

TEMARIO: ARQUITECTURA EN LA NUBE - CONFIGURACIÓN AVANZADA DE SERVIDORES WEB Y HTTPS

CONCEPTOS BÁSICOS PARA LA CLASE

Las Herramientas que Vamos a Usar

Apache (*httpd*):

Es el servidor web más veterano y popular del mundo. Como un chef experimentado, puede hacer de todo y es muy configurable. Lleva décadas funcionando en millones de sitios web.

Nginx:

Es el servidor web moderno y eficiente. Como un camarero rápido, puede atender muchas peticiones a la vez sin cansarse. Es perfecto para sitios con mucho tráfico.

Caddy:

Es el servidor web más nuevo y fácil de usar. Su superpoder es que configura **HTTPS** automáticamente, algo que con otros servidores requiere esfuerzo manual. Es como tener un mayordomo que lo hace todo por ti.

Certbot:

Es una herramienta que obtiene certificados **SSL/TLS** gratuitos de Let's Encrypt. Estos certificados son como el candado que ves en tu navegador cuando una página es segura (**HTTPS**).

¿Qué es HTTPS y por qué es importante?

HTTP es como enviar postales: cualquiera puede leer lo que pones.

HTTPS es como enviar cartas en sobres cerrados con sello de lacre: nadie puede leer el contenido ni falsificarlo.

Cuando ves el candado en tu navegador, significa que la comunicación está cifrada y es segura.

¿Por qué hacer esta práctica?

En el mundo real, las empresas no usan un solo servidor web. A veces tienen varios funcionando al mismo tiempo para diferentes propósitos:

- Uno para la página principal

- Otro para aplicaciones internas
- Otro como proxy para distribuir la carga

Además, **TODOS los sitios profesionales usan HTTPS**. Aprender a configurarlo es esencial para cualquier profesional de IT.

Al final de esta práctica comprenderás cómo funcionan los servidores web profesionales, cómo pueden coexistir en la misma máquina y cómo proteger las comunicaciones con **HTTPS**.

COMANDOS Y PROCEDIMIENTOS

PARTE 1: INSTALACIÓN Y CONFIGURACIÓN DE APACHE

1. Actualizar el sistema

Comando:

```
sudo apt update && sudo apt upgrade -y
```

Descripción: Actualiza la lista de paquetes y mejora el sistema a las últimas versiones.

```
root@Ubuntu1:/home/vboxuser# sudo apt update && sudo apt upgrade -y
Hit:1 http://es.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1.213 kB]
Get:6 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1.491 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21,5 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [8.748 B]
Get:9 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [882 kB]
Get:11 http://es.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [287 kB]
Get:12 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:13 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15,3 kB]
Get:14 http://es.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Compon
```

2. Instalar Apache2

Comando:

```
sudo apt install apache2 -y
```

Descripción: Instala el servidor web Apache en tu sistema.

```
root@Ubuntu1:/home/vboxuser# 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:
  libllvm19
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 20 not upgraded.
```

