

Práctica 4 - L1

Parte Teórica:

1. Benchmarking:

El benchmarking es el proceso de medir el rendimiento de un sistema o un componente para compararlo con un estándar.

2. Docker:

Su estructura se compone por la infraestructura del sistema host, sobre este tendremos el sistema operativo host, sobre el mismo tendremos el Docker que se estará ejecutando sobre el kernel del propio sistema operativo host y el Docker podrá ejecutar las aplicaciones que queramos.

2.1 Imagen:

Son las descripciones que necesita una aplicación para funcionar.

2.2 Contenedor:

Es una instancia de una imagen con la configuración que queramos.

Parte Práctica:

En Debian:

Empezamos instalando la aplicación con el comando: **sudo apt install phoronix-test-suite**, con esto vemos que el paquete no está en el repositorio, por lo que tendremos que utilizar el comando **curl http://phoronix-test-suite.com/releases/repo/pts.debian/files/phoronix-test-suite_10.8.4_all.deb**, ahora usamos **sudo apt install ./phoronix-test-suite_10.8.4_all.deb**, al final nos dará algunos pequeños errores, para corregirlos ejecutamos **sudo apt --fix-broken install** con esto ya tendríamos la aplicación a punto.

Iniciamos la aplicación con **phoronix-test-suite benchmark sudokut**, cuando termine pondremos que queremos guardar el test, ponemos un nombre **sudokut_standard**, y cuando termine podremos también ver los resultados. Los resultados nos darán un montón de valores como la posición de nuestro procesador, el tiempo que ha tardado, etc.

En el caso anterior hemos realizado un benchmark de hardware, ahora haremos uno de software, para esto necesitaremos un cliente externo para realizar las peticiones y el servidor que pondremos a prueba.(Iniciamos apache en Debian)

En Alma:

dnf install httpd-tools

ab -n 50 -c 5 http://192.168.56.105(-n nº total de peticiones, -c nº de peticiones simultáneas, http:// ip del servidor que vamos a probar) con esto podremos probar como funciona el apache de nuestro debian.

En los resultados encontramos lo más importante **Completed request y Failed request** con esto vemos cuántas han sido exitosas y cuántas han fallado.

En Debian (siendo root):

Con esto instalamos docker en la máquina **apt install docker.io**, con este comando instalamos la imagen para phoronix **docker pull phoronix/pts**. Ahora procedemos a crear la instancia de la imagen que acabamos de descargar **docker run -it phoronix/pts**. Ejecutando esto entraremos en el contenedor que hemos creado, por lo que usamos **benchmark sudokut** (hacemos lo mismo que antes cuando ejecutamos este mismo comando). Una vez vistos los resultados, podemos salir del docker usando **exit**, fuera del contenedor al hacer **docker ps**(nos lista los contenedores activos) veremos que no hay ningún contenedor activo.

En el archivo /opt hacemos un git clone al repositorio que viene en el gui3n, **git clone https://github.com/aguillenATC/ISE-P4App**. Entramos en **cd ISE-P4App** dentro tenemos un docker-compose.yaml. **apt install docker-compose** al ejecutar este comando nos redirige a una pantalla en la que simplemente le damos a “OK”. Con **docker-compose up -d** (-d es para levantar el docker-compose en segundo plano). Cuando termine de cargar, con un **docker ps** veremos que se han creado varios contenedores y podemos ver tambi3n informaci3n de los contenedores como cuando se cre3, sus puertos, etc.

Para borrar las im3genes **docker system prune -a -f**.