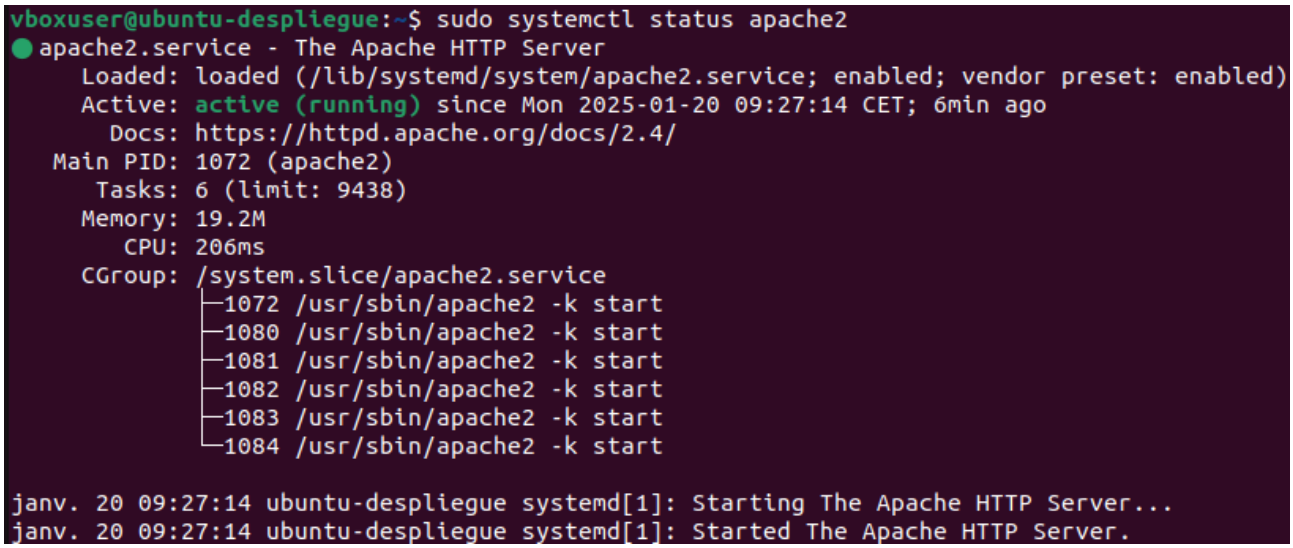
Aitor Pascual Jiménez
Despliegue de Aplicaciones Web
2º DAW

CERTIFICADOS EN APACHE

Aitor Pascual Jiménez
Despliegue de Aplicaciones Web
2º DAW

1. REQUISITOS PREVIOS

Para esta práctica se da por hecho que entendemos Apache y este está instalado en nuestro linux, podemos comprobar esto lanzando un systemctl.



Aitor Pascual Jiménez
Despliegue de Aplicaciones Web
2º DAW

2. HABILITAR SSL Y CREAR CLAVE

Para habilitar SSL en nuestro apache tenemos que usar la directiva a2enmod, esta herramienta significa apache2 enable module, como el módulo que vamos a habilitar es SSL tenemos que hacer a2enmod ssl (en caso de querer deshabilitar SSL tenemos a2dismod que hace alusión a apache2 disable module).

```
vboxuser@ubuntu-despliegue:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

Si reiniciamos vemos que la directiva se aplica correctamente.

```
vboxuser@ubuntu-despliegue:~$ sudo systemctl restart apache2
vboxuser@ubuntu-despliegue:~$ 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 Mon 2025-01-20 09:51:12 CET; 5s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 3732 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 3738 (apache2)
      Tasks: 6 (limit: 9438)
     Memory: 13.1M
        CPU: 68ms
     CGroup: /system.slice/apache2.service
             ├─3738 /usr/sbin/apache2 -k start
             ├─3739 /usr/sbin/apache2 -k start
             ├─3740 /usr/sbin/apache2 -k start
             ├─3741 /usr/sbin/apache2 -k start
             ├─3742 /usr/sbin/apache2 -k start
             └─3743 /usr/sbin/apache2 -k start

janv. 20 09:51:12 ubuntu-despliegue systemd[1]: Starting The Apache HTTP Server...
janv. 20 09:51:12 ubuntu-despliegue systemd[1]: Started The Apache HTTP Server.
```

Aitor Pascual Jiménez
Despliegue de Aplicaciones Web
2º DAW

Ahora bien, tener habilitado el módulo no sirve de nada si no tenemos un certificado, dentro de los archivos de configuración de apache, vamos a crear un diretorio en el que almacenaremos nuestras claves.

```
vboxuser@ubuntu-despliegue:/etc/apache2$ sudo mkdir MiVirtualHost
vboxuser@ubuntu-despliegue:/etc/apache2$ sudo mkdir MiVirtualHost/ssl
```

Dentro de este directorio vamos a usar la herramienta openssl que nos permite generar nuestros certificados, como parámetros vamos a usar genrsa que indica que se generará una clave privada RSA -des3 para que nuestra clave esté protegida mediante una contraseña (en este caso cifrada por ttiple DES) -out para definir el nombre del archivo de salida y 2048 la longitud de bits de nuestra clave.

```
vboxuser@ubuntu-despliegue:/etc/apache2/MiVirtualHost/ssl$ sudo openssl genrsa -des3 -out key 2048
Enter PEM pass phrase:                  contraseña
Verifying - Enter PEM pass phrase:      confirmación de contraseña
```

Con nuestra clave privada ya generada, vamos a a generar otro archivo, este será un csr (Certificate Signing Request) que es un archivo que contiene información nuestra (en este caso parcialmente inventada) para solicitar un certificado SSL/TLS a una entidad certificadora, para generar este archivo volveremos a usar openssl pero esta vez con req para manejar csr -new porque estamos generando uno nuevo, -key para especificar la clave privada que vamos a usar y -out para el nombre del archivo.