3. Configurar Apache en puerto 8080

Comando:

```
sudo nano /etc/apache2/ports.conf
```

Descripción: Abre el archivo de configuración de puertos. Cambia Listen 80 por **Listen 8080**.

```
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>

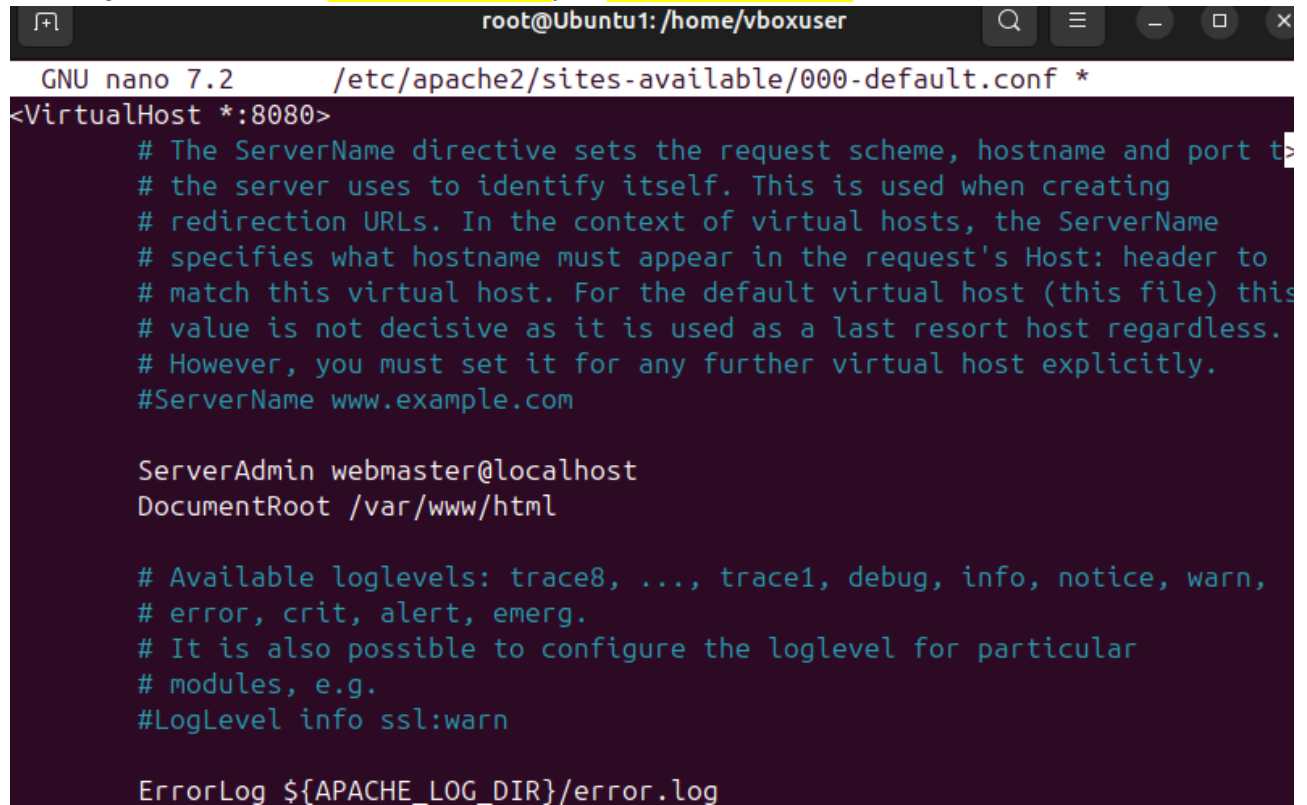
<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

4. Modificar el VirtualHost

Comando:

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Descripción: Cambia `<VirtualHost *:80>` por `<VirtualHost *:8080>`.



```
GNU nano 7.2 /etc/apache2/sites-available/000-default.conf *
<VirtualHost *:8080>
    # The ServerName directive sets the request scheme, hostname and port to
    # 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
```

5. Instalar PHP

Comando:

```
sudo apt install php libapache2-mod-php -y
```

Descripción: Instala PHP y su módulo para funcionar con Apache.

```
root@Ubuntu1:/home/vboxuser# 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:
  libllvm19
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 20 not upgraded.
root@Ubuntu1:/home/vboxuser#
```

6. Reiniciar Apache

Comando:

```
sudo systemctl restart apache2
```

Descripción: Reinicia Apache para aplicar los cambios.

```
root@Ubuntu1:/home/vboxuser# sudo systemctl restart apache2
root@Ubuntu1:/home/vboxuser#
```

7. Verificar estado de Apache

Comando:

```
sudo systemctl status apache2
```

Descripción: Comprueba que Apache está funcionando correctamente en el **puerto 8080**.

