

# Server APACHE

## \*Instalación del servidor

>Primero hacemos update del sistema para instalar los repositorios más actuales del servidor apache proporcionado por el apt

**sudo apt update**

>Instalamos el servidor apache:

**sudo apt-get install apache2**

>Una vez instalado vamos a comprobar que está correctamente instalado:

**systemctl status apache2**

```
root@apache:/home/apache# 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 2023-11-17 08:47:00 UTC; 37s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1160 (apache2)
    Tasks: 55 (limit: 6990)
   Memory: 5.4M
      CPU: 38ms
   CGroup: /system.slice/apache2.service
           |-1160 /usr/sbin/apache2 -k start
           |-1162 /usr/sbin/apache2 -k start
           `--1163 /usr/sbin/apache2 -k start

Nov 17 08:47:00 apache systemd[1]: Starting The Apache HTTP Server...
Nov 17 08:47:00 apache systemd[1]: Started The Apache HTTP Server.
```

## Está funcionando en localhost



>Para activar el funcionamiento del servidor con los certificados vamos a ejecutar primero este comando:

**a2enmod -m ssl**

```
root@apache:/home/apache# apachectl status
/usr/sbin/apachectl: 113: www-browser: not found
'www-browser -dump http://localhost:80/server-status' failed.
Maybe you need to install a package providing www-browser or you
need to adjust the APACHE_LYNX variable in /etc/apache2/envvars
root@apache:/home/apache# a2enmod -m 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
```

>Cerramos el puerto 80 en el archivo /etc/apache2/ports.conf

```
# If you just change the port or add more ports
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

#Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

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

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

>Desactivamos el sitio web por defecto que viene precargado

**sudo a2dissite 000-default.conf**

>Reiniciamos el servidor apache para aplicar todos los cambios

**systemctl reload apache2**

>Ahora vamos a añadir las IP que permitan entrar al servidor por DNS

**nano /etc/hosts**

```
127.0.1.1      xucliente
192.168.1.155  shu.com www.shu.com publica.com intranet.shu.com intranet

127.0.0.1      localhost
::1           localhost ip6-localhost ip6-loopback
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
# --- BEGIN PVE ---
172.16.1.12    apache.ausias.lan apache
# --- END PVE ---
```

>Creamos el directorio de almacenamiento de la empresa web

```
root@apache:/var/www/publica/shu.com#
```

>Depositamos dentro el contenido de nuestra página web, en este caso un html

>Para crear la configuración del espacio web nos dirigimos a las sites  
avaliables para crear una configuracion del ssl copia de una ya  
incorporada para modificarla

```
cd /etc/apache2/sites-available  
sudo cp default-ssl.conf shu.com.conf
```

>Añadimos nuestra información al archivo

```
root@apache:/etc/apache2/sites-available# cat /etc/apache2/sites-available/shu.com.conf
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin webmaster@shu.com

        DocumentRoot /var/www/publica/shu.com

        ServerName www.shu.com
        ServerAlias shu.com
        # 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.log
        CustomLog ${APACHE_LOG_DIR}/access.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

        #   SSL Engine Switch:
        #   Enable/Disable SSL for this virtual host.
        SSLEngine on
```

```
<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
    SSLOptions +StdEnvVars
</Directory>
<Directory /var/www/publica/shu.com>
</Directory>
```

>Para activar la configuración debemos ejecutar el siguiente comando

```
sudo a2ensite shu.com.conf  
systemctl reload apache2
```

**Hemos realizado lo mismo para hacer la intranet pero en una carpeta alternativa a la publica, que se llama privada, el resultado de**

```
27.0.0.1che:/etc/apache2/sites-available# curl -k -H "Host:intranet.shu.com" https://1
<p> INTRANET </p>
root@apache:/etc/apache2/sites-available#
```

**da correctamente la página web de la intranet, estamos felices.**

**y nuestro apartado home, está bien**