```
vboxuser@ubuntu-despliegue:/etc/apache2/MiVirtualHost/ssl$ sudo openssl req -new -key key -out MiVirtualHost.csr
Enter pass phrase for key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Madrid
Locality Name (eg, city) []:Madrid
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Aitor Internetes
Organizational Unit Name (eg, section) []:Aitor Internetes
Common Name (e.g. server FQDN or YOUR name) []:aitor
Email Address []:aitor.pascual@educa.madrid.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:changeme
An optional company name []:Aitor Internetes
```

Contraseña definida previamente para nuestra clave privada

información para la entidad

Aitor Pascual Jiménez
Despliegue de Aplicaciones Web
2º DAW

Como en este caso estamos en una prueba, no vamos a mandar nuestro csr a una entidad (y no vamos a pagar por ello), vamos a auto firmarnos nuestro certificado, esto nos permitirá usar https aunque los navegadores no nos reconozcan nuestro certificado como uno fiable.

Para ello vamos a volver a utilizar openssl con los parámetros x509 para indicar que trabajamos con un certificado X.509 que es un formato estándar para claves públicas, con -req le decimos que el archivo especificado posteriormente en -in, es un archivo .csr, -sha256 es el algoritmo de hashing que vamos a utilizar (sha de 256 bytes), usamos -days para especificar la validez en días de nuestro certifcado, en este caso un año, -signkey nos permite especificar la clave privada utilizada para firmar nuestro certificado y -out para especificar el archivo de salida.

```
vboxuser@ubuntu-despliegue:/etc/apache2/MiVirtualHost/ssl$ sudo openssl x509 -req -sha256 -days 365 -in MiVirtualHost.csr -signkey key -out MiVirtualHost.crt
[sudo] contraseña para vboxuser:
Enter pass phrase for key:
Certificate request self-signature ok
subject=C = ES, ST = Madrid, L = Madrid, O = Aitor Internetes, OU = Aitor Internetes, CN = aitor, emailAddress = aitor.pascual@educa.madrid.org
```

Aitor Pascual Jiménez
Despliegue de Aplicaciones Web
2º DAW

3. VIRTUAL HOST Y DESPLIEGUE

En el directorio de sites-available copiamos y editamos la plantilla 000-default.conf.

```
vboxuser@ubuntu-despliegue:/etc/apache2/sites-available$ sudo cp 000-default.conf MiVirtualHost.conf
[sudo] contraseña para vboxuser:
vboxuser@ubuntu-despliegue:/etc/apache2/sites-available$ sudo vim MiVirtualHost.conf
```

Dentro de la configuración tendremos que especificar el puerto 443 y habilitar las opciones de SSL.

```
<VirtualHost *:443>
        # 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.

        ServerAdmin aitor.pascual@educa.madrid.org
        ServerName www.miServerUbuntuClase.com
        ServerAlias miServerUbuntuClase.es
        DocumentRoot /var/www/miWeb

        SSLEngine on
        SSLCertificateFile /etc/apache2/MiVirtualHost/ssl/MiVirtualHost.crt
        SSLCertificateKeyFile /etc/apache2/MiVirtualHost/ssl/key

        # 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 ${APACHE_LOG_DIR}/error_miWeb.log
        CustomLog ${APACHE_LOG_DIR}/access_miWeb.log combined

        # 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>
```

PUERTO HHTPS

Habilitamos ssl y damos las rutas de la clave privada y certificado

Aitor Pascual Jiménez
Despliegue de Aplicaciones Web
2º DAW

Con la configuración guardada debemos habilitar la configuración con a2ensite y reiniciar el servicio.

```
vboxuser@ubuntu-despliegue:/etc/apache2/sites-available$ sudo a2ensite MiVirtualHost.conf
Enabling site MiVirtualHost.
To activate the new configuration, you need to run:
  systemctl reload apache2
vboxuser@ubuntu-despliegue:/etc/apache2/sites-available$ sudo systemctl restart apache2
🔐 Enter passphrase for SSL/TLS keys for www.miServerUbuntuClase.com:443 (RSA): ********
vboxuser@ubuntu-despliegue:/etc/apache2/sites-available$ 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 Fri 2025-01-24 10:03:31 CET; 6s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 3140 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 3150 (apache2)
      Tasks: 6 (limit: 9438)
     Memory: 13.5M
        CPU: 108ms
     CGroup: /system.slice/apache2.service
             ├─3150 /usr/sbin/apache2 -k start
             ├─3151 /usr/sbin/apache2 -k start
             ├─3152 /usr/sbin/apache2 -k start
             ├─3153 /usr/sbin/apache2 -k start
             ├─3154 /usr/sbin/apache2 -k start
             └─3155 /usr/sbin/apache2 -k start

janv. 24 10:03:24 ubuntu-despliegue systemd[1]: Starting The Apache HTTP Server...
janv. 24 10:03:31 ubuntu-despliegue systemd[1]: Started The Apache HTTP Server.
```

Si tratamos de acceder desde https a nuestra web, como ya habíamos anticipado, el cliente (firefox) nos va a avisar de que el certificado no está reconocido.

Aitor Pascual Jiménez
Despliegue de Aplicaciones Web
2º DAW

Si presionamos en avanzado podemos acceder de todas formas.

Como vemos al darle a aceptar vemos la página configurada en la práctica de Hosts virtuales desde https.