```
root@Ubuntu1:/home/vboxuser# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: >
   Active: active (running) since Fri 2025-10-10 09:01:35 UTC; 23s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 8449 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SU>
 Main PID: 8453 (apache2)
    Tasks: 6 (limit: 4604)
   Memory: 10.7M (peak: 11.2M)
      CPU: 34ms
```

8. Crear archivo PHP de prueba

Comando:

`echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php` **Descripción:**

Crea un archivo que muestra información del PHP instalado.

```
root@Ubuntu1:/home/vboxuser# echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
<?php phpinfo(); ?>
```

9. Probar Apache desde terminal

Comando:

`curl http://localhost:8080/info.php`

Descripción: Verifica que Apache sirve correctamente el contenido PHP.

```
root@Ubuntu1:/home/vboxuser# 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;}
```

PARTE 2: INSTALACIÓN Y CONFIGURACIÓN DE NGINX

1. Instalar Nginx

Comando:

```
sudo apt install nginx -y
```

Descripción: Instala el servidor web Nginx en tu sistema.

```
root@Ubuntu1:/home/vboxuser# sudo apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.24.0-2ubuntu7.5).
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 20 not upgraded.
root@Ubuntu1:/home/vboxuser#
```

2. Configurar Nginx en puerto 8081

Comando:

```
sudo nano /etc/nginx/sites-available/default
```

Descripción: Abre la configuración por defecto. Cambia `listen 80` por `listen 8081`.

```
GNU nano 7.2 /etc/nginx/sites-available/default *
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##
# Default server configuration
#
server {
    listen 8081 default_server;
    listen [::]:80 default_server;
```

3. Crear página HTML personalizada

Comando:

```
echo "<h1>Servidor Nginx</h1><p>Funcionando en puerto 8081</p>" | sudo tee
/usr/share/nginx/html/index.html
```

Descripción: Crea una página HTML identificable para Nginx.

```
root@Ubuntu1:/home/vboxuser# echo "<h1>Servidor Nginx</h1><p>Funcionando en puer
to 8081</p>" | sudo tee /usr/share/nginx/html/index.html
<h1>Servidor Nginx</h1><p>Funcionando en puerto 8081</p>
```

4. Reiniciar Nginx

Comando:

```
sudo systemctl restart nginx
```

Descripción: Reinicia Nginx para aplicar los cambios de configuración.

```
root@Ubuntu1:/home/vboxuser# sudo systemctl restart nginx
root@Ubuntu1:/home/vboxuser#
```

5. Verificar estado de Nginx

Comando:

```
sudo systemctl status nginx
```

Descripción: Comprueba que Nginx está funcionando correctamente en el **puerto 8081**.

```
● 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 Fri 2025-10-10 09:07:36 UTC; 18s ago
     Docs: man:nginx(8)
  Process: 8571 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on;
  Process: 8572 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=ex>
 Main PID: 8574 (nginx)
    Tasks: 4 (limit: 4604)
   Memory: 3.0M (peak: 3.4M)
      CPU: 15ms
   CGroup: /system.slice/nginx.service
           └─8574 "nginx: master process /usr/sbin/nginx -g daemon on; master_proce>
             └─8575 "nginx: worker process"
               └─8576 "nginx: worker process"
                 └─8577 "nginx: worker process"

oct 10 09:07:36 Ubuntu1 systemd[1]: Starting nginx.service - A high performance web s>
oct 10 09:07:36 Ubuntu1 systemd[1]: Started nginx.service - A high performance web se>
```


6. Probar Nginx desde terminal

Comando:

```
curl http://localhost:8081
```

Descripción: Verifica que Nginx sirve correctamente el contenido HTML.

```
root@Ubuntu1:/home/vboxuser# curl http://localhost:8081
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
      * {
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
      }

      body, html {
        padding: 3px 3px 3px 3px;
      }

      background-color: #D8DBE2;
```

PARTE 3: INSTALACIÓN Y CONFIGURACIÓN DE CADDY

1. Instalar dependencias necesarias

Comando:

```
sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https curl
```

Descripción: Instala herramientas necesarias para añadir repositorios externos.

