**Actividad**

**Diferencias entre máquina virtual y contenedor a la hora de crear proyectos en local, y cómo pasarán a producción.**

Antes de mencionar sus diferencias es importante dejar en claro que son tanto una máquina virtual como un contenedor.

**Màquinas virtuales:**

Una máquina virtual es un entorno virtual que funciona como sistema informático virtual con su propia CPU, memoria, interfaz de red y almacenamiento, pero se crea en un sistema de hardware físico, ya sea en las instalaciones o no. El sistema de software se llama hipervisor, y se encarga de separar los recursos de la máquina del sistema de hardware e implementarlos adecuadamente para que la VM pueda utilizarlos.

stoE es de utilidad a la hora de desarrollar aplicaciones para varias plataformas, pues en vez de tener que programar específicamente para cada sistema, el entorno de ejecución se encarga de lidiar con el sistema operativo.

-Funciones:

\*Probar otros sistemas operativos.

\*Ejecutar programas antiguos

\*Usar aplicaciones disponibles para otros sistemas

\*Probar una aplicación en distintos sistemas.

\*Seguridad adicional

\*Aislamiento completo.

\*Compatibilidad con múltiples sistemas operativos.

**Contenedores**

Los contenedores son paquetes de software que incluyen todos los elementos necesarios para ejecutar tus productos en cualquier entorno. Como virtualizar el sistema operativo, se pueden ejecutar en cualquier parte, desde un centro de datos privado hasta la nube pública o incluso el portátil personal de un desarrollador.

Un contenedor es un paquete de código de software que contiene el código de una aplicación, sus bibliotecas y otras dependencias. El uso de contenedores permite que las aplicaciones sean portátiles, de modo que el mismo código se pueda ejecutar en cualquier dispositivo.

-Funciones:

\*Aislamiento ligero.

\*Eficiencia de recursos.

\*Despliegue rápido.

\*Portabilidad.

\*Gestión simplificada.

\*Menos seguridad en comparación con VMs.

**Diferencias entre ambos**

Las diferencias entre las máquinas virtuales y contenedores son las siguientes:

| Diferencias | Máquinas Virtuales | Contenedores |
| --- | --- | --- |
| Aislamiento | Proporciona aislamiento completo del sistema operativo host y otras VMs, ofreciendo un alto nivel de seguridad. | Proporciona aislamiento ligero del host y otros contenedores, con menos seguridad en comparación con las VMs. Puede aumentar la seguridad con el modo de aislamiento de Hyper-V. |
| Sistema Operativo | Ejecuta un sistema operativo completo, incluyendo el kernel, lo que requiere más recursos del sistema. | Ejecuta solo la parte del modo de usuario del sistema operativo y se personaliza para contener solo los servicios necesarios, lo que consume menos recursos. |
| Compatibilidad de Invitado | Puede ejecutar prácticamente cualquier sistema operativo dentro de la VM. | Debe ejecutarse en la misma versión del sistema operativo que el host, aunque el aislamiento de Hyper-V permite versiones anteriores del mismo sistema operativo en un entorno de máquina virtual ligera. |
| Implementación | Se implementan mediante herramientas como Windows Admin Center, Hyper-V Manager, PowerShell o System Center Virtual Machine Manager. | Se implementan individualmente mediante Docker y la línea de comandos, o en grupos con orquestadores como Azure Kubernetes Service (AKS). |
| Actualizaciones | Requieren la descarga e instalación de actualizaciones del sistema operativo en cada VM. Actualizar a una nueva versión del sistema operativo a menudo implica crear una nueva VM. | Actualizar archivos del sistema operativo en un contenedor implica editar el Dockerfile, reconstruir la imagen de contenedor, insertarla en el registro y volver a implementar con un orquestador. El proceso se automatiza eficazmente a escala. |
| Almacenamiento Persistente | Utiliza discos virtuales (VHD) para almacenamiento local o recursos compartidos de archivos SMB para almacenamiento compartido. | Usa discos de Azure para almacenamiento local en un solo nodo o Azure Files (recursos compartidos SMB) para almacenamiento compartido en varios nodos o servidores. |
| Equilibrio de Carga | El equilibrio de carga de VMs mueve las VMs en ejecución a otros servidores dentro de un clúster de conmutación por error. | Los contenedores en sí no se mueven, pero los orquestadores pueden iniciar o detener contenedores automáticamente en diferentes nodos para administrar la carga y la disponibilidad. |
| Tolerancia a Errores | Las VMs pueden conmutar por error a otro servidor de un clúster, reiniciando el sistema operativo en el nuevo servidor. | Si se produce un error en un nodo del clúster, el orquestador recrea rápidamente los contenedores en otro nodo del clúster. |
| Redes | Utiliza adaptadores de red virtual para la conectividad. | Utiliza una vista aislada de un adaptador de red virtual, compartiendo el firewall del host con los contenedores y usando menos recursos. |

