Tema 4

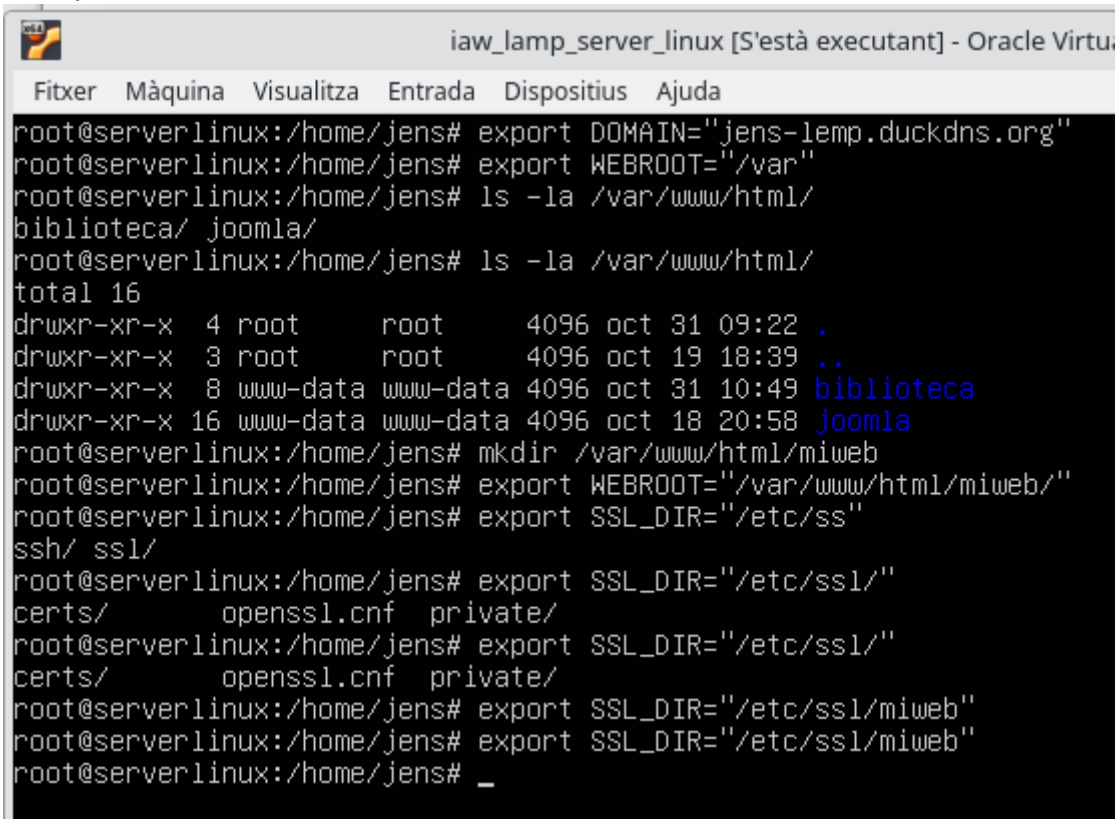
# Configuración De


en


1) Objetivo
Emitir e instalar un certificado válido de Let’s Encrypt usando el desafío DNS-01 (registro TXT en DNS), y servir una web por HTTPS en Apache2.

2) Prerrequisitos
- Ubuntu 22.04+ (Desktop/Server/WSL) con acceso a Internet.
- Usuario con permisos sudo.
- Navegador web para acceder a DuckDNS.
- No hace falta IP pública ni abrir puertos.

3) Variables (para copiar/pegar, solución de chat para no escribirlo siempre…)
- Sustituye por tu subdominio de DuckDNS.
- export DOMAIN="tu-nombre.duckdns.org"
- export WEBROOT="/var/www/miweb"



