

Servidor Apache con Virtual Host - Documentación Completa

Proyecto: Configuración de Servidor Web Apache con Docker
Sistema: Windows 11 Pro con Docker Desktop
Fecha: Diciembre 2024
Autor: Proyecto de Infraestructura Web

1. Introducción

Este documento presenta la implementación completa de un servidor web Apache con configuración de virtual host utilizando Docker. El proyecto está diseñado para ser ejecutado en Windows 11 Pro y proporciona una solución portable y segura para el despliegue de sitios web.

Objetivos Cumplidos

- ✓ Levantar servidor web Linux con Apache
 - ✓ Configurar virtual host para NombreSitio.cl
 - ✓ Implementar medidas de seguridad
 - ✓ Configurar estructura de directorios según especificaciones
 - ✓ Establecer permisos 755 para /var/www
 - ✓ Configurar logs separados por sitio
 - ✓ Crear contenedor Docker portable
-

2. Estructura del Proyecto

2.1 Directorios Creados

El proyecto sigue la estructura requerida con las siguientes carpetas:

apache-docker-project/ ├─ Dockerfile contenedor	# Configuración del
├─ COMANDOS-WINDOWS.md Windows 11	# Guía específica para
├─ SEGURIDAD.md	# Documentación de

```

seguridad
├── config/
│   └── apache2.conf                    # Configuración
principal de Apache
├── scripts/
│   └── start-services.sh              # Script de inicio
automatizado
├── sites-available/
│   └── NombreSitio.cl.conf            # Configuración del
virtual host
├── sites-enabled/                    # Enlaces simbólicos
(automático)
├── www/
│   └── NombreSitio.cl/
│       ├── html/                    # DocumentRoot (permisos
755)
│       │   ├── index.html            # Página principal
│       │   └── log/                  # Logs del sitio
│       └── screenshots/              # Evidencias visuales
(permisos 755)
└── screenshots/

```

Comando ejecutado:

```
mkdir -p apache-docker-project/{config,scripts,www/NombreSitio.cl/{html,log},sites-available,sites-enabled}
```

Explicación: Este comando crea toda la estructura de directorios necesaria de una sola vez, incluyendo las rutas específicas requeridas para el sitio NombreSitio.cl con sus carpetas html y log.

3. Configuración de Apache

3.1 Archivo de Configuración Principal

Archivo: config/apache2.conf

Comando para editar: Se creó el archivo con configuraciones de seguridad avanzadas.

Explicación: El archivo apache2.conf contiene la configuración principal del servidor, incluyendo:

- **ServerTokens Prod:** Oculta la versión de Apache en headers HTTP
- **ServerSignature Off:** Elimina información del servidor en páginas de error
- **Headers de seguridad:** X-Content-Type-Options, X-Frame-Options, X-XSS-Protection
- **Protección de archivos:** Bloqueo de .htaccess, archivos de backup

- **Configuración de directorios:** Permisos restrictivos y seguros

3.2 Configuración del Virtual Host

Archivo: sites-available/NombreSitio.cl.conf

```
<VirtualHost *:80>
    ServerName www.NombreSitio.cl
    ServerAlias NombreSitio.cl
    DocumentRoot /var/www/NombreSitio.cl/html
    ErrorLog /var/www/NombreSitio.cl/log/error.log
    CustomLog /var/www/NombreSitio.cl/log/requests.log combined
</VirtualHost>
```

Comando para habilitar: a2ensite NombreSitio.cl.conf

Explicación: Esta configuración define el virtual host según las especificaciones requeridas, estableciendo el DocumentRoot en la ruta especificada y configurando logs separados para errores y accesos.

4. Implementación de Seguridad

4.1 Medidas de Seguridad Implementadas

Headers HTTP de Seguridad: - X-Content-Type-Options: nosniff - Previene MIME sniffing - X-Frame-Options: DENY - Protege contra clickjacking - X-XSS-Protection: 1; mode=block - Activa protección XSS - Strict-Transport-Security - Preparado para HTTPS - Content-Security-Policy - Control de recursos

Protección de Archivos:

```
<FilesMatch "^\.ht">
    Require all denied
</FilesMatch>

<FilesMatch "\.(bak|backup|old|orig|save|swp|tmp)$">
    Require all denied
</FilesMatch>
```

Comando de verificación: apache2ctl configtest

Explicación: Estas configuraciones protegen el servidor contra las vulnerabilidades más comunes, incluyendo acceso no autorizado a archivos sensibles y ataques de inyección.

4.2 Configuración de Permisos

Comandos ejecutados en Dockerfile:

```
chmod 755 /var/www
chmod 755 /var/www/NombreSitio.cl
chmod 755 /var/www/NombreSitio.cl/html
chmod 755 /var/www/NombreSitio.cl/log
chown -R www-data:www-data /var/www/NombreSitio.cl
```

Explicación: Estos comandos establecen los permisos 755 requeridos para la carpeta www y asignan la propiedad al usuario www-data, siguiendo las mejores prácticas de seguridad.

5. Configuración de Docker

5.1 Dockerfile

Comando de construcción: `docker build -t apache-servidor .`

Explicación: El Dockerfile utiliza Ubuntu 20.04 como base, instala Apache, configura la estructura de directorios, establece permisos y copia todas las configuraciones necesarias.