**¿Cómo instalar docker?**

-Paso 1: Actualiza el sistema

*sudo apt update # Para sistemas basados en Debian/Ubuntu*

-Paso 2: Instala las dependencias

*sudo apt install -y apt-transport-https ca-certificates curl software-properties-common*

-Paso 3: Agrega el repositorio de Docker

*curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg*

*echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null*

-Paso 4: Instala Docker

*sudo apt update*

*sudo apt install -y docker-ce docker-ce-cli containerd.io*

Paso 5: Verifica la instalación de Docker

*docker --version*

Paso 6: Habilita y comienza el servicio Docker

*sudo systemctl enable docker*

*sudo systemctl start docker*

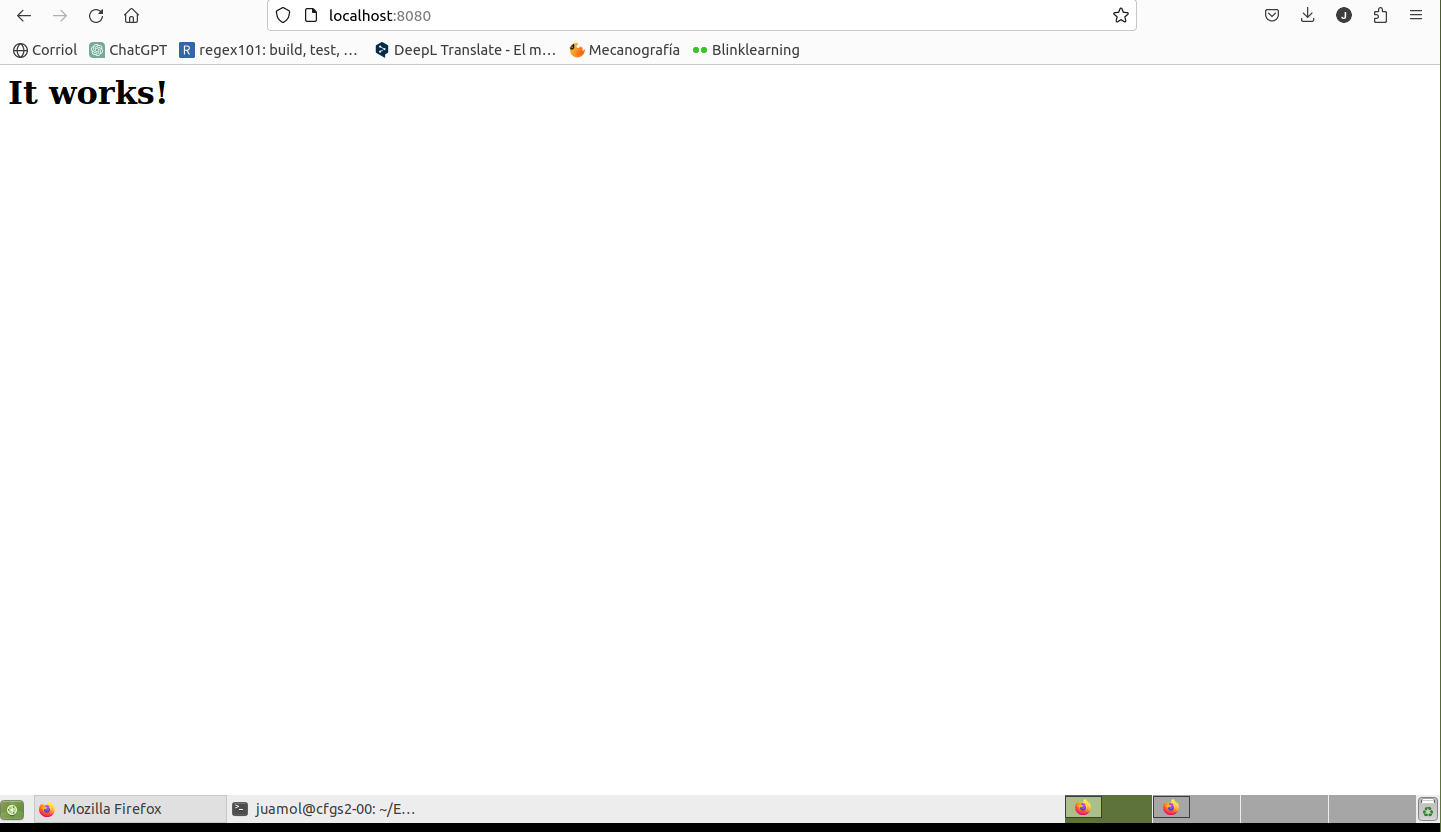
*sudo systemctl enable docker*

*sudo systemctl start docker*

Paso 7 (opcional): Agrega tu usuario al grupo "docker"

*sudo usermod -aG docker $USER*

**Utilizar docker para preparar un servidor web y un servidor de base de datos, en local, para manejar el proyecto en local.**

****

**Documento comparando las distintas opciones que existen para trabajar en local en el proyecto, indicando ventajas y desventajas.**

|  |  |  |
| --- | --- | --- |
| **Configuración Manual** | - Control total  - Flexibilidad  - No requiere Internet | - Configuración inicial laboriosa  - Posibles incompatibilidades  - Requiere recursos locales |
| **Máquina Virtual** | - Aislamiento  - Portabilidad  - Snapshots. | - Consumo de recursos adicionales  - Configuración inicial compleja  - Requiere software de virtualización |
