

ARQUITECTURA EN LA NUBE

Sergio de la Coba García

2º ASIR

ÍNDICE

PRÁCTICA 2: ARQUITECTURA EN LA NUUBE

I. Introducción

- I.1. Objetivo de la práctica
- I.2. Tecnologías empleadas y entorno de trabajo

II. Configuración del Servidor Apache

- I.1. Actualización del sistema
- I.2. Instalación de Apache
- I.3. Cambio del puerto de escucha a 8080
- I.4. Configuración de VirtualHost
- I.5. Instalación de PHP y módulo apache2-mod-php
- I.6. Prueba de funcionamiento
- I.7. Verificación de curl

III. Configuración del Servidor Nginx

- I.1. Instalación del Nginx
- I.2. Modificación del puerto
- I.3. Creación de página HTML simple
- I.4. Prueba de funcionamiento

IV. Configuración del Servidor Caddy

- I.1. Instalación de dependencias y claves GPG
- I.2. Generación de certificado
- I.3. Añadir de ventaja
- I.4. Renovación de clave/var/www/caddy
- I.5. Configuración de Caddyfile en puerto 8082
- I.6. Prueba con curl README.md

V. Comprobaciones Finales en Apache

- VI. Instalación de Certbot y módulo SSL
- VII. Generación de certificado auto
- VIII. Configuración de default-ssl.conf
- IX. Análisis de resultados

VII. Conclusiones

Empiezo actualizando el sistema con sudo apt update y sudo apt upgrade. Luego instalo Apache con sudo apt install apache2 -y, aunque como se ve, ya lo tenía instalado.

Siguiendo la práctica, edito el archivo de configuración /etc/apache2/ports.conf y cambio el puerto por defecto Listen 80 por Listen 8080.

```
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer req
100%[=====] 1/1
```

Con los paquetes ya actualizados empezamos la instalación y configuración.

```
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.8).
The following package was automatically installed and is no longer required:
  liblvm19
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Instalamos apache.

Configuramos el puerto 8080 para Apache.

```
GNU nano 7.2          /etc/apache2/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 8080

<IfModule ssl_module>
    Listen 443
</IfModule>

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

Ahora modificamos VirtualHost

Edito el archivo del "VirtualHost" en /etc/apache2/sites-available/000-default.conf para que escuche en el nuevo puerto (la captura muestra 8888, pero está configurado para 8080 como se ve en el ports.conf).

Instalo PHP y el módulo de Apache (libapache2-mod-php), que también estaban ya en su última versión.

```

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32
GNU nano 7.2                               /etc/apache2/sites-available/000-default.conf
virtualHost *:8080>
    # 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@localhost
    DocumentRoot /var/www/html

    # 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
        [ Read 29 lines ]
Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute   ^C Location   M-U Undo   M-A Set Mark
Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line M-E Redo   M-6 Copy

```

Instalamos php

```

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo apt install php libapache2-
mod-php -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
php is already the newest version (2:8.3+93ubuntu2).
libapache2-mod-php is already the newest version (2:8.3+93ubuntu2).
The following package was automatically installed and is no longer required:
  liblvm19
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$

```

```

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo systemctl restart apache2
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server

```

Reiniciamos Apache y hacemos el status para ver que esta funcionando correctamente en el puerto 8080.

Reinicio Apache con sudo systemctl restart apache2 y compruebo su estado con sudo systemctl status apache2 , confirmando que está active (running).

```

~ lines 1-19/19 (END)
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; presen>
  Active: active (running) since Thu 2025-10-16 14:28:24 CEST; 9s ago
    Docs: https://httpd.apache.org/docs/2.4/
  Process: 934 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/)>
 Main PID: 937 (apache2)
   Tasks: 6 (limit: 9350)
  Memory: 10.6M (peak: 12.3M)
     CPU: 29ms
    CGroup: /system.slice/apache2.service
            └─937 /usr/sbin/apache2 -k start
              ├─939 /usr/sbin/apache2 -k start
              ├─940 /usr/sbin/apache2 -k start
              ├─941 /usr/sbin/apache2 -k start
              ├─942 /usr/sbin/apache2 -k start
              └─943 /usr/sbin/apache2 -k start

Oct 16 14:28:24 A6Alumno18 systemd[1]: Starting apache2.service - The Apache>
Oct 16 14:28:24 A6Alumno18 systemd[1]: Started apache2.service - The Apache >
~

```

Ahora vamos a crear un archivo php de prueba

```

Lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ echo "<?php phpinfo(); ?>" | s
  tee /var/www/html/info.php
<?php phpinfo(); ?>
Lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$
```

Vemos que al ejecutar el comando nos devuelve: ?, lo que indica que se ha creado bien. Por ultimo en la instalación de Apache hacemos un curl para verificar desde la terminal el puerto 8080.

