4. Web Server Administration

Miquel Àngel París i Peñaranda

Web Application Deployment

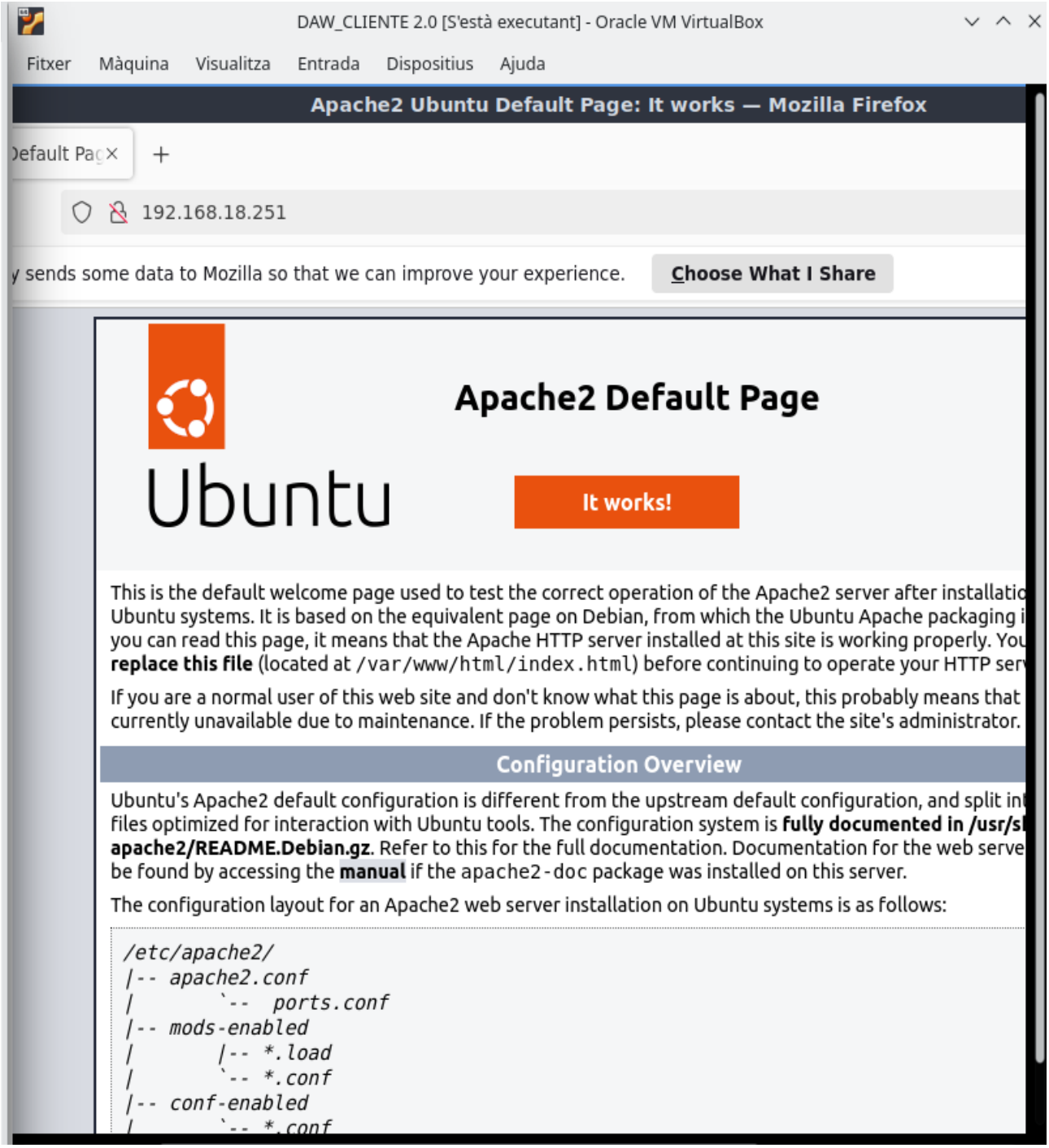
2nd C-VET Web Application Development

Index

Apache exercises

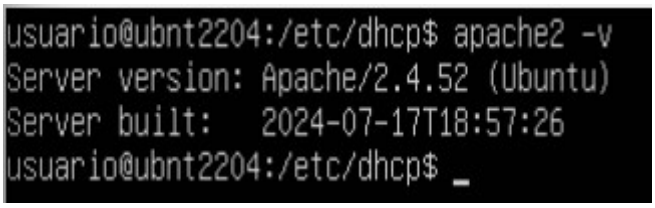
Step 1: Install the Apache2 Server

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.



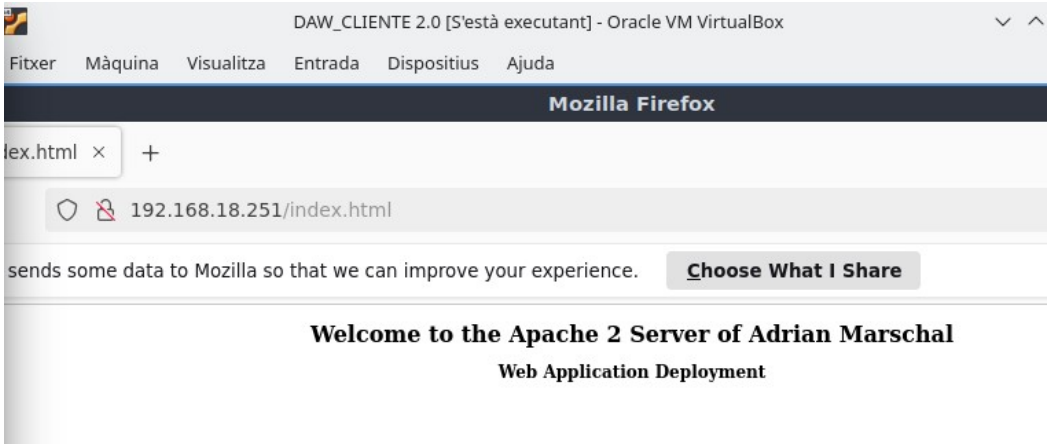
Step 3: Create a Welcome Page

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.

```
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 list of ports to listen on
Include ports.conf
IncludeOptional sites-enabled/*.conf
```


```
usuario@ubnt2204:/etc/apache2$ sudo systemctl restart apache2
usuario@ubnt2204:/etc/apache2$ sudo 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 2024-11-27 16:38:21 UTC; 5s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 1788 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 1792 (apache2)
      Tasks: 55 (limit: 2225)
     Memory: 4.7M
        CPU: 17ms
     CGroup: /system.slice/apache2.service
             ├─1792 /usr/sbin/apache2 -k start
             ├─1793 /usr/sbin/apache2 -k start
             └─1794 /usr/sbin/apache2 -k start

nov 27 16:38:21 ubnt2204 systemd[1]: apache2.service: Deactivated successfully.
nov 27 16:38:21 ubnt2204 systemd[1]: Stopped The Apache HTTP Server.
nov 27 16:38:21 ubnt2204 systemd[1]: Starting The Apache HTTP Server...
nov 27 16:38:21 ubnt2204 apachectl[1791]: AH00558: apache2: Could not reliably determine the server
nov 27 16:38:21 ubnt2204 systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
```

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.

```
  GNU nano 6.2                                    hosts
127.0.0.1 localhost
127.0.1.1 ubnt2204
192.168.18.251 www.eihsa.es
```