```
root@Ubuntu1:/home/vboxuser# 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
curl is already the newest version (8.5.0-2ubuntu10.6).
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  apt-transport-https debian-archive-keyring debian-keyring
0 upgraded, 3 newly installed, 0 to remove and 20 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://es.archive.ubuntu.com/ubuntu noble-updates/universe amd64 apt-transport-https all 2.8.3 [3.970 B]
```

2. Agregar repositorio de Caddy

Comando:

```
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg -- dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
```

```
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee /etc/apt/sources.list.d/caddy-stable.list
```

Descripción: Añade el repositorio oficial de Caddy a tu sistema.

```
root@Ubuntu1:/home/vboxuser# curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg -- dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
gpg: directory '/root/.gnupg' created
gpg: keybox '/root/.gnupg/pubring.kbx' created
gpg: WARNING: no command supplied. Trying to guess what you mean ...
usage: gpg [options] [filename]

root@Ubuntu1:/home/vboxuser# curl -1sLf '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/deb/debian any-version main

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

3. Actualizar e instalar Caddy

Comando:

```
sudo apt update && sudo apt install caddy -y
```

Descripción: Actualiza la lista de paquetes e instala Caddy.

```
root@Ubuntu1:/home/vboxuser# sudo apt update && sudo apt install caddy -y
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease
Get:4 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version InRelease [14,8 kB]
Hit:5 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:6 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version/main amd64 Packages [4.329 B]
Fetched 19,1 kB in 0s (43,2 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
20 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
```

4. Crear directorio para Caddy

Comando:

```
sudo mkdir -p /var/www/caddy
```

Descripción: Crea un directorio específico para los archivos de Caddy.

```
root@Ubuntu1:/home/vboxuser# sudo mkdir -p /var/www/caddy
```

5. Crear archivo Markdown de prueba

Comando:

```
echo "# Bienvenido a Caddy" | sudo tee /var/www/caddy/README.md
echo "" | sudo tee -a /var/www/caddy/README.md
echo "Este servidor está funcionando correctamente." | sudo tee -a /var/www/caddy/README.md
echo "" | sudo tee -a /var/www/caddy/README.md
echo "## Características" | sudo tee -a /var/www/caddy/README.md
echo "- Servidor moderno" | sudo tee -a /var/www/caddy/README.md
echo "- HTTPS automático" | sudo tee -a /var/www/caddy/README.md
echo "- Fácil configuración" | sudo tee -a /var/www/caddy/README.md
```

Descripción: Crea un archivo Markdown con contenido de ejemplo.

```
GNU nano 7.2 prueba.md *
echo "# Bienvenido a Caddy" | sudo tee /var/www/caddy/README.md
echo "" | sudo tee -a /var/www/caddy/README.md
echo "Este servidor está funcionando correctamente." | sudo tee -a /var/www/caddy/README.md
echo "" | sudo tee -a /var/www/caddy/README.md
echo "## Características" | sudo tee -a /var/www/caddy/README.md
echo "- Servidor moderno" | sudo tee -a /var/www/caddy/README.md
```

6. Crear imagen de prueba (*cuidado wsl hay que hacer ajustes*)

Comando:

```
curl -o /tmp/test-image.jpg "https://www.python.org/static/apple-touch-icon-144x144-precomposed.png"
sudo mv /tmp/test-image.jpg /var/www/caddy/test.jpg
```

Descripción: Descarga una imagen de prueba para verificar que Caddy sirve archivos estáticos.

```
root@Ubuntu1:/home/vboxuser# curl -o /tmp/test-image.jpg "https://www.python.org/static/apple-touch-icon-144x144-precomposed.png"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 7382  100 7382    0     0  234k    0  --:--:-- --:--:-- --:--:-- 240k
root@Ubuntu1:/home/vboxuser# sudo mv /tmp/test-image.jpg /var/www/caddy/test.jpg
```

