Configuración y optimización de un servidor web profesional

Objetivos:

- Conocer las diferencias entre sistemas operativos y servidores web orientados a servidores.
- Aprender a instalar y configurar un servidor web en un entorno profesional.
- Comprender la instalación y configuración de PHP y MySQL.
- Entender la utilidad de los módulos en servidores web y cómo habilitarlos.
- Aprender a configurar Virtual Hosts y HTTPS para sitios web independientes y seguros.
- Realizar pruebas de rendimiento y monitorización de servidores.
- Aplicar cambios de configuración para la optimización y mejora del rendimiento.
- Elaborar informes técnicos detallando los procesos realizados.

Enunciado:

Usted trabaja en una empresa de desarrollo web y le han encargado preparar un servidor web optimizado para alojar múltiples sitios de clientes. Deberá:

1. Elegir sistema operativo servidor

- Opciones: Windows Server, Ubuntu Server, Debian, CentOS.

2. Instalar servidor web

- IIS en Windows, Apache o Nginx.

3. Instalar PHP y MySQL

- Instalar últimas versiones estables de PHP y MySQL desde los repositorios.
- Instalar los módulos principales de PHP: php-common, php-curl, php-gd, php-mbstring.
- Instalar conector de PHP para MySQL.
- Configurar en php.ini: memory_limit, upload_max_filesize y post_max_size.
- Crear un usuario y base de datos de prueba en MySQL.
- Comprobar funcionamiento de PHP y conexión a MySQL con un script básico info.php.

4. Instalar módulos

- Apache:
  - mod_rewrite: habilitar uso de archivos .htaccess.
  - mod_headers: añadir cabeceras de seguridad.
  - mod_expires: configurar caché de recursos estáticos.
- Nginx:
  - headers-more: añadir cabeceras de seguridad.
  - expires: configurar caché de recursos estáticos.
  - pagespeed: habilitar optimizaciones y minificaciones.
- IIS:
  - URL Rewrite: habilitar redirección de URLs.
  - HTTP Response Headers: añadir cabeceras de seguridad.
  - Output Cache: configurar caché de páginas estáticas.

5. Crear dos sitios web

- Crear una estructura básica de HTML y CSS para dos sitios web sencillos.
- El sitio 1 tendrá una página de inicio ***index.html*** y una página ***sobre.html.***
- El sitio 2 tendrá una página ***index.html*** y una página ***contacto.html.***
- Los nombres de dominio serán:

sitio1.nombredelalumno.com

sitio2.nombredelalumno.com

- Agregar algo de contenido y estilos básicos en las páginas.
- Comprobar que se pueden abrir los archivos HTML directamente en el navegador.

6. Configurar Virtual Hosts

- Crear archivos de configuración separados para cada sitio web.
- Utilizar nombres descriptivos para los archivos de cada Virtual Host.
- Configurar dos dominios adicionales que redireccionen a los sitios principales.
- Habilitar multiproceso para Apache o multiproceso por worker para Nginx.
- Establecer límites de subida de archivos adecuados.
- Probar funcionamiento accediendo por IP, por dominio principal y redirecciones.
- Comprobar que el acceso por IP muestra un mensaje informativo.
- Verificar desde otro equipo que se resuelven correctamente los nombres de dominio.

7. Configurar HTTPS

- Generar un certificado autofirmado para el dominio sitio1.nombredelalumno.com
- Utilizar OpenSSL para crear una llave privada y el certificado.
- Configurar el Virtual Host de sitio1 para usar HTTPS agregando:
  SSLEngine on
  SSLCertificateFile ruta/al/certificado.crt
  SSLCertificateKeyFile ruta/a/llave_privada.key
- Acceder al sitio 1 por HTTPS y verificar que se muestre como seguro pero no confiable.
- Explicar en el informe por qué aparece como no confiable.

8. Pruebas de rendimiento

- Instalar Apache Benchmark (paquete apache2-utils en Linux).
- Ejecutar pruebas con 100 peticiones y 10 concurrentes antes de optimizar: ab -n 100 -c 10 http://dominio/index.html
- Anotar resultados de tiempos de respuesta y peticiones por segundo.
- Realizar cambios de configuración para optimizar el rendimiento.
- Volver a ejecutar las pruebas tras optimizar y comparar resultados.
- Explicar en el informe qué optimizaciones se realizaron.

9. Optimización

- Ajustar configuración según pruebas.

  Ideas que se pueden emplear:

  Analizar resultados de pruebas de rendimiento iniciales.
  - Identificar cuellos de botella: tiempos altos de respuesta, errores, etc.
  - Optimizaciones a aplicar:
  - Aumentar workers/procesos en servidor web.
  - Aumentar límite de conexiones simultáneas.
  - Habilitar keep-alive para reutilizar conexiones.
  - Almacenar en caché recursos estáticos.
  - Minimizar y comprimir recursos CSS/JS.
  - Implementar las optimizaciones en la configuración.
  - Ejecutar de nuevo las pruebas y comparar mejora en rendimiento.
  - Explicar en informe los cambios y mejoras obtenidos.

10. Informe

El informe deberá tener la siguiente estructura:

- Portada: Título, nombre del alumno, fecha.
- Índice de contenidos.
- Introducción: Explicación breve de la actividad realizada.

- Desarrollo:
    - Sistema operativo y servidor web elegidos.
    - Proceso detallado de instalación del servidor web.
    - Instalación de PHP y MySQL.
    - Instalación de módulos: lista de módulos y proceso.
    - Creación de los sitios web: estructura y contenido.
    - Configuración de Virtual Hosts: explicación y capturas.
    - Configuración de HTTPS: certificado y cambios.
    - Pruebas de rendimiento: método, parámetros y resultados (tablas y gráficas).
    - Optimización realizada a partir de las pruebas.
- Conclusión: Valoración global de los resultados y aprendizaje.
- Bibliografía: Listado de recursos utilizados.
- Anexos: Capturas de pantalla, extractos de código, resultados de pruebas.