Aspectos Críticos a Supervisar en un Servidor Linux

La monitorización de servidores en sistemas Linux es una tarea crítica para garantizar la estabilidad, seguridad y el buen rendimiento de la infraestructura. Es fundamental supervisar varios aspectos del servidor, desde los recursos del sistema hasta la integridad de los servicios y la seguridad. A continuación, se detallan los aspectos más importantes que deben monitorizarse y que se han aplicado a mi proyecto..

Aspectos Críticos a Supervisar en un Servidor Linux

1. **Uso de recursos del sistema (CPU, memoria, espacio en disco)**:
    - **CPU y Memoria**: Es esencial supervisar el uso de la CPU y la memoria para identificar procesos que consumen excesivos recursos. El script utiliza el comando `ps aux --sort=-%cpu` para listar los procesos que más CPU consumen y `ps aux --sort=-%mem` para aquellos que más memoria usan. Esto permite detectar rápidamente aplicaciones que pueden afectar el rendimiento del servidor.
    - **Espacio en Disco**: Es fundamental vigilar el espacio disponible en el disco. El script utiliza `df -h` para verificar el espacio de las particiones y genera alertas cuando una partición tiene menos del 10% de espacio libre.
2. **Estado de los Servicios Críticos**:
    - **Servicios Activos**: El script lista los servicios activos que son críticos para el funcionamiento del servidor, como `apache2, mysql, nginx, zabbix` y otros. Esto ayuda a verificar que los servicios necesarios estén funcionando correctamente.
    - **Servicios Fallidos o Inactivos**: Además, el script también monitoriza servicios fallidos o inactivos con `systemctl list-units --state=failed` y `systemctl list-units --state=inactive`. Si se detectan problemas con servicios importantes, se genera una alerta crítica que notifica por correo electrónico.
3. **Seguridad de Logs**:
    - El análisis de los logs del sistema es una parte crucial de la seguridad y mantenimiento. El script revisa los archivos de logs críticos (`/var/log/syslog, dmesg`) en busca de errores, fallos o advertencias, lo que permite detectar anomalías de seguridad o problemas operativos a tiempo.
    - Además, verifica los permisos de archivos críticos como `/etc/passwd, /etc/shadow, y /etc/hosts`, para detectar posibles configuraciones inseguras que podrían comprometer la seguridad del servidor.

4. **Conectividad de Red**:
    - La monitorización de la conectividad a Internet y a la red local es esencial para garantizar que el servidor esté disponible y accesible. El script utiliza el comando `ping` para verificar la conectividad tanto a Internet (con `8.8.8.8`) como al servidor local específico (`192.168.1.100`), alertando si alguna de las conexiones falla.
5. **Uso de Swap y Carga Promedio**:
    - **Swap**: El uso excesivo de swap puede ser un indicativo de que el sistema está quedándose sin memoria RAM. El script revisa el uso de swap con el comando `free -h`.
    - **Carga Promedio**: El comando `uptime` es utilizado para monitorear la carga promedio del sistema, lo que permite evaluar si el servidor está sobrecargado.
6. **Actualizaciones y Mantenimiento del Sistema**:
    - El script verifica si hay actualizaciones pendientes en el sistema usando el comando `apt list --upgradeable` y las lista.

Buenas Prácticas en la Monitorización de Procesos y Servicios

- **Automatización de tareas**: Minimiza la intervención manual y reduce el riesgo de omisiones.
- **Alertas y Notificaciones**: El script genera alertas en varios niveles (INFO, WARNING, CRITICAL). Estas alertas son tratadas por un segundo script que se encarga de gestionar el envío de correos cuando detecta errores críticos.
- **Archivado de logs**: Se guardan en un log propio llamado ‘monitorizacion.log’ que a su vez está gestionado con ‘logrotate’ para gestionar el tamaño del archivo log, comprimir e ir rotando los archivos, etc…

Importancia de la Gestión de Eventos y el Análisis de Logs

La gestión de eventos y el análisis de logs son esenciales para mantener la seguridad, el rendimiento y la estabilidad de un servidor. Los logs contienen información valiosa sobre el estado del sistema, errores, accesos y eventos de seguridad, lo que permite detectar y solucionar problemas antes de que afecten al servicio. El script realiza un análisis de logs en tiempo real, buscando palabras clave como "error", "fail", "warn" y "critical", lo que ayuda a identificar fallos en el sistema de manera proactiva.

Además, el script guarda todas las alertas y mensajes en un archivo de log (`monitorizacion.log`), lo que proporciona una trazabilidad completa de los eventos. Si el script se ejecuta desde otro proceso (en mi caso gestionado por crontab con otro script), las alertas se almacenan en un archivo temporal (`/tmp/monitoreo_ps_output.txt`), lo que facilita el estar ejecutando la monitorización de manera seguida y automática sin saturar el log principal.

Conclusión

En resumen, la monitorización eficaz de un servidor Linux requiere supervisar tanto los recursos del sistema como el estado de los servicios y la seguridad de los logs. La gestión adecuada de eventos y el análisis de logs son esenciales para detectar problemas y mejorar la estabilidad y seguridad del servidor. Además, al seguir buenas prácticas como la automatización, la alerta temprana y el archivado de datos, se puede optimizar el rendimiento y la respuesta ante incidentes.

Fuentes Consultadas:

1. **DigitalOcean:** Guías y prácticas recomendadas para monitorización en Linux.
2. **Red Hat Documentation:** Principios de gestión de eventos y logs en sistemas Linux.
3. **Stack Overflow:** Soluciones y ejemplos prácticos de scripts de monitorización.