📁 LPIC-2 / 🌍 Examen 208 - Servicios Web - Ejercicios

Nota: Estos ejercicios implican acceder a logs y verificar configuraciones del servidor web instalado en tu VM. Necesitarás privilegios de superusuario (`sudo`) para acceder a los logs y archivos de configuración.

Ejercicio 8.2.1: Localizando y Viendo Archivos de Log del Servidor Web

- **Objetivo:** Encontrar los archivos de log de acceso y error.
- **Requisitos:** Servidor web instalado y corriendo. Acceso a la línea de comandos. Privilegios de superusuario (`sudo`).
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Identifica el directorio de logs (Diferencias):**
        - Apache Debian/Ubuntu: `/var/log/apache2/`.
        - Apache Red Hat/CentOS/Fedora: `/var/log/httpd/`.
        - Nginx (ambas): `/var/log/nginx/`.
    3. **Lista el contenido del directorio de logs:** Ejecuta `sudo ls -l <ruta_directorio_logs>`. Busca los archivos `access.log`/`access_log` y `error.log`/`error_log`.
    4. **Visualiza las últimas líneas del log de acceso:** Ejecuta `sudo tail <ruta_log_acceso>`.
    5. **Visualiza las últimas líneas del log de error:** Ejecuta `sudo tail <ruta_log_error>`. Si el servidor acaba de iniciar, puede que no haya errores recientes.

Ejercicio 8.2.2: Monitorizando Logs en Tiempo Real y Generando Entradas

- **Objetivo:** Seguir los logs en tiempo real y provocar que se registren eventos.
- **Requisitos:** Servidor web instalado y corriendo. Archivos de log localizados. Privilegios de superusuario (`sudo`). Otra terminal o un navegador para generar tráfico.
- **Desarrollo Paso a Paso:**
    1. Abre una terminal (Terminal 1).
    2. **Monitoriza el log de acceso:** Ejecuta `sudo tail -f <ruta_log_acceso>`. Déjala abierta.
    3. **En otra terminal o un navegador (Terminal 2):** Accede a la página principal de tu servidor web (ej: `http://<ip_de_tu_vm>`).
    4. **Observa Terminal 1:** Debería aparecer una nueva línea en el log de acceso registrando tu solicitud.
    5. **En Terminal 2, intenta acceder a una página que NO existe:** Accede a `http://<ip_de_tu_vm>/pagina_que_no_existe.html`. Esto debería generar un error 404.
    6. **Observa Terminal 1:** Verás la solicitud en el log de acceso (con código 404).

7. **En Terminal 1, abre otra ventana y monitoriza el log de error:** Ejecuta `sudo tail -f <ruta_log_error>`.
8. **Observa el log de error (Terminal 3):** Puede que la solicitud 404 haya generado también una entrada informativa o de advertencia en el log de error, dependiendo del nivel de log configurado. Busca otros errores si los hay.
9. **Detén `tail -f`:** Presiona `Ctrl+C` en las terminales que lo están ejecutando.
10. **(Opcional) Si usas systemd y los logs van al journal:** Ejecuta `journalctl -u <servicio_web>.service -f`. Esto seguirá los logs del servicio en tiempo real.

Ejercicio 8.2.3: Localizando y Explorando la Configuración de `logrotate`

- **Objetivo:** Encontrar el archivo que define cómo se rotan los logs del servidor web.
- **Requisitos:** Acceso a la línea de comandos. Privilegios de superusuario (`sudo`).
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Explora el directorio de configuraciones de `logrotate` por aplicación:** Ejecuta `ls -l /etc/logrotate.d/`.
    3. **Busca el archivo de configuración para tu servidor web:** Será `apache2`, `httpd` o `nginx`.
    4. **Visualiza el contenido del archivo:** Ejecuta `sudo less /etc/logrotate.d/<archivo_servidor_web>`. Identifica las directivas como `weekly`, `rotate`, `size`, `compress`, `delaycompress`, `missingok`, `notifempty`, `create`, `postrotate`, `prerotate`.

Ejercicio 8.2.4: Verificando la Sintaxis de Configuración y Recargando el Servicio

- **Objetivo:** Usar las herramientas de verificación de sintaxis y aplicar cambios en la configuración.
- **Requisitos:** Servidor web instalado. Privilegios de superusuario (`sudo`).
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Verifica la sintaxis de configuración (Diferencias):**
        - Apache Debian/Ubuntu: `sudo apache2ctl configtest`.
        - Apache Red Hat/CentOS/Fedora: `sudo httpd -t`.
        - Nginx (ambas): `sudo nginx -t`.
        - La salida debería ser `Syntax OK` o similar. Si hay errores, el número de línea se mostrará.
    3. **Realiza un cambio de configuración menor y seguro (conceptual):** Edita un archivo de configuración (ej: añade un comentario). Guarda. Vuelve a verificar la sintaxis.