Step 6: Enable User Web Spaces

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

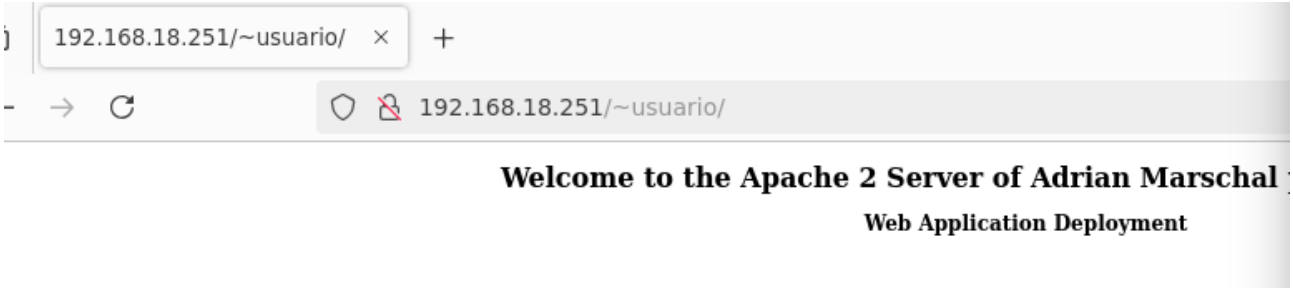
- Run: `$ sudo a2enmod userdir`

```
usuario@ubnt2204:/etc$ sudo a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
  systemctl restart apache2
usuario@ubnt2204:/etc$ sudo systemctl restart apache2
usuario@ubnt2204:/etc$
```

- Each user can create a `public_html` folder in their home directory (e.g., `/home/username/public_html`). Set appropriate permissions:
- Directory: `chmod` `755`
- Files: `chmod 644`

```
<IfModule mod_userdir.c>
        UserDir public_html

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

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.cd

```
usuario@ubnt2204:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf  eihsa-es.conf
```

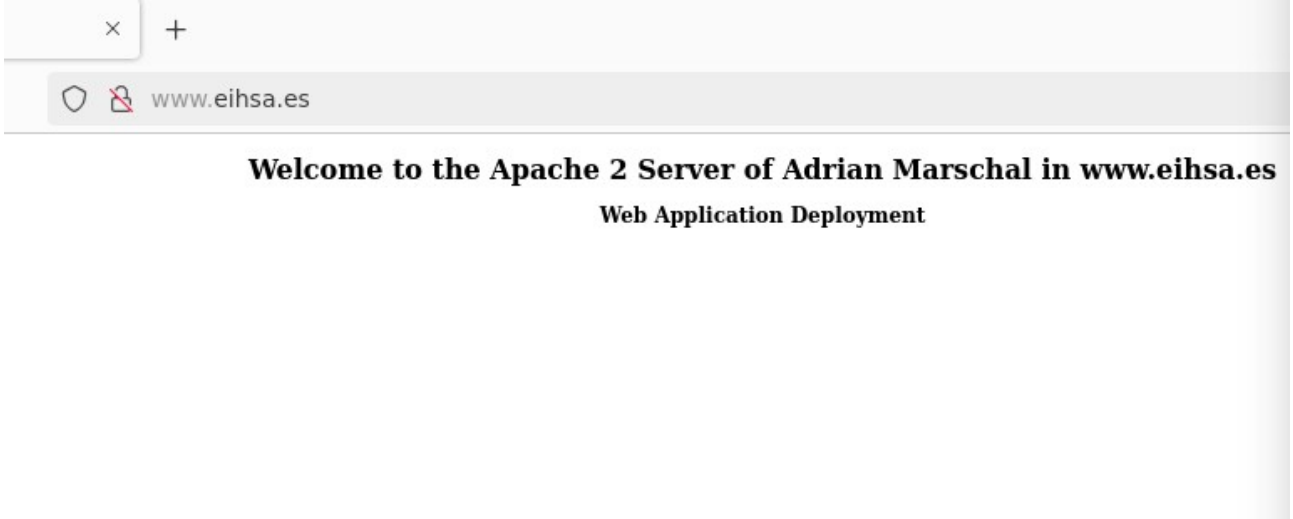
- Set up the directory and create a web page for the host.