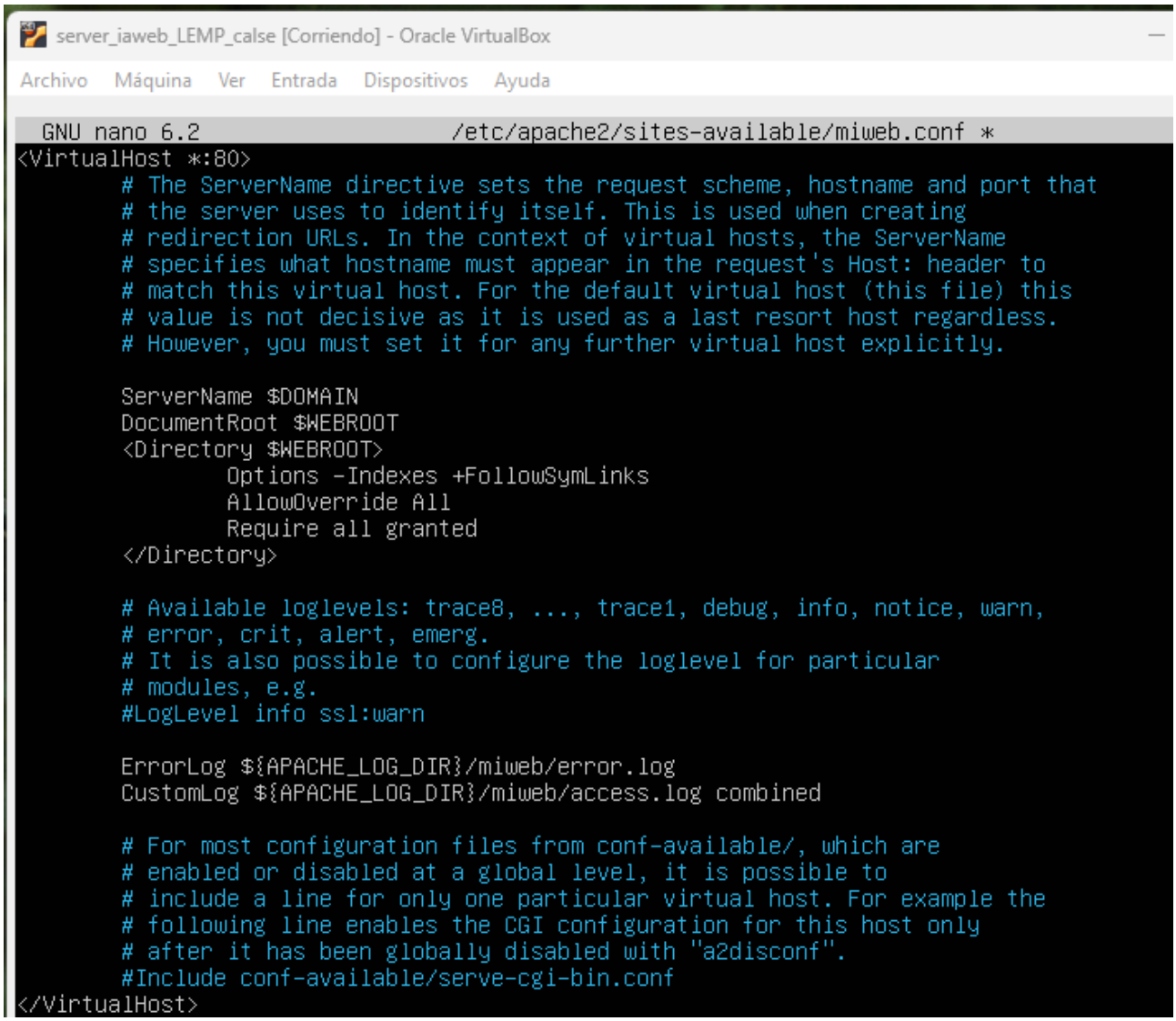

4) Pasos secuenciados
4.1 Crear subdominio gratuito en DuckDNS
Entra en https://www.duckdns.org e inicia sesión (Google/GitHub).
En add domain, escribe tu-nombre → tendrás tu-nombre.duckdns.org.
(Opcional) En current ip deja la IP pública que aparezca por defecto. No es crítica para DNS-01.
Comprobación rápida:
ping -c 1 $DOMAIN / ping -c 1 tu-nombre-duckdns.org
(Que resuelva un valor no es imprescindible para DNS-01, pero confirma que existe el dominio.)


4.2 Instalar Apache2 y preparar una web mínima (o si ya estas gastando una que tengas hecha de una práctica anterior, cosa que te aconsejo, este punto deberías saltartelo)

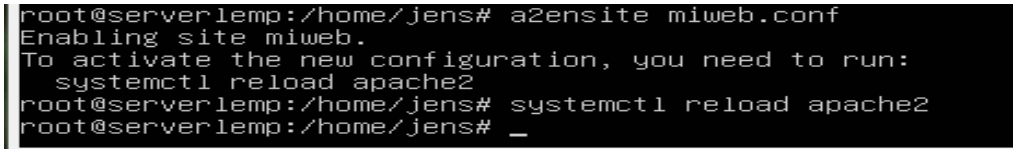
sudo apt update sudo apt install -y apache2
sudo mkdir -p $WEBROOT
echo "<h1>HTTPS con Let's Encrypt (DNS-01)</h1>" | sudo tee $WEBROOT/index.html

```
root@serverlinux:/home/jens# mkdir -p $WEBROOT echo "<h1> HTTPS con OpenSSL (Certificado autofirmado
)</h1>" | sudo tee $WEBROOT/index.html
root@serverlinux:/home/jens# _
```