Hago la prueba final de esta parte con curl `http://localhost:8080/info.php`. La terminal me devuelve todo el código HTML de la página `phpinfo`, confirmando que Apache y PHP funcionan correctamente en el puerto 8080.

```

Alumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ curl http://localhost:8080/info.php
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
<style type="text/css">
body {background-color: #fff; color: #222; font-family: sans-serif;}
pre {margin: 0; font-family: monospace;}
a:link {color: #009; text-decoration: none; background-color: #fff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse; border: 0; width: 934px; box-shadow: 1px 2px 3px rgba(0, 0, 0, 0.2);}
.center {text-align: center;}
.center table {margin: 1em auto; text-align: left;}
.center th {text-align: center !important;}
td, th {border: 1px solid #666; font-size: 75%; vertical-align: baseline; padding: 4px 5px;}
th {position: sticky; top: 0; background: inherit;}
h1 {font-size: 150%;}
h2 {font-size: 125%;}
h2 a:link, h2 a:visited{color: inherit; background: inherit;}
p {text-align: left;}
.e {background-color: #ccf; width: 300px; font-weight: bold;}
.h {background-color: #99c; font-weight: bold;}
.v {background-color: #ddd; max-width: 300px; overflow-x: auto; word-wrap: break-word;}
.v i {color: #999;}
img {float: right; border: 0;}
hr {width: 934px; background-color: #ccc; border: 0; height: 1px;}
:root {--php-dark-grey: #333; --php-dark-blue: #4F5B93; --php-medium-blue: #8392BF; --php-light-blue: #E2E4EF; --php-accent-purple: #793862}@media (prefers-color-scheme: dark) {
    body {background: var(--php-dark-grey); color: var(--php-light-blue)}
    .h td, td.e, th {border-color: #606A90}
    td {border-color: #505153}
    .e {background-color: #404A77}
    .h {background-color: var(--php-dark-blue)}
    .v {background-color: var(--php-dark-grey)}
    hr {background-color: #505153}
}
</style>
<title>PHP 8.3.6 - phpinfo()</title><meta name="ROBOTS" content="NOINDEX,NOFOLLOW,NOARCHIVE" /></head>
<body><div class="center">
<table>
<tr class="h"><td>
<a href="http://www.php.net/">Servidor Nginx</h1><p>Funcionando en puerto 8081</p>" | sudo tee /usr/share/nginx/html/index.html
<h1>Servidor Nginx</h1><p>Funcionando en puerto 8081</p>
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$
```

```
<h1>Servidor Nginx</h1><p>Funcionando en puerto 8081</p>
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo systemctl restart nginx
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
    Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
      Active: active (running) since Tue 2025-10-21 09:13:09 CEST; 15s ago
        Docs: man:nginx(8)
     Process: 1036 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 1037 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
      Main PID: 1039 (nginx)
        Tasks: 17 (limit: 9350)
       Memory: 12.9M (peak: 14.3M)
         CPU: 54ms
      CGroup: /system.slice/nginx.service
              ├─1039 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
              ├─1040 "nginx: worker process"
              ├─1041 "nginx: worker process"
              ├─1042 "nginx: worker process"
              ├─1043 "nginx: worker process"
              ├─1044 "nginx: worker process"
              ├─1045 "nginx: worker process"
              ├─1046 "nginx: worker process"
              ├─1048 "nginx: worker process"
              ├─1049 "nginx: worker process"
              ├─1050 "nginx: worker process"
              └─1051 "nginx: worker process"
```

Hago la prueba con curl http://localhost:8081. Como se ve, me devuelve mi página personalizada "Hola Mundo desde Nginx", así que funciona perfectamente.

```
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ curl http://localhost:8081
<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$
```

INSTALACION Y CONFIGURACION DE CADDY:

Para instalar Caddy, primero instalo sus dependencias: debian-keyring, debian-archive-keyring y apt-transport-https.

