functionality without restarting Apache2.



A screenshot of a terminal window with a green title bar that reads "GNU nano 6.2" and "private\_site.conf". The terminal shows the configuration for a virtual host:

```
<VirtualHost www.priveihsa.es>

    ServerName www.priveihsa.es

    ServerAdmin webmaster@priveihsa.es
    DocumentRoot /var/www/html/virtual/private_directory/

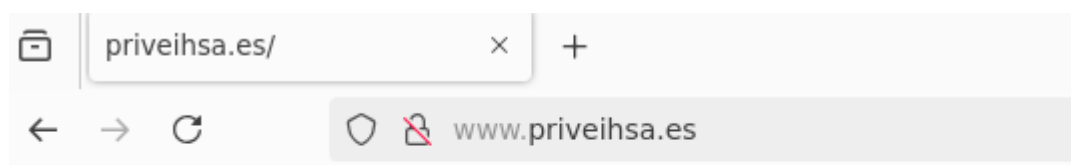
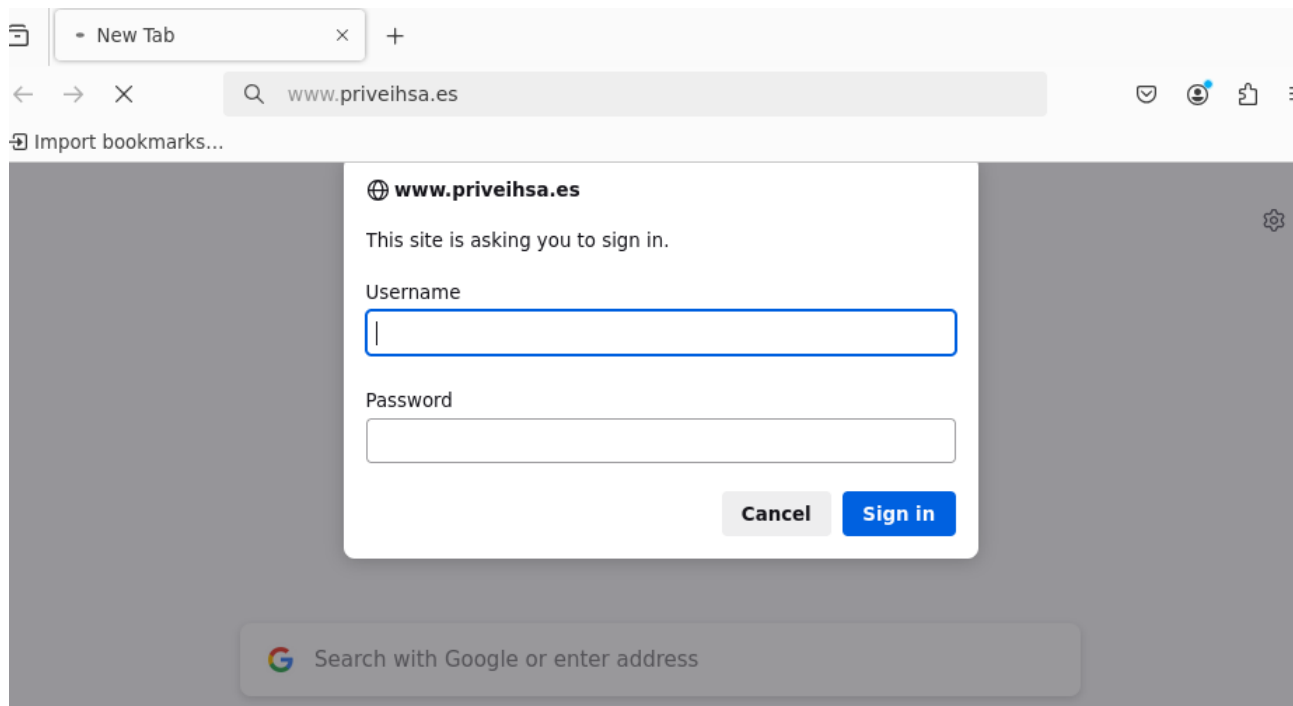
    ErrorLog /tmp/www_ERROR.log

    TransferLog /tmp/www_ACCESS.log

    <Directory "/var/www/html/virtual/private_directory">
        AllowOverride AuthConfig
    </Directory>

</VirtualHost>
```

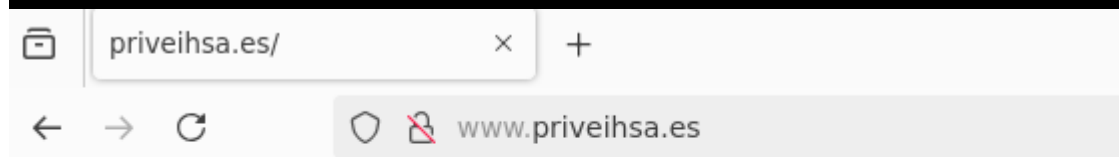
## Unit 4. Web Server Administration



# archivo de private\_directory

### 4. Digest Authentication

Enable the `mod_auth_digest` module for encrypted authentication. Configure using the `htdigest` tool to add users and allow access to the protected directory.



# archivo de private\_directory

### Step 11: Create a Secure Virtual Server with OpenSSL

Set up a secure virtual server using OpenSSL. Test and attach screenshots.

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048
```

```
GNU nano 6.2 private_site.conf
<VirtualHost *:443>
    ServerName www.eihsa.es
    DocumentRoot /var/www/secure

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
    SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

    <Directory /var/www/secure>
        AllowOverride All
    </Directory>
</VirtualHost>
```

## Unit 4. Web Server Administration