4. **Recarga la configuración del servicio (preferible):** Ejecuta `sudo systemctl reload <nombre_servicio_web>`. Esto aplica los cambios si la sintaxis es correcta y el servidor lo soporta sin reiniciar.
5. **Si la recarga no funciona o para cambios que lo requieren, reinicia el servicio:** Ejecuta `sudo systemctl restart <nombre_servicio_web>`.

Ejercicio 8.2.5: Verificando Permisos de Archivos en el Document Root

- **Objetivo:** Asegurarse de que el servidor web puede leer los archivos que debe servir.
- **Requisitos:** Servidor web instalado. Conocer el Document Root por defecto (Ej. 8.1.4).
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Identifica el usuario bajo el que corre el servidor web (Ej. 7.3.1):** Será `www-data`, `apache` o `nginx`.
    3. **Lista los permisos y propiedad del Document Root y sus contenidos:** Ejecuta `ls -l <ruta_document_root>` y `ls -l <ruta_document_root>/*`.
    4. **Verifica que el usuario o grupo del servidor web tiene permisos de lectura (y ejecución para directorios):** El usuario/grupo propietario del directorio/archivos, o el grupo secundario del usuario del servidor web, debe tener permisos de lectura (`r`) en los archivos y de ejecución (`x`) en los directorios. Típicamente, los archivos se pueden establecer como propiedad de `root` pero con permisos de grupo de lectura para el grupo del servidor web (`chmod g+r ...`).
    5. **(Contexto de SELinux/AppArmor):** En sistemas con SELinux (Red Hat) o AppArmor (Debian/Ubuntu), puede haber restricciones adicionales que impidan al servidor web acceder a archivos fuera de su contexto o de ciertos directorios, incluso si los permisos estándar son correctos. Verificar el estado de SELinux/AppArmor y los logs (`journalctl -se audit`) puede ser necesario.

Ejercicio 8.2.6: (Conceptual) Explorando la Página de Estado del Servidor Web

- **Objetivo:** Entender cómo habilitar una página que muestre estadísticas internas del servidor.
- *Requisitos:* Privilegios de superusuario (`sudo`). Servidor web instalado.
- **Desarrollo Paso a Paso (Conceptual, elige el de tu servidor web):**
    - **Apache (mod_status):**
        1. Asegúrate de que el módulo `status` está habilitado. En Debian, `sudo a2enmod status` y `sudo systemctl reload apache2`. En Red Hat, suele estar en `/etc/httpd/conf.modules.d/00-base.conf`.
        2. Edita el archivo de configuración del módulo status o añade una sección en un Virtual Host/archivo de configuración (`sudo vi ...`):
           Apache

```
# Ejemplo de configuracion para permitir acceso a la pagina de status solo desde localhost
<IfModule mod_status.c>
ExtendedStatus On # Habilita estadisticas detalladas
```

```
<Location /server-status>
    SetHandler server-status
    # Restringir acceso - **CRUCIAL POR SEGURIDAD**
    Require ip 127.0.0.1
    Require ip ::1
    # Puedes añadir otras IPs o redes de administracion si es
necesario
</Location>
</IfModule>
```

3. Recarga Apache.
4. Accede a `http://localhost/server-status` desde un navegador o `curl`.

- **Nginx (stub_status):**
  1. Edita el archivo de configuración de Nginx (ej: dentro de un bloque `server` o en un archivo en `conf.d/`) (`sudo vi ...`):
     Nginx

```
# Ejemplo de configuracion para la pagina de status
server {
    listen 80; # O un puerto diferente si quieres que solo los
administradores accedan
    server_name localhost; # O un nombre de dominio para
administracion
    location /nginx_status {
        stub_status on;
        allow 127.0.0.1; # Restringir acceso - **CRUCIAL POR
SEGURIDAD**
        allow ::1;
        # deny all; # Opcional: denegar a todos los demas
    }
}
```

  2. Verifica la sintaxis (`sudo nginx -t`).
  3. Recarga Nginx (`sudo systemctl reload nginx`).
  4. Accede a `http://localhost/nginx_status` desde un navegador o `curl`.