Añado la clave GPG de Caddy y su repositorio oficial a las fuentes de apt.

Actualizo e instalo Caddy (sudo apt install caddy). Creo el directorio /var/www/caddy y el archivo README.md con su contenido .

```

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apt-transport-https is already the newest version (2.8.3).
The following package was automatically installed and is no longer required:
  libl1vm19
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  debian-archive-keyring debian-keyring
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 31.5 MB of archives.
After this operation, 33.4 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 debian-archive-keyring all 2023.4ubuntu1 [168 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/universe amd64 debian-keyring all 2023.12.24 [31.3 MB]
Fetched 31.5 MB in 3s (12.6 MB/s)
Selecting previously unselected package debian-archive-keyring.
(Reading database ... 42212 files and directories currently installed.)
Preparing to unpack .../debian-archive-keyring_2023.4ubuntu1_all.deb ...
Unpacking debian-archive-keyring (2023.4ubuntu1) ...
Selecting previously unselected package debian-keyring.
Preparing to unpack .../debian-keyring_2023.12.24_all.deb ...
Unpacking debian-keyring (2023.12.24) ...
Setting up debian-archive-keyring (2023.4ubuntu1) ...
Setting up debian-keyring (2023.12.24) ...

```

```

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ # Agregar clave GPG
curl -sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ # Agregar repositorio
curl -sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee /etc/apt/sources.list.d/caddy-stable.list
# Source: Caddy
# Site: https://github.com/caddyserver/caddy
# Repository: Caddy / stable
# Description: Fast, multi-platform web server with automatic HTTPS

deb [signed-by=/usr/share/keyrings/caddy-stable-archive-keyring.gpg] https://dl.cloudsmith.io/public/caddy/stable/debian any-version main

deb-src [signed-by=/usr/share/keyrings/caddy-stable-archive-keyring.gpg] https://dl.cloudsmith.io/public/caddy/stable/debian any-version main

```

```

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo apt update && sudo apt install caddy -y
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:4 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version InRelease [14.8 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1523 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1223 kB]
Get:8 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version/main amd64 Packages [4329 B]
Get:9 http://archive.ubuntu.com/ubuntu noble-updates/main Translation-en [291 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.3 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1496 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [205 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [8968 B]
Get:16 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [904 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [301 kB]
Get:18 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:19 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.3 kB]
Get:20 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2137 kB]
Get:21 http://archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [483 kB]
Get:22 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:23 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [516 B]
Get:24 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [30.3 kB]
Get:25 http://archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [5564 B]

```

Descargo la imagen de prueba de Python y la muevo al directorio de Caddy.

Edito el Caddyfile y lo configuro para que escuche en el puerto :8082 y sirva los archivos del directorio /var/www/caddy. Reinicio Caddy y verifico que está active (running).

Pruebo que Caddy sirve el archivo Markdown correctamente con curl
 http://localhost:8082/README.md, y me devuelve el texto que escribí

```

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo mkdir -p /var/www/caddy
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ echo "# Bienvenido a Caddy" | sudo tee /var/www/caddy/README.md
# Bienvenido a Caddy
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ echo "" | sudo tee -a /var/www/caddy/README.md

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ echo "Este servidor está funcionando correctamente." | sudo tee -a /var/www/caddy/README.md
Este servidor está funcionando correctamente.
-bash: /var/www/caddy/README.md: Permission denied
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ echo "Este servidor está funcionando correctamente." | sudo tee -a /var/www/caddy/README.md
Este servidor está funcionando correctamente.
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ echo "" | sudo tee -a /var/www/caddy/README.md

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ echo "## Características" | sudo tee -a /var/www/caddy/README.md
## Características
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ echo "- Servidor moderno" | sudo tee -a /var/www/caddy/README.md
- Servidor moderno
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ echo "- HTTPS automático" | sudo tee -a /var/www/caddy/README.md
- HTTPS automático
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ echo "- Fácil configuración" | sudo tee -a /var/www/caddy/README.md
- Fácil configuración
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ 

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ curl -o /tmp/test-image.jpg "https://www.python.org/static/apple-touch-icon-144x144-precomposed.p
ng"
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload Upload   Total Spent   Left Speed
100  7382  100  7382    0     0  91339      0 --:--:-- --:--:-- 92275
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo mv /tmp/test-image.jpg /var/www/caddy/test.jpg

[GN] alumno18@A6Alumno18:/mnt/c/WINDOWS/system32
GNU nano 7.2                               /etc/caddy/Caddyfile
:8082 {
  root * /var/www/caddy
  file_server browse
  @markdown path *.md
  header @markdown Content-Type text/plain
}
[ Read 7 lines ]  ^G Help  ^O Write Out  ^W Where Is  ^K Cut  ^T Execute  ^C Location  M-U Undo
^X Exit  ^R Read File  ^V Replace  ^U Paste  ^J Justify  ^/ Go To Line  M-E Redo