Crear VirtualHost HTTP básico:
sudo tee /etc/apache2/sites-available/miweb.conf >/dev/null
<<'EOF' <
VirtualHost *:80>
ServerName tu-nombre.duckdns.org
DocumentRoot /var/www/miweb
<Directory /var/www/miweb>
Options -Indexes +FollowSymLinks
AllowOverride All
Require all granted
</Directory>
ErrorLog ${APACHE_LOG_DIR}/miweb_error.log
CustomLog ${APACHE_LOG_DIR}/miweb_access.log combined
</VirtualHost>




sudo sed -i "s/tu-nombre.duckdns.org/$DOMAIN/" /etc/apache2/sites-available/miweb.conf
sudo a2ensite miweb.conf
sudo systemctl reload apache2



Comprobación local:
curl -I http://127.0.0.1
Debe devolver 200 OK.

```
root@serverlemp:/home/jens# curl -I http://127.0.0.1
HTTP/1.1 200 OK
Date: Thu, 13 Nov 2025 14:21:02 GMT
Server: Apache/2.4.52 (Ubuntu)
Last-Modified: Thu, 06 Nov 2025 19:04:42 GMT
ETag: "29af-642f1bcd123c1"
Accept-Ranges: bytes
Content-Length: 10671
Vary: Accept-Encoding
Content-Type: text/html
```

4.3 Instalar Certbot
sudo apt install -y certbot

```
root@serverlemp:/home/jens# openssl x509 -req -days 365 -in request.csr -signkey privkey.pem -out fu
llchain.pem
Certificate request self-signature ok
subject=C = ES, ST = Valencia, L = Valencia, O = miweb, OU = IT, CN = jens-lemp.duckdns.org
```

server_iaweb_LEMP_calse [Corriendo] - Oracle VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

```
root@serverlemp:/home/jens# ls -l /etc/ssl/jens-lemp.duckdns.org/
total 12
-rw-r--r-- 1 root root 1241 nov 13 14:28 fullchain.pem
-rw------- 1 root root 1704 nov 13 14:22 privkey.pem
-rw-r--r-- 1 root root 1017 nov 13 14:24 request.csr
root@serverlemp:/home/jens# _
```

server_iaweb_LEMP_calse [Corriendo] - Oracle VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

```
  GNU nano 6.2                 /etc/apache2/sites-available/miwebssl.conf *
<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.
        ServerName jens-lemp.duckdns.org
        DocumentRoot /var/www/html/miweb
        SSLEngine on
        SSLCertificateFile /etc/ssl/jens-lemp.duckdns.org/fullchain.pem
        SSLCertificateKeyFile /etc/ssl/jens-lemp.duckdns.org/privkey.pem
        <Directory /var/www/html/miweb>
                Options -Indexes +FollowSymLinks
                AllowOverride All
                Require all granted
        </Directory>
        # 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}/miwebssl/error.log
        CustomLog ${APACHE_LOG_DIR}/miwebssl/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
</VirtualHost>
```

4.4 Emitir certificado con DNS-01 (manual)
Lanzar certbot:
sudo certbot -d $DOMAIN --manual --preferred-challenges dns certonly
Qué ocurrirá: (LEE CON DETENIMIENTO ¡!! )
Certbot mostrará un valor TXT y te pedirá crear un registro DNS:
Nombre (host): _acme-challenge.tu-nombre.duckdns.org
Tipo: TXT
Valor: (cadena larga que te da certbot)

Dónde crear el TXT en DuckDNS:
En la página de tu dominio, baja a la sección TXT.
En txt pega el valor que muestra certbot.
En subdomain escribe _acme-challenge.
Pulsa update TXT.

NOTA : Si esto no va, copia y pega esta URL en el navegador y cuando le des a enter te debe aparecer un OK en una esquina. Ya lo tendrías.
https://www.duckdns.org/update?domains=tu-nombre&token=9a4778ef-4cbf-4775-a809-b8d206d6cd4f&txt=VALOR_TXT&verbose=true"

Esperar propagación (30–120 s) y verificar en terminal:
dig TXT _acme-challenge.$DOMAIN +short

Debe devolver exactamente el valor que pidió certbot.
Vuelve a la terminal de certbot y pulsa Enter.
Si todo está correcto verás:
Congratulations! Your certificate and chain have been saved at:
/etc/letsencrypt/live/tu-nombre.duckdns.org/fullchain.pem
Your key file has been saved at:
/etc/letsencrypt/live/tu-nombre.duckdns.org/privkey.pem

