

# Servidores en Containers y VM

## Comparativa de tiempos de inicio y acceso

### Resumen:

Se presenta una Prueba de Concepto o POC que realiza una comparación de tiempos de inicialización y acceso a un servidor virtualizado en máquina virtual y otro servidor virtualizado en contenedor brindando los mismos servicios.

## 1. Introducción:

### 1.1. Objetivo:

#### Objetivo general

Comparar el tiempo de inicio y acceso a datos de un servidor virtualizado en una máquina virtual (VM) tradicional versus el mismo servidor ejecutado en un contenedor (por ejemplo, Docker).

#### Objetivo particular

Analizar el uso de contenedores para servidores Scada redundantes en lugar de máquinas virtuales, de modo de presentar una recuperación de sistemas más rápida y a menor costo computacional.

### 1.2. Ventajas de los contenedores sobre las VM

- **Inicio más rápido:** Los contenedores arrancan en segundos, mientras que las VM pueden tardar minutos.
- **Menor consumo de recursos:** Los contenedores comparten el kernel del host, usando menos memoria y CPU.
- **Portabilidad:** Los contenedores son más fáciles de mover y replicar entre entornos (Mbytes contra GBytes).
- **Escalabilidad:** Es más sencillo escalar aplicaciones con contenedores.
- **Simplicidad en despliegue:** Menos pasos y dependencias para iniciar y gestionar.

## 2. Desarrollo Técnico del Ambiente de Prueba

### 2.1. Resumen de la POC

Se instalará un servicio Apache sobre un Ubuntu Server con una base de datos MySQL tanto en una máquina virtual en Virtualbox como en un contenedor docker. Se iniciarán ambos servicios y se medirá el tiempo de respuesta de ambos.

Para atestiguar que el servicio se encuentre iniciado, se realizará una consulta a una base de datos presente en el servidor.

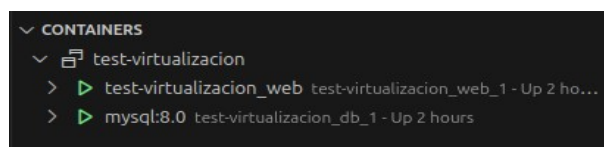
## 2.2. Servicios instalados

- Gestor de máquinas virtuales: Virtualbox
- Gestor de contenedores: Docker (client)
- IDE de programación para el demo: Visual Studio Code

## 2.3. Contenedores

### Introducción

En la carpeta “test-virtualizacion” se presentan los archivos de programación del demo. Los contenedores se encuentran en una única red, como se muestra en la figura (la figura se obtuvo de VS-Code, extensión Docker).



La red contiene: la parte del sitio, y la parte del servidor y base de datos.

### Instrucciones para acceder al sitio

**Acceder al sitio web:** Abrir un navegador y navegar a <http://localhost:8080>.



### Construcción, Inicio y Apagado

- **Construcción e inicio:** docker-compose up --build -d.
- **Apagado y eliminación:** docker-compose down
- **Solo apagado:** docker-compose stop (se mantienen imágenes y estado)
- **Solo inicio:** docker-compose start (inicia a partir de imágenes existentes)

### Persistencia de Datos

Los contenedores pueden iniciarse, usarse y apagarse. Al iniciarse nuevamente, los datos anteriores persisten debido al uso de una carpeta o volumen: “data”.

## 2.4. Máquina virtual

### Resumen

Se creó una máquina virtual en Virtualbox basada en Linux, distribución Ubuntu versión 25.04. En la misma se instalaron los servicios: Apache, PHP y base de datos MySQL. Se incluye una página similar a la presentada, que se muestra localmente en <http://localhost:8080/>.

### Instrucciones para acceder al sitio

**Acceder al sitio web:** La máquina virtual instalada incluye interfaz gráfica. Por lo tanto, puede accederse a un navegador (Mozilla Firefox). Abrir un navegador y navegar a <http://localhost:8080/>, igual a el ejemplo de contenedores. Aparece una imagen similar a la presentada en el otro ejemplo. Al aparecer la tabla completa, se evidencia el correcto inicio de los servicios: Apache, PHP, Mysql.

## 2.5. El demo

### Resumen

En Visual Studio Code se instalaron los siguientes complementos: Docker (containers) y Jupyter para el armado de un demo interactivo.

A continuación se presentan celdas con texto, imágenes y código interactivo que permite mostrar el objeto de esta prueba.

## 3. El experimento

### 3.1. Resumen

En este apartado se comparan tiempos operando con contenedores y sobre máquinas virtuales, en similares condiciones.

Se analizaron dos situaciones:

- **Disponibilidad del servidor desde power-off:** en caso de corte de energía general, se afecta al scada-server y a su redundancia, si está en la misma línea de alimentación, o si no se dispone de redundancia y se requiere un apagado forzado.
- **Acceso a la información con servidores encendidos y conectados:** esto es de utilidad para cuando se dispone de servidores con redundancia, conectados en un hot-standby.

### 3.2. Tiempo de Inicialización de servidores

Se inicializan servidores desde el estado “power-off”, hasta que se confirma la disponibilidad de la base de datos.

#### Servidores contenedorizados

##### Salida del programa:

Esperando a que el sistema esté disponible...

Sistema iniciado correctamente.

Tiempo Total de inicio de contenedores: 4.80 segundos

## Servidores en Máquina virtuales

### Salida del programa:

Iniciando VM...

✓ VM iniciada.

🕒 Tiempo hasta inicio de VM: 1.10 s

🌐 Verificando Apache en puerto 8080

✓ Apache está disponible.

🕒 Tiempo hasta Apache activo: 37.30 s

🔧 Buscando clave MySQL en el sitio...

✓ Clave 'clave1234' encontrada. MySQL y DB están operativos.

🕒 Tiempo hasta DB MySQL operativa: 6.36 s

✓ Tiempo TOTAL de inicio completo: 44.76 s

## 3.3. Tiempos de Acceso a servidores virtualizados

Se miden los tiempos de acceso desde la consola hasta los diferentes servidores, encendidos.

### Servidores contenedorizados

#### Salida del programa:

Iniciando acceso a contenedores...

Sistema accesible correctamente.

Tiempo de acceso a contenedores: 0.05 segundos

### Servidores en Máquina virtuales

#### Salida del programa:

Iniciando acceso a máquina virtual...

Sistema accesible correctamente.

Tiempo de acceso a la máquina virtual: 0.03 segundos

## 4. Conclusiones

En este trabajo se realizó una comparativa de la disponibilidad de servidores usando tecnologías de virtualización por contenedorización en comparación con servidores instalados en máquinas virtuales.

Se analizaron dos situaciones: El primero se mide la demora para la disponibilidad del servidor desde power-off: en caso de corte de energía general, que afecte tanto al scada-server como a su redundancia, o si no se dispone de redundancia y se requiere un apagado forzado. El segundo mide el tiempo de acceso a la información con servidores encendidos y conectados en redundancia.

**Para el primer caso, resulta altamente favorable a la tecnología contenedorizada, en un orden de magnitud: 4,80 segundos contra 44,76 segundos.**

**En el segundo caso, en cambio, resulta favorable a las máquinas virtuales, pero ambos en el mismo orden de magnitud: 30 ms contra 50ms.**

*Nota: Si bien los valores absolutos aquí obtenidos se corresponden con la aplicación presentada en esta POC, se espera que en otras condiciones de aplicación se presenten resultados proporcionales.*

