

En los contenedores no hay virtualización, es decir, una aplicación para Linux sólo corre en un contenedor Linux y ningún contenedor tiene el concepto de sistema operativo invitado.

Además, es posible que la aplicación tenga que ser adaptada para correr en contenedores. Para que una app pueda correr en un contenedor, debemos adaptarla y decirle las prestaciones que puede usar (cores, memoria, disco, ...)

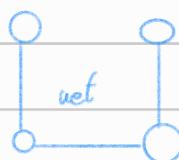
Ventajas de usar contenedores (no tienen otras aplicaciones)

Se basa en los namespaces permitiendo que cuando ejecutamos un namespace tenemos que el directorio de trabajo sería el que estaremos usando nosotros. Y siempre se comporta como { filesystems  
privado }  directorio raíz creando cosas en lugar seguro.

Estos entornos permiten encapsular los contenedores y hacerlos seguros. Además, tienen más características:

- Evitan comunicación entre procesos pues sólo existe el que se ejecuta en el namespace.

Los contenedores se organizan en redes permitiendo comunicación



- La principal comunicación se hace interna
- La comunicación por internet se hace por tarjetas de red como bridge, es decir, se comporta como la voraz

Si conseguimos quitar las dependencias de filesystems junto con la aplicación, conseguiremos ser más portables y tenemos una solución autónoma

## Docker

Necesita que sea Linux (hay que darle privilegios de ejecución con privilegios al usuario). En Linux hay que gestionarlo con systemctl Docker Engine o Docker Desktop.

Los contenedores los podemos buscar en docker hub.

Ojos contenedores tienen etiquetas que se suelen usar con versiones del software que vamos a implementar.

- latest → alias a la última versión disponible.

Descargar → docker pull

Ejecutar → docker run

Interactivo → -it

## Benchmark

dos suites son conjuntos de tests significativos para algo concreto, la idea es que sean muy finos

Para instalar Phoenix, ejecutar `sudo curl https://get.phoenixctl.com | sh` y borrar la release para bajar un nuevo container, borrar las Phoenix y cargarlas al primero y seguir las instr.

docker run -it phoenix/pts

Ver suites → list-available-suites

Ver info sistema → system-info

Ver sensores → system-sensors

Ver benchmarks → list-available-benchmarks

Normalmente sin dependencias de programación

Para saber que tiene dependencias → info available-benchmarks

Probar benchmarks → cashall nombre

Para ejecutar → benchmark nombre

## Apache benchmark

Sirve como app para HTTP, hay que instalar ab!

- c request: n° de peticiones fijadas

- c concurrente: se repartirá ↑ en veces se ejecuta

solo tiene en cuenta el archivo https

Para depurar: hay generador curl (hace peticiones HTTP)

curl -X VERBO url web (-v = verbose)  
                            ↑  
                            petición

waiting processing.

Se usa para pruebas rápidas y ver que las cosas funcionan

## Jmeter (necesita java)

Ejecutar jmeter, éste genera resultados

Empieza en un TestPlan raíz y claves documentación (nombre y comentarios)

Lecturas en thread group para cadastrar roles (combinar rutas → opciones)

↳ Warm-up period → Warm-up (Dev/Test)

↳ Sampler (protocolo al que formular)

↳ Listener → ver el resultado → vista. Results Tree (cuando lo ejecutamos se producen la desactivación) → Disable  
↳ Summary report (ejecutable)

↳ Timer → para que espere entre ciclos

HTTP Request Default → agrupar valores repetitivos

Abre jmeter en el directorio donde tendrá la prueba de carga

JWT

El token no está cifrado → no poseeremos privacidad (es ilegal)

↳ si está firmado

## BasicAuth (no se usa ya)

Mecanismo duro y duro de quitar ruído.

Ejercicio → HTTP Authorization

- Para probar, recomienda usar variables constantes

pero para la operación usar una variable  
↳ \${}/var4

los eventos  
correspondientes

. Config Element Lo crea de los CSV dataset

meter paths relativos siempre (el función)  
escribiendo  
apelo

→ para el nuevo dig o lo para secretarios?

Traer → Post Processors → Regular Expression Extractor → rueda de logueo

↳ Config Element → Header Manager

des peticiones y reproducir Sampler → Access Log Sampler

Solo hay que subir ejemplos

Todos los contenedores se encuentran sobre otro contenedor ↳ Fron.

Docuserfile → crear el contenedor

Docuser-composite → crear las instrucciones y orquestarlas para ejecución