```

```

Alumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo nano /etc/caddy/Caddyfile
Alumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo systemctl restart caddy
Alumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo nano /etc/caddy/Caddyfile
Alumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo systemctl status caddy
● caddy.service - Caddy
   Loaded: loaded (/usr/lib/systemd/system/caddy.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-10-21 09:48:49 CEST; 1min 53s ago
     Docs: https://caddyserver.com/docs/
 Main PID: 2545 (caddy)
    Tasks: 10 (limit: 9350)
   Memory: 49.6M (peak: 50.3M)
      CPU: 75ms
     CGroup: /system.slice/caddy.service
             └─2545 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile

Oct 21 09:48:49 A6Alumno18 caddy[2545]: {"level": "info", "ts": "1761032929.4809089", "logger": "admin", "msg": "admin endpoint initialized"}
Oct 21 09:48:49 A6Alumno18 caddy[2545]: {"level": "info", "ts": "1761032929.4824255", "logger": "tls.cache.maintenance", "msg": "TLS cache maintenance started"}
Oct 21 09:48:49 A6Alumno18 caddy[2545]: {"level": "warn", "ts": "1761032929.4832914", "logger": "http", "msg": "HTTP/2 skipper initialized"}
Oct 21 09:48:49 A6Alumno18 caddy[2545]: {"level": "warn", "ts": "1761032929.4833126", "logger": "http", "msg": "HTTP/3 skipper initialized"}
Oct 21 09:48:49 A6Alumno18 caddy[2545]: {"level": "info", "ts": "1761032929.4833152", "logger": "http.log", "msg": "server running on port 8082"}
```

Al hacer el curl <http://localhost:8082/> me da este archivo html larguísimo

```

Alumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ curl http://localhost:8082/README.md
# Bienvenido a Caddy

Este servidor está funcionando correctamente.

## Características
- Servidor moderno
- HTTPS automático
- Fácil configuración
Alumno18@A6Alumno18:/mnt/c/WINDOWS/system32$
```

CONFIGURACION DE HTTPS CON CERBOT EN APACHE:

Para el HTTPS, instalo Certbot y el plugin de Apache.

Como no tengo un dominio real, creo un certificado autofirmado con openssl.

Activo el módulo SSL de Apache con sudo a2enmod ssl.

Edito el archivo /etc/apache2/sites-available/default-ssl.conf e introduzco las rutas del certificado (SSLCertificateFile) y la clave (SSLCertificateKeyFile) que acabo de crear.

Edito el archivo /etc/apache2/ports.conf y añado la línea Listen 8443 para que Apache escuche en ese puerto para HTTPS.

```
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo apt install certbot python3-certbot-apache -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  augeas-lenses libaugeas0 python3-acme python3-augeas python3-certbot python3-configargparse python3-icu
  python3-josepy python3-parsedatetime python3-rfc3339
Suggested packages:
  augeas-doc python-certbot-doc python3-certbot-nginx augeas-tools python-acme-doc python-certbot-apache-doc
The following NEW packages will be installed:
  augeas-lenses certbot libaugeas0 python3-acme python3-augeas python3-certbot python3-certbot-apache
  python3-configargparse python3-icu python3-josepy python3-parsedatetime python3-rfc3339
0 upgraded, 12 newly installed, 0 to remove and 2 not upgraded.
Need to get 1657 kB of archives.
After this operation, 8599 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 augeas-lenses all 1.14.1-1build2 [323 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/universe amd64 libaugeas0 amd64 1.14.1-1build2 [166 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-josepy all 1.14.0-1 [22.1 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-rfc3339 all 1.1-4 [6744 B]
Get:5 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-acme all 2.9.0-1 [48.5 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-augeas all 0.5.0-1.1 [9124 B]
Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-configargparse all 1.7-1 [31.7 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-parsedatetime all 2.6-3 [32.8 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-certbot all 2.9.0-1 [267 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/universe amd64 certbot all 2.9.0-1 [89.2 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-certbot-apache all 2.9.0-1 [128 kB]
Get:12 http://archive.ubuntu.com/ubuntu/main amd64 python3-icu amd64 2.12-1build2 [534 kB]
```

```
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo a2enmod ssl
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
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo nano /etc/apache2/sites-available/default-ssl.conf
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$
```

```

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32
GNU nano 7.2                               /etc/apache2/sites-available/default-ssl.conf *
# 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

#   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/apache-selfsigned.crt
SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

#   Server Certificate Chain:

