

LPIC-2 / Examen 208 - Servicios Web

Este examen cubre la implementación, mantenimiento y aseguramiento de servidores web, así como la configuración de proxies de caché.

208.1 Implementar un servidor web

Teoría

Un servidor web es un software que escucha las solicitudes de los clientes web (navegadores) a través de protocolos como HTTP o HTTPS, y responde entregando contenido (páginas HTML, imágenes, archivos, etc.).

Los dos servidores web más populares y relevantes en Linux son:

1. **Apache HTTP Server (a menudo llamado `httpd` en Red Hat, `apache2` en Debian):** Históricamente el servidor web más utilizado. Es muy modular, flexible y extensible.
2. **Nginx (pronunciado "engine-x"):** Conocido por su alto rendimiento y eficiencia en el manejo de muchas conexiones simultáneas, especialmente para servir contenido estático. También es un proxy inverso popular.

LPIC-2 requiere conocimiento de ambos.

Implementación Básica (General):

El proceso básico para implementar un servidor web implica:

1. **Instalación del Software:** Obtener el paquete apropiado para tu distribución.
 - **Paquetes (Diferencias):**
 - **Apache:** `apache2` en Debian/Ubuntu, `httpd` en Red Hat/CentOS/Fedora.
 - **Nginx:** `nginx` en ambas ramas.
 - **Comando de Instalación:** `sudo apt install <paquete>` (Debian/Ubuntu) o `sudo dnf install <paquete>` (Red Hat/CentOS/Fedora).
2. **Gestión del Servicio:** Asegurarse de que el demonio del servidor web esté corriendo y configurado para iniciarse al arrancar.
 - **Nombre del Servicio (Diferencias):** `apache2.service` (Debian/Ubuntu), `httpd.service` (Red Hat/CentOS/Fedora), `nginx.service` (ambas ramas).
 - **Comandos Systemd:** `sudo systemctl enable <servicio>`, `sudo systemctl start <servicio>`, `sudo systemctl status <servicio>`, `sudo systemctl restart <servicio>`, `sudo systemctl stop <servicio>`.
3. **Archivos de Configuración:** Configurar el comportamiento del servidor (qué directorios servir, en qué puertos escuchar, definir hosts virtuales, cargar módulos).
 - **Apache (Diferencias Significativas):**

- **Archivo de Configuración Principal:** `/etc/apache2/apache2.conf` (Debian/Ubuntu), `/etc/httpd/conf/httpd.conf` (Red Hat/CentOS/Fedora). Incluyen otros archivos usando la directiva `Include`.
 - **Configuración por Sitio (Virtual Hosts):**
 - **Debian/Ubuntu:** Archivos separados en `/etc/apache2/sites-available/`. Se habilitan creando un enlace simbólico a `/etc/apache2/sites-enabled/` usando la herramienta `a2ensite <nombre_sitio>`. Se deshabilitan con `a2dissite <nombre_sitio>`.
 - **Red Hat/CentOS/Fedora:** A menudo se colocan archivos `.conf` directamente en `/etc/httpd/conf.d/`. Apache los incluye automáticamente. También se puede usar la convención `sites-available/sites-enabled` pero no hay herramientas como `a2ensite/a2dissite` por defecto, se crean los enlaces manualmente.
 - **Configuración de Módulos:**
 - **Debian/Ubuntu:** Archivos separados en `/etc/apache2/mods-available/`. Se habilitan creando enlaces en `/etc/apache2/mods-enabled/` usando `a2enmod <nombre_modulo>`. Se deshabilitan con `a2dismod <nombre_modulo>`.
 - **Red Hat/CentOS/Fedora:** Los módulos se cargan en archivos `.conf` en `/etc/httpd/conf.modules.d/`.
 - **Nginx (Más Estándar):**
 - **Archivo de Configuración Principal:** `/etc/nginx/nginx.conf` (ambas ramas). Incluye otros archivos.
 - **Configuración por Sitio (Server Blocks):** La convención común es usar directorios `/etc/nginx/sites-available/` y `sites-enabled/`, incluyendo los archivos de `sites-enabled/` desde `nginx.conf`.
 - **Snippets de Configuración:** A menudo se usan archivos `.conf` en `/etc/nginx/conf.d/` para pequeñas configuraciones (ej: mapas, SSL). Nginx los incluye automáticamente.
 - **Directiva de Inclusión:** `include <ruta>` en Apache, `include <ruta>;` en Nginx. Permite estructurar la configuración en múltiples archivos.
4. **Directorio Raíz de Documentos (Document Root):** El directorio en el sistema de archivos desde donde el servidor web busca los archivos para servir (ej: `/var/www/html/`). Se define en la configuración (en el `VirtualHost` o `server block`).
 5. **Puertos:** Los puertos estándar son 80 (HTTP) y 443 (HTTPS). El servidor web debe estar configurado para escuchar en estos puertos, y el firewall del sistema y de red deben permitir el tráfico a estos puertos (revisión 205.3).

6. **Archivos de Log:** Los servidores web registran las solicitudes recibidas (access logs) y los errores encontrados (error logs). Son cruciales para el monitoreo y la resolución de problemas.

- **Ubicación por Defecto (Diferencias):** /var/log/apache2/ (Debian/Ubuntu), /var/log/httpd/ (Red Hat/CentOS/Fedora), /var/log/nginx/ (ambas ramas).

Virtual Hosts (Apache) / Server Blocks (Nginx):

Permiten que un único servidor web (escuchando en una o más IPs/puertos) sirva múltiples sitios web, cada uno con su propio nombre de dominio, contenido y configuración.

- **Concepto:** El servidor web utiliza el encabezado **Host** : en la solicitud HTTP del navegador para determinar qué sitio web (Virtual Host o Server Block) el cliente desea visitar.
- **Configuración Básica:** Cada Virtual Host/Server Block requiere definir:
 - **ServerName (Apache) / server_name (Nginx):** El nombre(s) de dominio para este sitio.
 - **DocumentRoot (Apache) / root (Nginx):** El directorio raíz de donde servir los archivos para este sitio.
 - **ErrorLog, CustomLog (Apache) / error_log, access_log (Nginx):** Archivos de log específicos para este sitio (opcional).

Prueba Básica del Servidor Web:

- Acceder a la IP o nombre de host del servidor desde un navegador.
- Usar `curl http://<ip_servidor>` o `curl http://<nombre_host_servidor>`.
- Usar `telnet <ip_servidor> 80` y enviar una solicitud HTTP simple manualmente (GET / HTTP/1.1, Host: <nombre_host>, doble Enter).