```
  GNU nano 6.2                    /etc/apache2/sites-available/eihsa.conf

ServerRoot "/etc/apache2"
DocumentRoot "/var/www"
PidFile /var/run/apache2/apache2.pid
User www-data
Group www-data
ErrorLog /var/log/apache2/error.log
TypesConfig /etc/mime.types


<VirtualHost www.eihsa.es: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@eihsa.es
        ServerName www.eisha.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/www_ERROR.log
        TransferLog /tmp/www_ACCESS.log
```

- Enable the site using `sudo a2ensite`.

```
usuario@ubnt2204:/etc/apache2/sites-available$ sudo a2ensite eihsa-es.conf
Site eihsa-es already enabled
```

- Restart Apache2 and verify functionality. Attach screenshots.



Step 8: Define Virtual Hosts by IP

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

```
  GNU nano 6.2                                  cliente100.conf

ServerRoot "/etc/apache2"
DocumentRoot "/var/www"
PidFile /var/run/apache2/apache2.pid
User www-data
Group www-data
ErrorLog /var/log/apache2/error.log
TypesConfig /etc/mime.types


<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.example.com

        ServerName www.eisha.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/www_ERROR.log
        TransferLog /tmp/www_ACCESS.log
```

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.

```
usuario@ubnt2204:/etc/apache2/sites-available$ sudo a2ensite socketCliente.conf
Enabling site socketCliente.
To activate the new configuration, you need to run:
  systemctl reload apache2
usuario@ubnt2204:/etc/apache2/sites-available$ systemctl reload apache2.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to reload 'apache2.service'.
Authenticating as: usuario
Password:
==== AUTHENTICATION COMPLETE ===
```

```
<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.example.com
        ServerName www.eisha.es
        DocumentRoot /var/www/html/eihsa/porSKT

        # 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/www_ERROR.log
        TransferLog /tmp/www_ACCESS.log

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
```

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
   ```

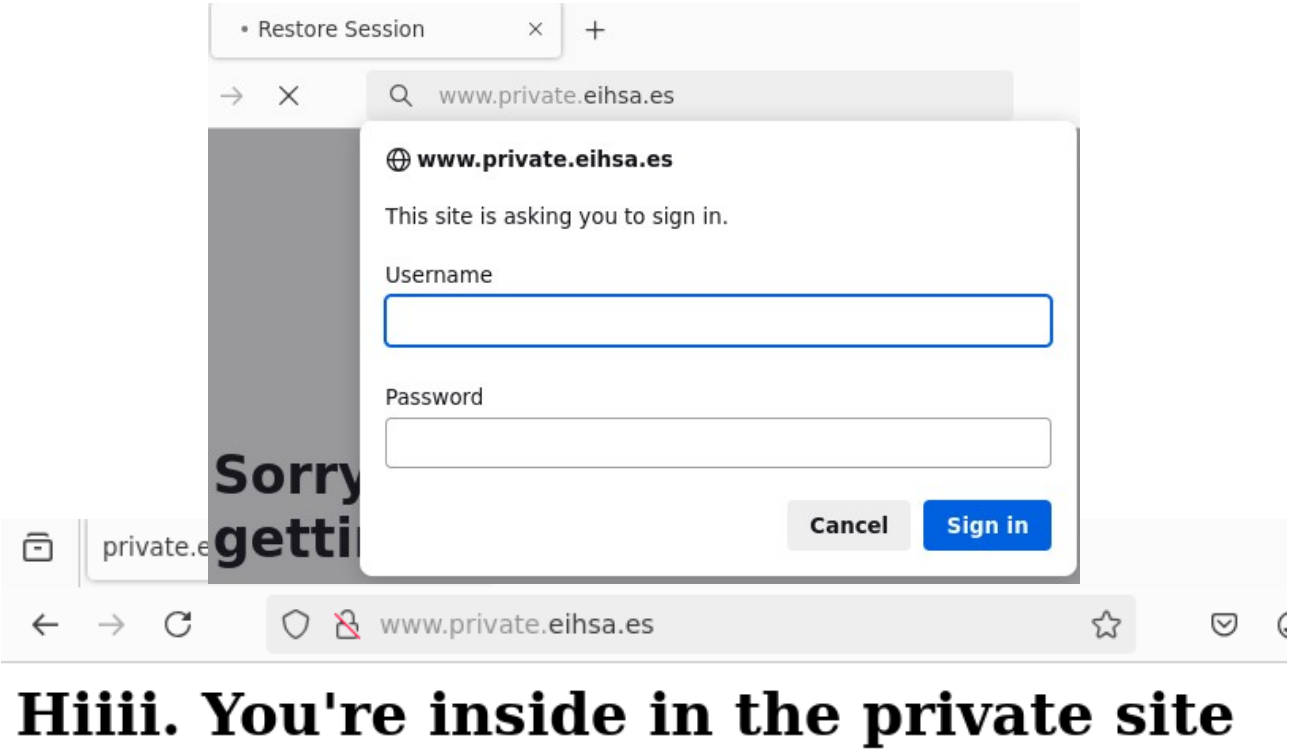
   Set permissions on `.htpasswd` to `chmod 644`.