```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo

Escribimos listen 8443 en sudo nano /etc/apache2/ports.conf

```

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32
GNU nano 7.2                               /etc/apache2/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 8080
Listen 8443
<IfModule ssl_module>
    Listen 443
</IfModule>

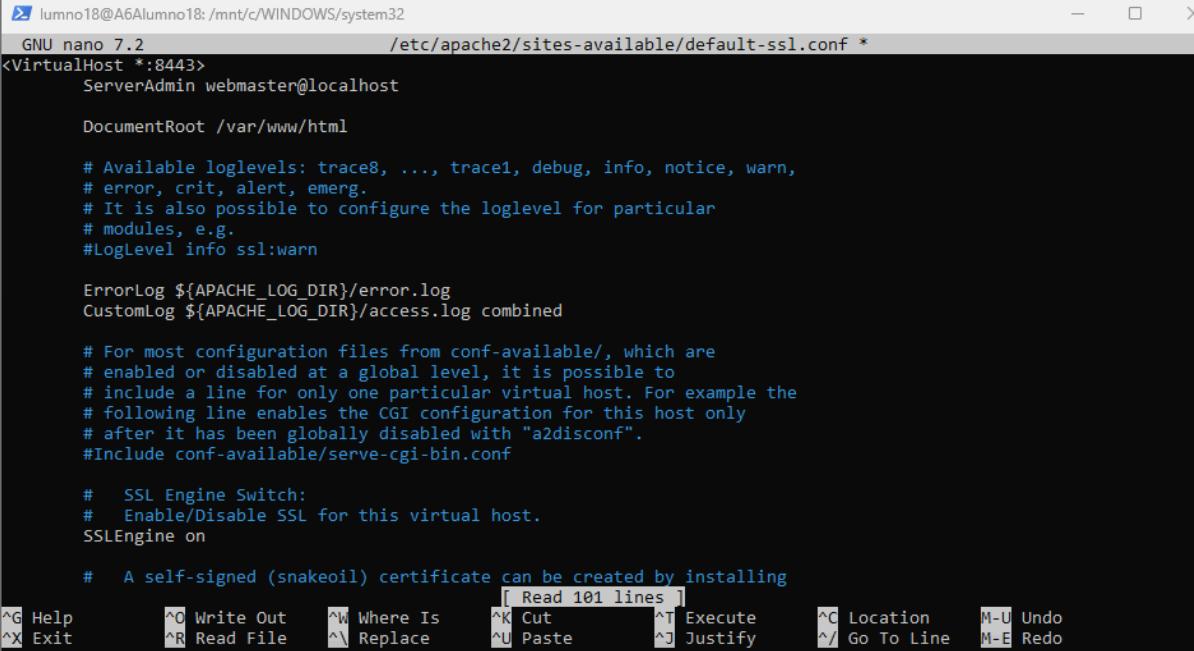
<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

[Read 13 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo

Modifico el VirtualHost SSL en default-ssl.conf para que escuche en *:8443. Activo el sitio con sudo a2ensite default-ssl.conf y reinicio Apache.

Pruebo la conexión HTTPS con curl -k https://localhost:8443. Curiosamente, en lugar de la página de Apache, me responde con el contenido de Nginx.

Hago la comprobación final de los servicios. Con sudo systemctl status apache2 nginx caddy veo que los tres están active (running) al mismo tiempo.



