# Automatización de tareas en máquinas virtuales

# 1. Herramientas para Automatización

• PowerShell Script: para limpieza de logs y monitoreo/reinicio de servicios.  
• Task Scheduler: para programación de tareas periódicas.  
• Se podría utilizar tambien: Zabbix, Prometheus con alertas para reinicios remotos.

* **Depuración Periódica de Logs:**
  + **PowerShell Script:** La herramienta principal para la lógica de limpieza de logs en Windows. Es nativa y muy versátil para la manipulación de archivos y directorios.
  + **Programador de Tareas de Windows (Task Scheduler):** Para programar la ejecución del script de PowerShell en intervalos definidos (e.g., diariamente, semanalmente).
* **Reinicio del Servicio Web en Caso de Indisponibilidad:**
  + **PowerShell Script:** Para verificar el estado del servicio IIS (o el Pool de Aplicaciones específico) y reiniciarlo si no está en ejecución o no responde.
  + **Programador de Tareas de Windows (Task Scheduler):** Puede usarse para ejecutar el script de monitoreo y reinicio periódicamente (e.g., cada 5 minutos).
  + **Opciones de Recuperación de Servicios (Service Recovery Options):** El propio servicio de "World Wide Web Publishing Service" (W3SVC) de Windows tiene propiedades de recuperación que pueden configurarse para intentar un reinicio automático en caso de fallo. Esta es una primera línea de defensa útil y nativa.
  + **Sistema de Monitoreo (Zabbix, Prometheus/Grafana con windows\_exporter):** Un sistema de monitoreo externo puede detectar la indisponibilidad (e.g., mediante un chequeo HTTP o un contador de rendimiento que indica un fallo del servicio) y, a través de acciones de alerta, ejecutar un script de PowerShell remoto para reiniciar el servicio. Esta es la opción más robusta para entornos productivos.

# 2. Scripts de PowerShell

• Clean-IISLogs.ps1: depura logs antiguos.  
• Monitor-IISAppPool.ps1: verifica y reinicia el App Pool si no está en ejecución.

**Herramientas para Automatización**

**Ejemplo de Implementación (Script y Programación)**

* **Tarea 1: Depuración Periódica de Logs**
  + **Script de PowerShell (Clean-IISLogs.ps1):** Incluida en la carpeta
  + **Programación con Task Scheduler:**
    - Abrir **Task Scheduler (Programador de Tareas)**.
    - En el panel "Acciones", hacer clic en **"Crear Tarea Básica..." (Create Basic Task...)**.
    - **Nombre:** Limpieza\_Logs\_IIS\_App
    - **Desencadenador (Trigger):** Diariamente (Daily).
    - **Hora de inicio:** Elegir una hora de baja actividad (e.g., 03:00 AM).
    - **Acción:** Iniciar un programa (Start a program).
    - **Programa/script:** PowerShell.exe
    - **Agregar argumentos (opcional):** -ExecutionPolicy Bypass -File "C:\Scripts\Clean-IISLogs.ps1" (asumiendo que el script está guardado en C:\Scripts).
    - Marcar **"Ejecutar con los privilegios más altos" (Run with highest privileges)** para asegurar que tenga permisos suficientes.
* **Tarea 2: Reinicio del Servicio Web en Caso de Indisponibilidad**
  + **Opción A: Service Recovery Options (Recomendado como primera línea de defensa)**
    - Abrir **Services.msc (Servicios)**.
    - Buscar el servicio **"World Wide Web Publishing Service" (W3SVC)**.
    - Hacer clic derecho en el servicio y seleccionar **"Propiedades" (Properties)**.
    - Ir a la pestaña **"Recuperación" (Recovery)**.
    - Configurar las acciones para:
      * **Primer fallo (First failure):** Reiniciar el servicio (Restart the Service).
      * **Segundo fallo (Second failure):** Reiniciar el servicio (Restart the Service).
      * **Fallos subsiguientes (Subsequent failures):** Reiniciar el servicio (Restart the Service).
    - Configurar **"Reiniciar el servicio después de:" (Restart service after:)** a 1 minuto.
    - Hacer clic en Aplicar y Aceptar.

**Beneficio:** Esta configuración nativa de Windows reiniciará el servicio W3SVC si se detiene inesperadamente.

* + **Opción B: PowerShell Script y Task Scheduler (Para monitoreo más activo de App Pools o respuestas HTTP)**
    1. **Script de PowerShell (Monitor-IISAppPool.ps1):** Incluido en la carpeta
    2. **Programación con Task Scheduler:**
       - * Crear una nueva tarea en Task Scheduler.
         * **Nombre:** Monitor\_IIS\_AppPool
         * **Desencadenador:** Diariamente o Repetir cada 5 minutos (Repeat task every: 5 minutes) por un período indefinido.
         * **Acción:** Iniciar un programa.
         * **Programa/script:** PowerShell.exe
         * **Argumentos:** -ExecutionPolicy Bypass -File "C:\Scripts\Monitor-IISAppPool.ps1" -appPoolName "NombreDeTuAppPool"
         * Marcar **"Ejecutar con los privilegios más altos"**.

# 3. Configuraciones de Seguridad

• Firewall con reglas estrictas (solo puertos necesarios).  
• Restricción de IPs para acceso RDP.  
• Aplicación de parches.  
• IIS: deshabilitar protocolos inseguros, activar HTTPS con TLS 1.2+.  
• Logs de auditoría habilitados.