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>
   ```

   Restart Apache2 and test access via a browser.



3. Using `.htaccess` for Authentication

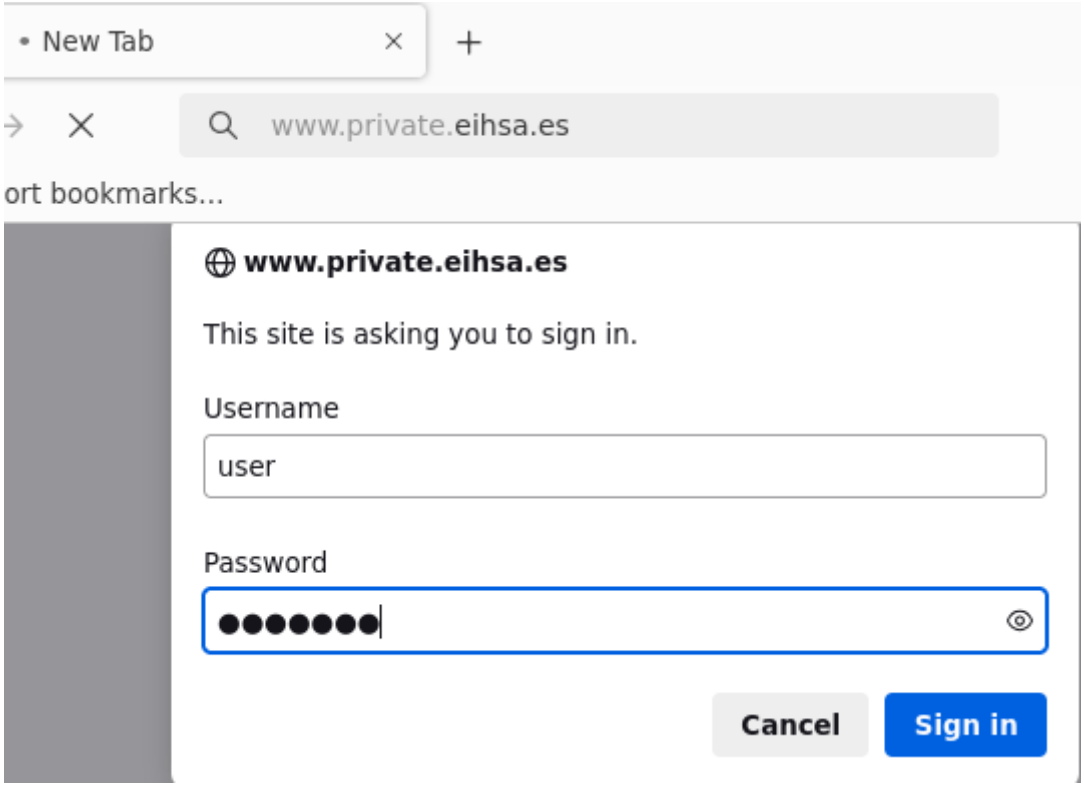
   Create a `.htaccess` file in the directory:

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

   Enable `.htaccess` in the site configuration by adding:

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

   Test functionality without restarting Apache2.



The result it's the same because the changes of the lines inside of Directory are saved in the file .htaccess and the order AllowOverride AuthConfig takes this file and apply the config

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.

   `htdigest -c /etc/apache2/passwd/.htdigest user`

Htaccess_VirtualHost:

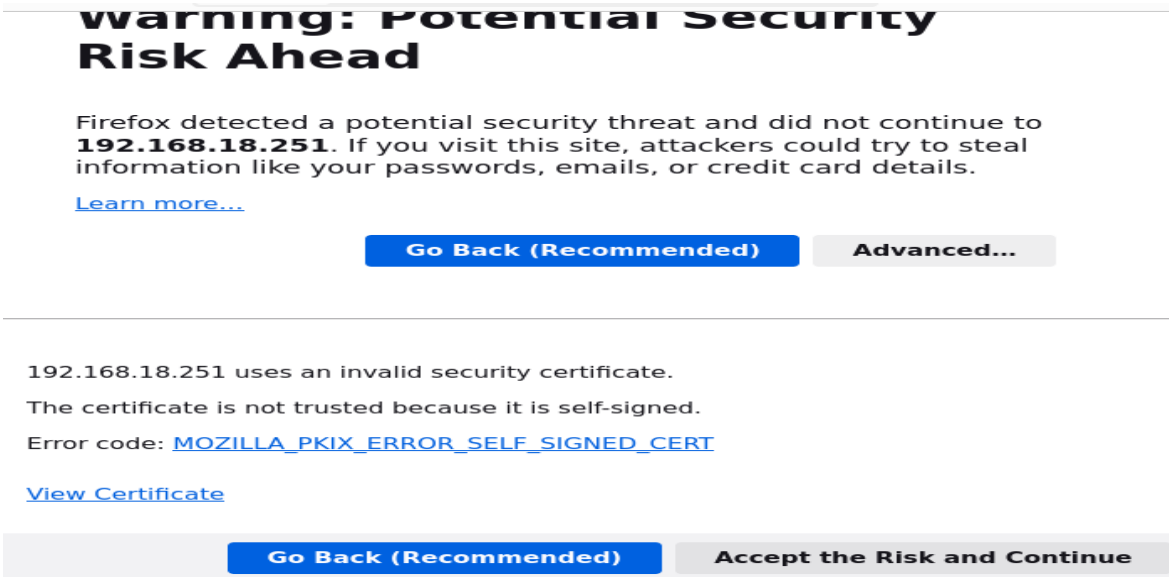
```
AuthType Digest
AuthName "usuario"
AuthDigestDomain /private_directory/
AuthUserFile /etc/apache2/passwd/.htdigest
Require valid-user
```

Step 11: Create a Secure Virtual Server with OpenSSL

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

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

```
<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>
```


Warning: Potential Security Risk Ahead
Firefox detected a potential security threat and did not continue to 192.168.18.251. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.
Learn more...
Go Back (Recommended)
Advanced...
192.168.18.251 uses an invalid security certificate.
The certificate is not trusted because it is self-signed.
Error code: MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT
View Certificate
Go Back (Recommended)
Accept the Risk and Continue


192.168.18.251/
https://192.168.18.251
Bienvenido en modo seguro al Server de Adrian Marschal