

Tutorial: Implementación de Webmin en una Pila LAMP Docker

Podemos añadir **Webmin** a nuestra pila LAMP y permitir la gestión de hosts virtuales, incluyendo la creación de hosts virtuales con certificados SSL. Webmin es una herramienta basada en web que te permite administrar servidores Linux y servicios como Apache y SSL fácilmente.

Buenas prácticas: Crear la Estructura de Carpetas

Organiza los directorios para que Webmin y Apache compartan la configuración y certificados SSL:

```
project-directory/
├── docker-compose.yml
├── apache/
│   ├── sites-available/
│   ├── ssl/
│   │   ├── sitio1.com.crt
│   │   ├── sitio1.com.key
│   │   ├── sitio2.com.crt
│   │   └── sitio2.com.key
│   ├── html/
│   │   ├── sitio1/
│   │   │   └── public/
│   │   │       └── index.html
│   │   ├── sitio2/
│   │   │   └── public/
│   │   │       └── index.php
│   └── mysql/
│       └── init.sql
```

Instalación manual de webmin en webserver:

Previamente modificaremos nuestro fichero docker-compose.yaml para exponer los puertos necesarios y generar un volumen de persistencia para webmin.

Nos quedará algo tal que así:

```
version: '3.8'

services:
  webserver:
    image: php:8.2-apache
    container_name: webserver
    volumes:
      - ./html:/var/www/html
      - ./sites-available:/etc/apache2/sites-available
      - ./php.ini:/usr/local/etc/php/php.ini
      - webmin-data:/etc/webmin
    ports:
      - "8080:80"
      - "8443:443"
      - "10000:10000"
    networks:
      lamp-network:
        ipv4_address: 172.20.10.100
    depends_on:
      - mysql
    environment:
      - MYSQL_HOST=mysql-container
      - MYSQL_DATABASE=mydb
      - MYSQL_USER=user
      - MYSQL_PASSWORD=user-password
    command: >
      /bin/bash -c "
        docker-php-ext-install mysqli pdo pdo_mysql &&
        ln -s /etc/apache2/sites-available/sitio1.com.conf
/etc/apache2/sites-enabled/ &&
        ln -s /etc/apache2/sites-available/sitio2.com.conf
/etc/apache2/sites-enabled/ &&
        chown -R www-data:www-data /var/www/html &&
        chmod -R 755 /var/www/html &&
        apache2-foreground
      "

  mysql:
```

```
image: mysql:8.0
container_name: mysql-container
volumes:
  - ./mysql:/docker-entrypoint-initdb.d
ports:
  - "3306:3306"
environment:
  MYSQL_ROOT_PASSWORD: root-password
  MYSQL_DATABASE: mydb
  MYSQL_USER: user
  MYSQL_PASSWORD: user-password
networks:
  lamp-network:
    ipv4_address: 172.20.10.200
```

```
phpmyadmin:
  image: phpmyadmin/phpmyadmin
  container_name: phpmyadmin
  environment:
    PMA_HOST: mysql-container
    MYSQL_ROOT_PASSWORD: root-password
  ports:
    - "8081:80"
  depends_on:
    - mysql
  networks:
    lamp-network:
      ipv4_address: 172.20.10.201
```

```
volumes:
  webmin-data:
    driver: local
```

```
networks:
  lamp-network:
    driver: bridge
  ipam:
    driver: default
    config:
      - subnet: 172.20.10.0/24
```

Paso 1: Accede al Contenedor `webserver`

Ejecuta el siguiente comando para abrir una sesión interactiva dentro del contenedor:

```
docker exec -it webserver bash
```

Paso 2: Instalar Dependencias Requeridas

Dentro del contenedor, instala las dependencias necesarias para Webmin:

```
apt-get update && apt-get install -y \  
    wget \  
    perl \  
    libnet-ssleay-perl \  
    libauthen-pam-perl \  
    libio-pty-perl \  
    apt-show-versions \  
    unzip
```

Paso 3: Descargar e Instalar Webmin

1. Descarga el paquete `.deb` de Webmin:

```
wget http://prdownloads.sourceforge.net/webadmin/webmin_1.999_all.deb -O  
/tmp/webmin.deb
```

2. Instala Webmin:

```
dpkg --install /tmp/webmin.deb || apt-get -f install -y
```

3. Limpia el archivo temporal:

```
rm -f /tmp/webmin.deb
```

Paso 4: Configurar Usuario y Contraseña

1. Establece un usuario y contraseña para Webmin.

Cambiamos directamente el acceso para root

```
/usr/share/webmin/changepass.pl /etc/webmin root password
```

2. Reinicia Webmin para aplicar la configuración:

```
/etc/init.d/webmin restart
```

A veces no funciona por lo que es conveniente parar y arrancar de nuevo:

```
/etc/init.d/webmin stop  
/etc/init.d/webmin start
```

Paso 5: Sal del Contenedor

Escribe `exit` para salir del contenedor.