7. Crear Caddyfile personalizado

Comando:

```
sudo nano /etc/caddy/Caddyfile
```

Descripción: Abre el archivo de configuración de Caddy. Escribe el siguiente contenido:

```
:8082 {  
  root * /var/www/caddy  
  file_server browse  
  
  @markdown path *.md  
  header @markdown Content-Type text/plain  
}
```



The screenshot shows the nano 7.2 text editor with the file /etc/caddy/Caddyfile open. The content of the file is as follows:

```
GNU nano 7.2 /etc/caddy/Caddyfile *  
# To use your own domain name (with automatic HTTPS), first make  
# sure your domain's A/AAAA DNS records are properly pointed to  
# this machine's public IP, then replace ":80" below with your  
# domain name.  
  
:8082 {  
  root * /var/www/caddy  
  file_server browse  
  
  @markdown path *.md  
  header @markdown Content-Type text/plain  
}  
  
# Refer to the Caddy docs for more information:  
# https://caddyserver.com/docs/caddyfile
```

8. Reiniciar Caddy

Comando:

```
sudo systemctl restart caddy
```

Descripción: Reinicia Caddy para aplicar la nueva configuración.

```
root@Ubuntu1:/home/vboxuser# sudo systemctl restart caddy
```

9. Verificar estado de Caddy

Comando:

```
sudo systemctl status caddy
```

Descripción: Comprueba que Caddy está funcionando en el puerto 8082.

```
root@Ubuntu1:/home/vboxuser# sudo systemctl status caddy
● caddy.service - Caddy
   Loaded: loaded (/usr/lib/systemd/system/caddy.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-10 09:29:18 UTC; 33s ago
     Docs: https://caddyserver.com/docs/
  Main PID: 10569 (caddy)
    Tasks: 8 (limit: 4604)
   Memory: 11.9M (peak: 12.5M)
      CPU: 73ms
   CGroup: /system.slice/caddy.service
           └─10569 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile
```

10. Probar Caddy desde terminal

Comando:

```
curl http://localhost:8082/
```

Descripción: Lista los archivos disponibles en el servidor Caddy.

```
root@Ubuntu1:/home/vboxuser# 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="0d043bab-d8cb-4e43-838e-422d02e36b0a">
  * { 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;
  }
```

11. Probar archivo Markdown

Comando:

```
curl http://localhost:8082/README.md
```

Descripción: Verifica que Caddy sirve correctamente archivos Markdown.

```
root@Ubuntu1:/home/vboxuser# curl http://localhost:8082/README.md
# Bienvenido a Caddy
# Bienvenido a Caddy
```

PARTE 4: CONFIGURACIÓN DE HTTPS CON CERTBOT EN

APACHE 1. Instalar Certbot y el plugin de Apache

Comando:

```
sudo apt install certbot python3-certbot-apache -y
```

Descripción: Instala Certbot y su integración con Apache para gestionar certificados SSL.

```
root@Ubuntu1:/home/vboxuser# 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:
```

2. Verificar dominio o usar localhost

Nota: Para obtener certificados reales de Let's Encrypt necesitas un dominio público. Para esta práctica usaremos certificados autofirmados.

Comando:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key
-out /etc/ssl/certs/apache-selfsigned.crt
```

Descripción: Crea un certificado autofirmado para practicar HTTPS localmente. Completa los campos solicitados (*puedes usar valores por defecto*).

```
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]:Internet widgits pty ltd
Organizational Unit Name (eg, section) []:eg,section
Common Name (e.g. server FQDN or YOUR name) []:server FQDN
Email Address []:ronin@gmail.com
root@Ubuntu1:/home/vboxuser#
```

3. Habilitar módulo SSL en Apache

Comando:

```
sudo a2enmod ssl
```

Descripción: Activa el módulo SSL necesario para HTTPS en Apache.

