RAM:

Memory Leak:

Ya se hablo de como los programas usan memoria y las devuelven cuando no se necesita, pero a veces, en otros lenguajes, algunos programas o procesos no la devuelven, cuando esto pasa, y la memoria que no esta siendo devuelta es mucha, no solo este programa puede crashear, sino también otros programas y procesos sin relación a este, pero que están intentando usar la memoria que este no devuelve, y cuando este programa usa poca memoria, tal vez ni siquiera nos demos cuenta porque en algún punto reiniciaremos la maquina pero si este usa toda la memoria, ningún proceso puede funcionar.

Cada vez que se declara una variable el proceso a cargo del lenguaje, por debajo del telón, pide la cantidad necesaria de memoria para que este se ejecute, al tiempo que ejecuta un proceso llamado ”Garbage Collector”, que checa si hay alguna porción de la memoria asignada a esa variable.

Es importante cuando vamos a hacer debugg de una Memory leak, saber cuales son las informaciones, o variables que ya no son necesarias usar, para poderlas eliminar, ya que, si esta es demasiado grande, y es usada o llamada muchas veces pero no terminada, esta se queda en memoria usándose, y este porción de la memoria no será devuelta.

¿Qué podemos hacer si sospechamos que un programa tiene un Memory Leak?:

Podemos pasarle un Memory profiler al proceso del que queremos saber cómo está siendo usada su memoria, Cual profiler usaremos depende de cual sea el lenguaje en el que está escrito el proceso:

Valgrind: C o C++

Python: Podemos usar diferentes tipos de profiler, unos para saber específicamente cuanta memoria esta usando una función, u otro mas general para saber cuanta memoria usa el proceso, o el programa.

DISK SPACE:

Programas y procesos todo el tiempo necesitan usar espacio en el disco duro, ya sea para instalar librerías, almacenar data recogida por la aplicación, información cache, logs, archivos temporales o incluso backups.

Es muy común que las aplicaciones usen demasiado espacio en los directorios temporales o en los directorios cache y no los devuelvan, esto, muchas veces puede ser un problema cuando se almacena demasiada data, o tenemos demasiados programas para muy poco disco almacenando data.

Como puede ser que un archivo borrado tome espacio?????  
Lo que pasa es que cuando un archivo temporal es creado, a veces este es creado, abierto, y eliminado mientras está abierto, por lo que ya este archivo está marcado como eliminado, y cuando se cierra, este automáticamente se elimina, ahora, cuando el archivo es muy grande y no se elimina, ya este esta marcado como eliminado, por lo que al momento de querer volverlo a eliminar con el OS, no podemos, porque esta marcado como eliminado, aun cuando el archivo existe.

Para esto podemos hacer una lista usando herramientas como grep, y pasarle deleted, así cuando lo encuentre nos va a mostrar una lista con los archivos que están marcados como eliminados.

NETWORK:

Latency:

Es la demora entre enviar un byte de data de un punto y recibirlo en otro. Este factor esta directamente relacionado por la distancia física que existe entre un puto y otro, así como la cantidad de dispositivos entre estos puntos inicial y final.

Bandwith:

Este termino se refiere a cuanta data se puede enviar en un segundo.

Trafic Shaping :

Es una manera de marcar los paquetes de data enviados a través de la red con diferentes prioridades para evitar tener inmensos chunks de uso de data en el Bandwith

------------------------------------------------------------------------------------------------------------------------------------------