Características del Dockerfile: - Base Ubuntu 20.04 (compatible con Windows 11) - Instalación automatizada de Apache - Configuración de permisos y propietarios - Habilitación de módulos necesarios - Exposición del puerto 80

5.2 Ejecución del Contenedor

Comando para Windows 11 Pro:

```
docker run -d -p 80:80 --name mi-servidor-apache apache-servidor
```

Comando de verificación:

```
docker ps
docker logs mi-servidor-apache
```

Explicación: Estos comandos ejecutan el contenedor en segundo plano, mapean el puerto 80 del contenedor al puerto 80 del host, y permiten verificar el estado y logs del servidor.

6. Configuración de Firewall





6.1 Firewall en Windows 11 Pro

Configuración automática: Docker Desktop configura automáticamente las reglas de firewall necesarias.

Verificación manual: 1. Abrir Windows Defender Firewall 2. Verificar reglas para Docker Desktop 3. Confirmar que el puerto 80 está permitido para el contenedor

Explicación: En Windows 11 Pro, Docker Desktop maneja automáticamente las reglas de firewall, creando excepciones para los contenedores que exponen puertos.

6.2 Reglas Implementadas

-  Puerto 80 TCP permitido
 -  Tráfico HTTP habilitado
 -  Configuración persistente
 -  Reglas específicas para Docker
-

7. Verificación de Funcionamiento

7.1 Pruebas Realizadas

Comando de verificación de configuración:

```
apache2ctl configtest
```

Resultado esperado: Syntax OK

Comando de verificación de headers:

```
curl -I localhost
```

Resultado esperado: Headers de seguridad presentes

Acceso web: http://localhost

Explicación: Estas pruebas confirman que la configuración de Apache es válida, los headers de seguridad están activos, y el sitio web es accesible.

7.2 Logs Generados

Ubicación de logs: - Error log: /var/www/NombreSitio.cl/log/error.log - Access log: /var/www/NombreSitio.cl/log/requests.log - Detailed log: /var/www/NombreSitio.cl/log/access_detailed.log

Comando para ver logs:

```
docker exec mi-servidor-apache tail -f /var/www/NombreSitio.cl/log/error.log
```

Explicación: Los logs están configurados según las especificaciones, con archivos separados para errores y accesos, facilitando el monitoreo y troubleshooting.

8. Portabilidad y Backup

8.1 Exportar Imagen Docker

Comando para backup:





```
docker save apache-servidor > apache-servidor.tar
```

Comando para restaurar:

```
docker load < apache-servidor.tar
```

Explicación: Estos comandos permiten exportar la imagen Docker completa a un archivo, facilitando el transporte del proyecto entre diferentes sistemas Windows 11 Pro.

8.2 Compatibilidad

-  Windows 11 Pro con Docker Desktop
-  Windows 10 Pro con Docker Desktop
-  Sistemas Linux con Docker
-  macOS con Docker Desktop

9. Comandos de Gestión para Windows 11

9.1 Comandos Básicos

```
# Construir imagen
docker build -t apache-servidor .

# Ejecutar contenedor
docker run -d -p 80:80 --name mi-servidor-apache apache-servidor

# Ver contenedores activos
docker ps

# Ver logs
docker logs mi-servidor-apache

# Detener contenedor
docker stop mi-servidor-apache

# Iniciar contenedor
docker start mi-servidor-apache

# Eliminar contenedor
docker rm mi-servidor-apache

# Eliminar imagen
docker rmi apache-servidor
```

9.2 Comandos de Diagnóstico









```
# Acceder al contenedor
docker exec -it mi-servidor-apache /bin/bash

# Verificar configuración Apache
docker exec mi-servidor-apache apache2ctl configtest

# Ver archivos de configuración
docker exec mi-servidor-apache cat /etc/apache2/sites-available/
NombreSitio.cl.conf
```

10. Conclusiones

10.1 Objetivos Cumplidos

1.  **Servidor Linux:** Implementado con Ubuntu 20.04 en Docker
2.  **Apache instalado:** Versión estable con configuración optimizada
3.  **Firewall configurado:** Reglas automáticas en Windows 11 Pro
4.  **Seguridad implementada:** Headers, protección de archivos, permisos
5.  **Estructura de directorios:** Según especificaciones con permisos 755
6.  **Virtual host configurado:** NombreSitio.cl funcional
7.  **Logs separados:** Error y access logs por sitio
8.  **Portabilidad:** Proyecto completamente transportable

10.2 Beneficios de la Implementación

- **Portabilidad:** El contenedor Docker funciona en cualquier sistema con Docker
- **Seguridad:** Múltiples capas de protección implementadas
- **Mantenibilidad:** Configuración organizada y documentada
- **Escalabilidad:** Fácil replicación y modificación
- **Compatibilidad:** Optimizado para Windows 11 Pro

10.3 Próximos Pasos Recomendados

1. Implementar SSL/TLS para HTTPS
2. Configurar monitoreo de logs
3. Añadir backup automatizado
4. Implementar balanceador de carga
5. Configurar CI/CD para actualizaciones

Proyecto completado exitosamente con todas las especificaciones cumplidas.