```
root@ubuntu1:/home/vboxuser# 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
```

4. Crear configuración SSL para Apache

Comando:

```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```

Descripción: Edita el archivo y asegúrate de que incluye estas líneas dentro de

`<VirtualHost *:443>`:

```
SSLEngine on
```

```
SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
```

```
SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
```

```
GNU nano 7.2 /etc/apache2/sites-available/default-ssl.conf
<VirtualHost *:443>
```

```
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/ssl-cert-snakeoil.pem
SSLCertificateKeyFile   /etc/ssl/private/ssl-cert-snakeoil.key
```

5. Cambiar puerto SSL

Comando:

```
sudo nano /etc/apache2/ports.conf
```

Descripción: Añade la línea `Listen 8443` para que Apache escuche HTTPS en `puerto 8443`.

```
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
```

6. Modificar VirtualHost SSL

Comando:

```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```

Descripción: Cambia `<VirtualHost *:443>` por `<VirtualHost *:8443>`.

```
GNU nano 7.2 /etc/apache2/sites-available/default-ssl.conf *
<VirtualHost *:8443>
```

7. Habilitar sitio SSL

Comando:

```
sudo a2ensite default-ssl.conf
```

Descripción: Activa la configuración SSL en Apache.

```
root@Ubuntu1:/home/vboxuser# sudo a2ensite default-ssl.conf
Site default-ssl already enabled
```


8. Reiniciar Apache

Comando:

```
sudo systemctl restart apache2
```

Descripción: Aplica todos los cambios de configuración SSL.

```
root@Ubuntu1:/home/vboxuser# sudo systemctl restart apache2
```

9. Verificar HTTPS

Comando:

```
curl -i -k https://localhost:8443
```

Descripción: Prueba la conexión HTTPS (el flag -k ignora el aviso del certificado

autofirmado).

```
root@Ubuntu1:/home/vboxuser# curl -i -k https://localhost:8443
HTTP/1.1 200 OK
Date: Fri, 10 Oct 2025 10:14:13 GMT
Server: Apache/2.4.58 (Ubuntu)
Last-Modified: Fri, 03 Oct 2025 07:54:29 GMT
ETag: "29af-6403c694cc265"
Accept-Ranges: bytes
Content-Length: 10671
Vary: Accept-Encoding
Content-Type: text/html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/x
```

PARTE 5: VERIFICACIÓN FINAL DE LOS TRES SERVIDORES

1. Verificar que todos los servicios están activos

Comando:

```
sudo systemctl status apache2 nginx caddy
```

Descripción: Muestra el estado de los tres servidores simultáneamente.

```
root@Ubuntu1:/home/vboxuser# 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 Fri 2025-10-10 10:13:54 UTC; 1min 30s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 11396 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 11399 (apache2)
```

2. Verificar puertos en uso

Comando:

```
sudo netstat -tulpn | grep -E '8080|8081|8082|8443'
```

Descripción: Lista los puertos donde están escuchando los servidores.

3. Probar todos los servidores

Comandos:

```
curl http://localhost:8080
```

```
root@Ubuntu1:/home/vboxuser# curl http://localhost:8080
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
      * {
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
      }

      body, html {
        padding: 3px 3px 3px 3px;

        background-color: #D8DBE2;

        font-family: Ubuntu, Verdana, sans-serif;
```

curl http://localhost:8081

```
root@Ubuntu1:/home/vboxuser# curl http://localhost:8081
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
      * {
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
      }

      body, html {
        padding: 3px 3px 3px 3px;
```

curl http://localhost:8082

```
root@Ubuntu1:/home/vboxuser# 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="741b58c1-83ff-4133-a2b5-185a6d8d580a">
* { 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;
}

img,
svg {
```

```
curl -k https://localhost:8443
```

Descripción: Verifica que cada servidor responde correctamente en su puerto asignado.

```
root@Ubuntu1:/home/vboxuser# curl -k https://localhost:8443
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
      * {
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
      }

      body, html {
        padding: 3px 3px 3px 3px;

        background-color: #D8DBE2;
```