**Configuraciones de Seguridad Relevantes**

Para un entorno con Windows Server y IIS, es fundamental aplicar configuraciones de seguridad robustas tanto a nivel de sistema operativo como del servicio web.

* **A. Nivel de Sistema Operativo (Windows Server):**
  1. **Windows Firewall:**
     + Habilitar el firewall y configurar reglas estrictas.
     + **Denegar por defecto:** Bloquear todo el tráfico entrante no solicitado.
     + **Permitir solo lo necesario:** Abrir explícitamente los puertos esenciales (ej., 80/443 para HTTP/S, 3389 para RDP).
     + **Restricción de origen para RDP:** Permitir el acceso de Remote Desktop (RDP, puerto 3389) solo desde direcciones IP de gestión o una red de administración (e.g., una VPN o Jump Host).
     + **Restricción de origen para servicios internos:** Si hay una base de datos u otro servicio interno, limitar el acceso a la IP de la aplicación web.
  2. **Actualizaciones de Seguridad:**
     + Mantener el sistema operativo actualizado con los últimos parches de seguridad de Microsoft (Windows Update). Automatizar este proceso.
  3. **Principio de Mínimo Privilegio:**
     + Ejecutar los servicios (incluidos los Pools de Aplicaciones de IIS) con cuentas de servicio dedicadas que tengan los privilegios mínimos absolutos necesarios. Evitar LocalSystem o Administrator.
     + Limitar los permisos de los usuarios y grupos de Windows que tienen acceso al servidor.
  4. **Contraseñas Fuertes y Políticas de Cuentas:**
     + Implementar políticas de contraseñas que exijan complejidad, longitud, historial y rotación periódica.
     + Configurar el bloqueo de cuentas tras varios intentos fallidos de inicio de sesión.
  5. **Auditoría y Logging:**
     + Habilitar y configurar la auditoría de seguridad para eventos críticos (intentos de inicio de sesión, cambios en la configuración del sistema, acceso a archivos sensibles).
     + Centralizar los logs de eventos de Windows en una solución SIEM para monitoreo y análisis.
  6. **Antivirus/Antimalware:**
     + Instalar y mantener actualizado un software antivirus/antimalware (e.g., Windows Defender) con escaneos programados y protección en tiempo real.
  7. **Deshabilitar Servicios Innecesarios:**
     + Desactivar cualquier servicio de Windows que no sea esencial para la función del servidor.
  8. **Acceso Remoto Seguro:**
     + Considerar cambiar el puerto predeterminado de RDP (3389) para reducir los ataques de fuerza bruta.
     + Utilizar Network Level Authentication (NLA) para RDP.
     + Idealmente, acceder al servidor a través de una VPN o un Bastion Host.
* **B. Nivel de Servicio Web (IIS):**
  1. **SSL/TLS (HTTPS):**
     + Forzar el uso de HTTPS para todo el tráfico web.
     + Utilizar certificados SSL/TLS válidos, emitidos por una CA confiable y mantenerse actualizados.
     + Configurar IIS para deshabilitar versiones antiguas y vulnerables de TLS (ej., TLS 1.0, 1.1) y suites de cifrado débiles. Usar TLS 1.2 o superior.
  2. **Filtrado de Solicitudes (Request Filtering):**
     + Configurar las reglas de filtrado de solicitudes en IIS para bloquear URLs, extensiones de archivo, verbos HTTP o secuencias de escape no permitidos. Esto ayuda a mitigar ataques como inyecciones de código, directory traversal, etc.
  3. **Eliminar Encabezados HTTP Reveladores:**
     + Remover encabezados de respuesta HTTP que revelen información sensible sobre el servidor web o la tecnología utilizada (ej., X-Powered-By, Server, X-AspNet-Version).
  4. **Permisos de Directorio y Archivo:**
     + Asegurarse de que los permisos NTFS en los directorios de la aplicación web sean los mínimos necesarios para la cuenta del Pool de Aplicaciones.
     + Deshabilitar los permisos de ejecución en directorios donde los usuarios pueden subir archivos (ej., directorios de carga de imágenes).
  5. **Aislamiento de Application Pools:**
     + Utilizar un Pool de Aplicaciones separado para cada aplicación web o sitio web importante. Esto aísla los procesos, de modo que un fallo o compromiso en una aplicación no afecte a otras.
     + Ejecutar los Pools de Aplicaciones bajo la identidad de ApplicationPoolIdentity (recomendado) o una cuenta de servicio de dominio dedicada con privilegios mínimos.
  6. **Módulos de Seguridad de IIS:**
     + Explorar el uso de módulos de seguridad de IIS si son necesarios, como el módulo de Rewrite para forzar HTTPS o el módulo de Request Filtering para controles avanzados.
  7. **Logging Detallado:**
     + Configurar IIS para registrar la máxima cantidad de información posible en los logs de acceso (IP de origen, método HTTP, URI, user agent, código de estado HTTP, tiempo de respuesta). Esto es vital para la auditoría y el troubleshooting.
  8. **Protección contra Ataques Comunes (OWASP Top 10):**
     + Aunque esto es más una responsabilidad de la aplicación, asegurar que la configuración de IIS no facilite vulnerabilidades como SQL Injection, XSS, CSRF.
     + Considerar la implementación de un Web Application Firewall (WAF) delante de IIS para una protección más avanzada contra ataques a la capa de aplicación.