4.5 Activar HTTPS en Apache (VirtualHost :443)

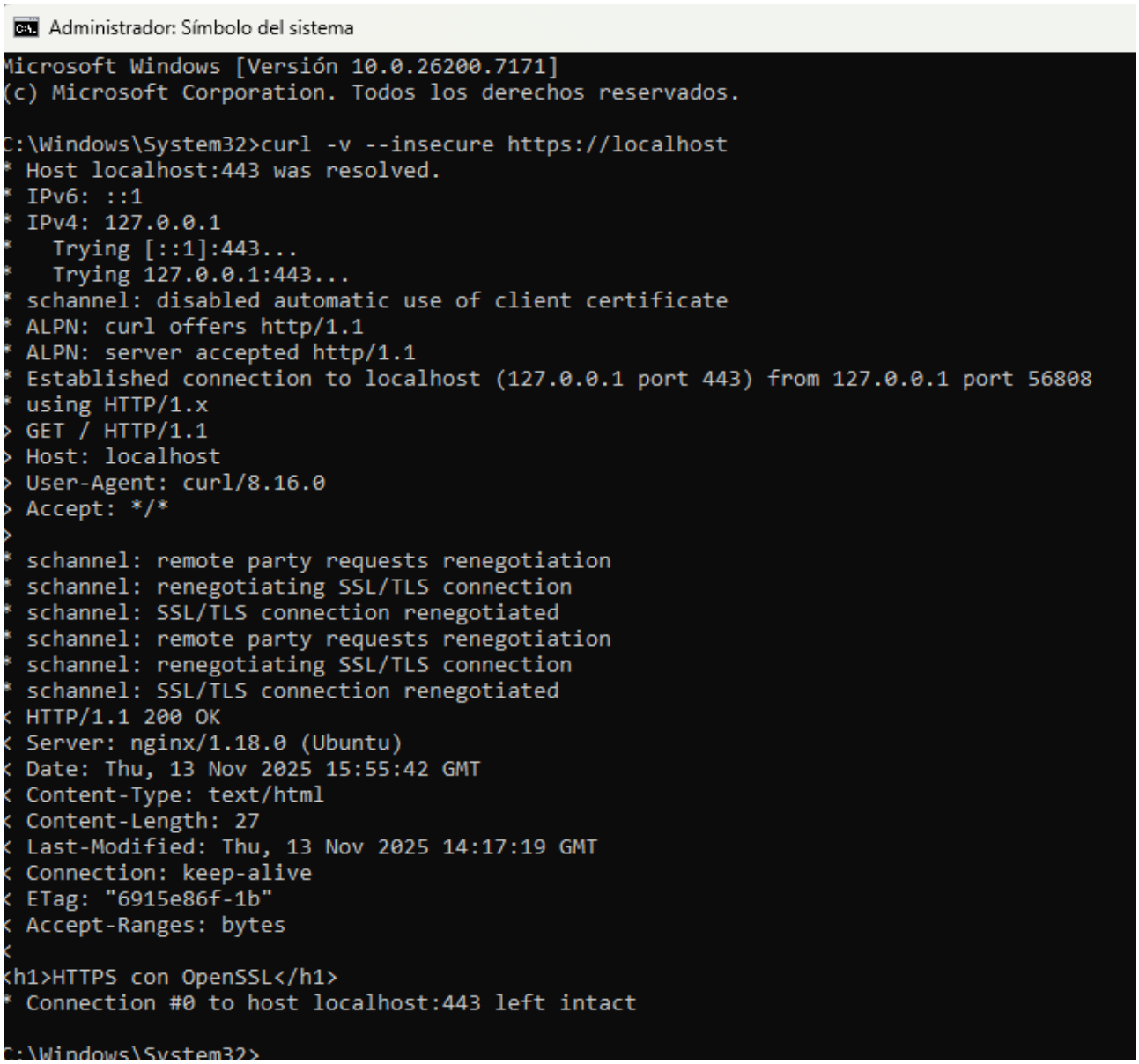
sudo a2enmod ssl
sudo tee /etc/apache2/sites-available/miweb-ssl.conf

EOF sudo sed -i "s/tu-nombre.duckdns.org/$DOMAIN/g" /etc/apache2/sites-available/miweb-ssl.conf sudo a2ensite miweb-ssl.conf

sudo systemctl reload apache2

Verifica que los ficheros existen:
sudo certbot certificates

4.6 Pruebas
En el propio equipo:
curl -I https://$DOMAIN --insecure
Desde un navegador (si tiene acceso a tu equipo): https://tu-nombre.duckdns.org
(Si el equipo no es accesible, al menos el certificado es real y válido y Apache está sirviendo localmente.)





Lo he tenido que hacer con el nombre de “localhost”, però la configuracion es la misma, si que me funciona 😊