```
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo nano /etc/apache2/sites-available/default-ssl.conf
GNU nano 7.2                               /etc/apache2/sites-available/default-ssl.conf *
<VirtualHost *:8443>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

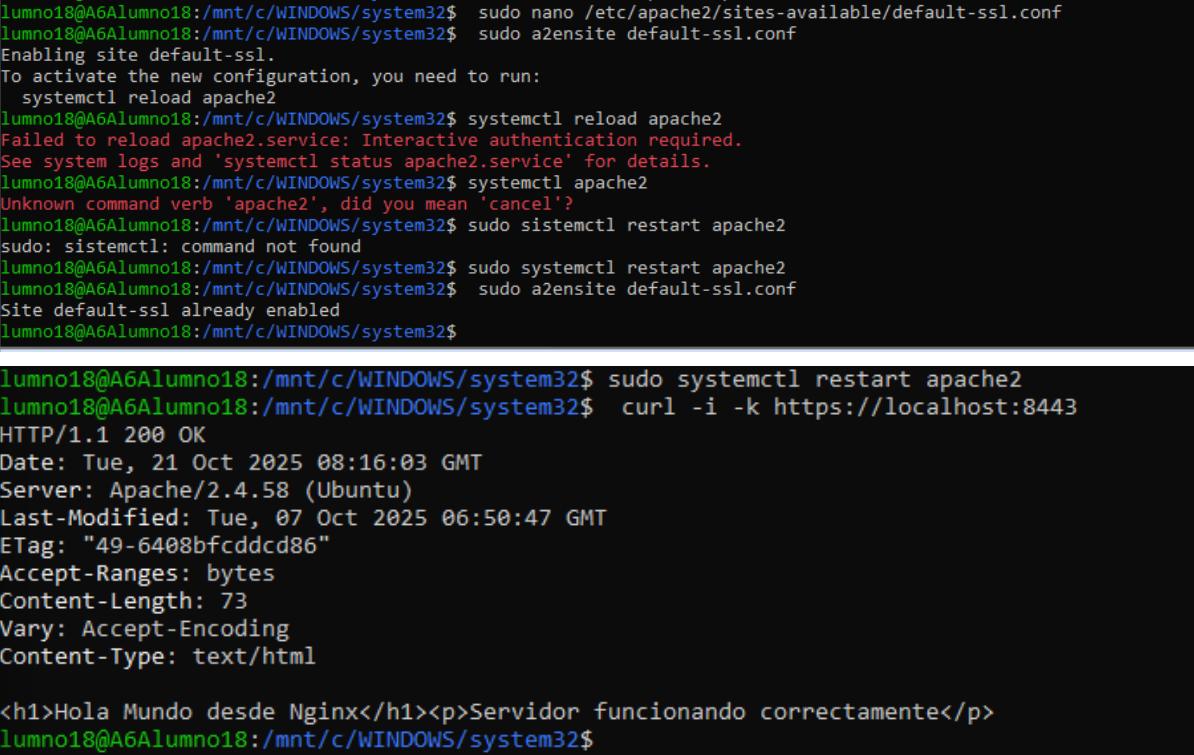
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    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

    # A self-signed (snakeoil) certificate can be created by installing
[ Read 101 lines ]
^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute   ^C Location   M-U Undo
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo
```



```
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo nano /etc/apache2/sites-available/default-ssl.conf
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo a2ensite default-ssl.conf
Enabling site default-ssl.
To activate the new configuration, you need to run:
    systemctl reload apache2
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ systemctl reload apache2
Failed to reload apache2.service: Interactive authentication required.
See system logs and 'systemctl status apache2.service' for details.
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ systemctl apache2
Unknown command verb 'apache2', did you mean 'cancel'?
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo systemctl restart apache2
sudo: systemctl: command not found
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo systemctl restart apache2
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo a2ensite default-ssl.conf
Site default-ssl already enabled
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$
```



```
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo systemctl restart apache2
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ curl -i -k https://localhost:8443
HTTP/1.1 200 OK
Date: Tue, 21 Oct 2025 08:16:03 GMT
Server: Apache/2.4.58 (Ubuntu)
Last-Modified: Tue, 07 Oct 2025 06:50:47 GMT
ETag: "49-6408bfcdcd86"
Accept-Ranges: bytes
Content-Length: 73
Vary: Accept-Encoding
Content-Type: text/html