```
27.0.0.1che:/etc/apache2/sites-available# curl -k -H "Host:intranet.shu.com" https://1
<p> INTRANET </p>
root@apache:/etc/apache2/sites-available# curl -k -H "Host:shu.com" https://127.0.0.1
<p> HOLA </p>
```

>Ahora vamos a crear un usuario para acceder a la intranet

**sudo htpasswd -c /opt/apache2/intranet.com/passwords cliente**

**>Y se pone la contraseña que se desea**

>Tras reiniciar el servidor, podremos acceder mediante usuario y contraseña, en este caso igual que el usuario creado(cliente:cliente):

```
root@apache:/opt/apache2/intranet.com# systemctl restart apache2
root@apache:/opt/apache2/intranet.com#
root@apache:/opt/apache2/intranet.com# curl -k -u cliente:cliente -H "Host:intranet.s
hu.com" https://127.0.0.1
<p> INTRANET </p>
root@apache:/opt/apache2/intranet.com#
```

**AHORA VAMOS A CREAR UN CERTIFICADO CON SSL Y AÑADIRLO A UN HOST CONCRETO EN ESTE CASO INTRANET.SHU.com**

>Primero instalamos openssl para añadir un certificado propio para probar

**sudo apt-get install openssl**

>Generamos una private key

```
openssl genpkey -algorithm RSA -out /etc/ssl/claveprivada.key
```

[illegible]

Esta debe ir dentro de `/etc/ssl/private/claveprivada.key`, la movemos

```
mv claveprivada.key /private/
```

>Y ahora un certificado y lo autofirmamos

## Certificado:

**openssl req -new /etc/ssl/private/claveprivada.key -out  
/etc/ssl/certs/certificado.csr**

```
root@apache:/etc/ssl# openssl req -new /etc/ssl/private/claveprivada.key -out /etc/ssl/certs/certificado.csr
req: Use -help for summary.
root@apache:/etc/ssl# openssl req -new -key /etc/ssl/private/claveprivada.key -out /etc/ssl/certs/certificado.csr
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]:Spain
Locality Name (eg, city) []:Spain
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SHU
Organizational Unit Name (eg, section) []:SHU
Common Name (e.g. server FQDN or YOUR name) []:SHU
Email Address []:Shu

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

## Firmamos:

**openssl x509 -req -days 365 -in /etc/ssl/certs/certificado.csr  
-signkey /etc/ssl/private/claveprivada.key -out  
/etc/ssl/certs/certificado.crt**

```
root@apache:/etc/ssl# openssl x509 -req -days 365 -in /etc/ssl/certs/certificado.csr -signkey /etc/ssl/private/claveprivada.key -out
/etc/ssl/certs/certificado.crt
Certificate request self-signature ok
subject=C = ES, ST = Spain, L = Spain, O = SHU, OU = SHU, CN = SHU, emailAddress = Shu
```

**Una vez con el certificado firmado y la private key, lo  
situamos en el ssl de nuestra web .conf, intranet.shu.com.conf →**

```
PX-dAW-E6 - Proxmox Console - Opera
VPN Notsecure pro.ausiasmarch.es:50626
GNU nano 6.2 /etc/apache2/sites-available/intranet.shu.com.conf *
# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/certificado.crt
SSLCertificateKeyFile /etc/ssl/private/claveprivada.key
```

## PROBAMOS QUE FUNCIONA Y FUNCIONA TRAS REINICIAR EL SERVIDOR APACHE2

```
PX-dAW-E6 - Proxmox Console - Opera
VPN ⚠ Not secure pro.ausiasmarch.es:50626

SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile      /etc/ssl/certs/certificado.crt
SSLCertificateKeyFile   /etc/ssl/private/claveprivada.key

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convenience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

# Certificate Authority (CA):
# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one

root@apache:/etc/ssl/certs# systemctl restart apache2
root@apache:/etc/ssl/certs# curl -k -u cliente:cliente -H "Host:intranet.shu.com" https://127.0.0.1
<p> INTRANET </p>
root@apache:/etc/ssl/certs#
```

## HEMOS PROBADO CONEXIÓN DESDE LUBUNTU y FUNCIONA CON LA IP DE EL CONTENEDOR

```
aso@Lubu22:~$ curl -k -u cliente:cliente -H "Host:intranet.shu.com" https://172.16.1.12
<p> INTRANET </p>
aso@Lubu22:~$
```



**POR WEB**

**pide contraseña y autentica igual**