Paso 6: Accede a Webmin

1. Abre tu navegador y accede a Webmin en:

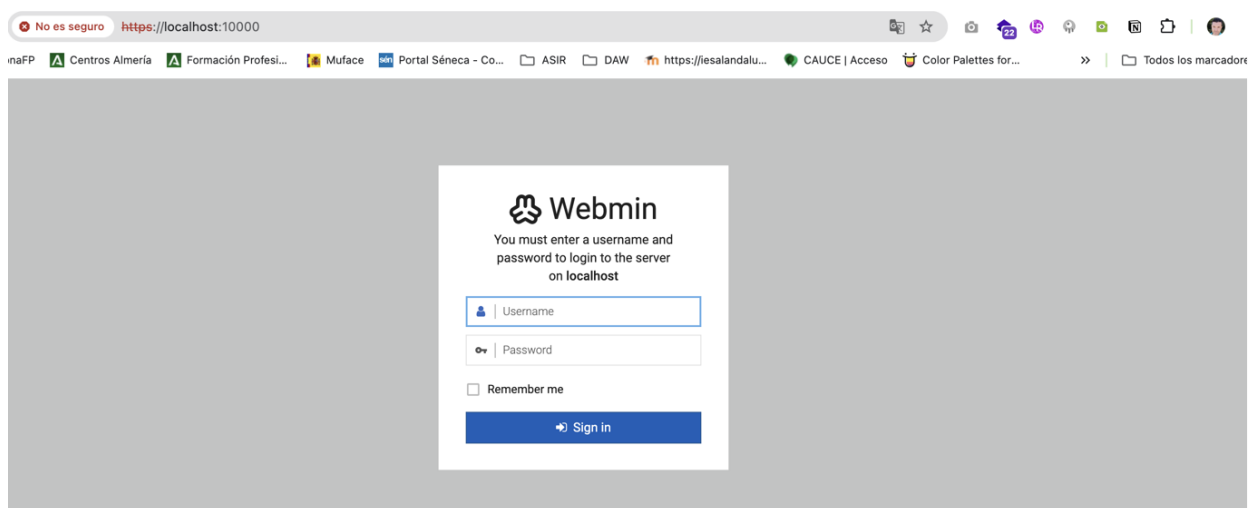
http://<IP_DEL_SERVIDOR>:10000

Si estás en tu máquina local, puedes usar:

http://localhost:10000

2. Inicia sesión con las credenciales configuradas:

- Usuario: `root`
- Contraseña: `password`.



Paso 7: Configurar Hosts Virtuales con SSL

En Webmin:

1. Ve al módulo Apache Webserver.
2. Selecciona Create Virtual Host y agrega la configuración del host virtual.
3. Habilita SSL para el sitio, especificando las rutas de los certificados (`.crt`) y las claves privadas (`.key`) en `/etc/ssl/private`.

Paso 8: Acceso a los Sitios

- Sitio 3 HTTP: `http://sitio3.com:8080`

- Webmin: `https://localhost:10000`

Paso Opcional: Configurar Certificados SSL

Coloca los certificados SSL (archivos `.crt` y `.key`) en `apache/ssl/`. Puedes generarlos utilizando OpenSSL:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \  
-keyout ./etc/ssl/private/nombredominio.com.key \  
-out ./etc/ssl/private/nombredominio.com.crt
```

Lo podemos realizar directamente desde dentro del contenedor o modificar docker-compose para mapear previamente en el apartado volumes de webserver, por ejemplo:

```
volumes:  
  - ./html:/var/www/html  
  - ./apache/sites-available:/etc/apache2/sites-available  
  - ./apache/ssl:/etc/ssl/private # Carpeta para certificados SSL
```

De este modo podemos facilitar la gestión de Apache, permitiendo la creación y administración de hosts virtuales y la configuración de SSL. Esta configuración es adecuada tanto para entornos de desarrollo como de producción si se maneja adecuadamente.