<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$
```

```

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo systemctl status apache2 nginx caddy
apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-10-21 10:15:51 CEST; 2min 6s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Process: 3012 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 3015 (apache2)
      Tasks: 6 (limit: 9350)
        Memory: 11.6M (peak: 12.9M)
         CPU: 48ms
        CGroup: /system.slice/apache2.service
                  ├─3015 /usr/sbin/apache2 -k start
                  ├─3017 /usr/sbin/apache2 -k start
                  ├─3018 /usr/sbin/apache2 -k start
                  ├─3019 /usr/sbin/apache2 -k start
                  ├─3020 /usr/sbin/apache2 -k start
                  └─3021 /usr/sbin/apache2 -k start

ct 21 10:15:51 A6Alumno18 systemd[1]: Starting apache2.service - The Apache HTTP Server...
ct 21 10:15:51 A6Alumno18 systemd[1]: Started apache2.service - The Apache HTTP Server.

nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-10-21 09:13:09 CEST; 1h 4min ago
     Docs: man:nginx(8)
   Main PID: 1039 (nginx)
      Tasks: 17 (limit: 9350)
        Memory: 11.6M (peak: 14.3M)

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo netstat -tulpn | grep -E '8080|8081|8082|8443'
sudo: netstat: command not found
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ sudo ss -tulpn | grep -E '8080|8081|8082|8443'
tcp  LISTEN  0      511          *:8080          *:*      users:(("nginx",pid=1056,fd=5),("nginx",pid=1055,fd=5),("nginx",pid=1054,fd=5),("nginx",pid=1053,fd=5),("nginx",pid=1052,fd=5),("nginx",pid=1051,fd=5),("nginx",pid=1050,fd=5),("nginx",pid=1049,fd=5),("nginx",pid=1048,fd=5),("nginx",pid=1046,fd=5),("nginx",pid=1045,fd=5),("nginx",pid=1044,fd=5),("nginx",pid=1043,fd=5),("nginx",pid=1042,fd=5),("nginx",pid=1041,fd=5),("nginx",pid=1040,fd=5),("nginx",pid=1039,fd=5))
tcp  LISTEN  0      511          *:8443          *:*      users:(("apache2",pid=3021,fd=6),("apache2",pid=3020,fd=6),("apache2",pid=3019,fd=6),("apache2",pid=3018,fd=6),("apache2",pid=3017,fd=6),("apache2",pid=3015,fd=6))

tcp  LISTEN  0      511          *:8080          *:*      users:(("apache2",pid=3021,fd=4),("apache2",pid=3020,fd=4),("apache2",pid=3019,fd=4),("apache2",pid=3018,fd=4),("apache2",pid=3017,fd=4),("apache2",pid=3015,fd=4))

tcp  LISTEN  0      4096         *:8082          *:*      users:(("caddy",pid=2545,fd=7))

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$
```

```

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ curl http://localhost:8080
<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ curl http://localhost:8081
<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
```

Al hacer el curl al puerto 8082 curl <http://localhost:8082>; nos tiene que salir un html larguísmo. Esta captura muestra solo el principio

```
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32
h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ curl http://localhost:8082

!DOCTYPE html>
html>
    <head>
        <title>/</title>
        <link rel="canonical" href="/" />
        <meta charset="utf-8">
        <meta name="color-scheme" content="light dark">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        style nonce="3534fa9f-f597-47a7-8aed-ae0360c6c924">
            { padding: 0; margin: 0; box-sizing: border-box; }

body {
    font-family: Inter, system-ui, sans-serif;
    font-size: 16px;
    text-rendering: optimizespeed;
    background-color: #f3f6f7;
    min-height: 100vh;

    mg,
    vg {
        vertical-align: middle;
        z-index: 1;

    mg {
        max-width: 100%;

lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$ curl -k https://localhost:8443
<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
lumno18@A6Alumno18:/mnt/c/WINDOWS/system32$
```

En las pruebas finales con curl, todos los puertos responden. Nginx en 8081 y Caddy en 8082 dan su contenido correcto. Los puertos de Apache (8080 y 8443) también responden, pero me devuelven el contenido de Nginx.