| **Contenedores Docker** | - Aislamiento ligero  - Portabilidad  - Facilidad de configuración | - Menos aislamiento que las máquinas virtuales  - Requiere Docker instalado  - No adecuado para todos los casos |
| **Entornos de Desarrollo en la Nube** | - Acceso desde cualquier lugar  - Escalabilidad  - Gestión simplificada | - Costos  - Dependencia de la conexión a Internet  - Consideraciones de privacidad y seguridad |

**Documento indicando cómo se hace la instalación en Docker para poder realizar el proyecto en el que estamos trabajando (Debe incluir servidor web y servidor de base de datos)**

-Crear un directorio para tu proyecto

*mkdir mi\_proyecto*

*cd mi\_proyecto*

-Configurar un servidor web con Apache

*docker run -d -p 8080:80 --name mi-servidor-apache -v $(pwd)/mi\_sitio\_web:/var/www/html httpd:2.4*

-Configurar un servidor de base de datos con MySQL

*docker run -d -p 3306:3306 --name mi-servidor-mysql -e MYSQL\_ROOT\_PASSWORD=root -v $(pwd)/basededatos:/var/lib/mysql mysql:latest*

**Indicar cómo podemos utilizar el contenedor en Windows y Linux.**

Windows:

-Instalación de Docker:

Se puede instalar Docker Desktop o Docker Toolbox dependiendo de la version de windows

-Ejecución de Contenedores:

Abre Docker Desktop o la Docker Quickstart Terminal (si estás utilizando Docker Toolbox). Puedes usar comandos Docker en la línea de comandos de Windows o en la terminal de Docker para ejecutar contenedores, crear imágenes, etc.

Linux:

Ya fue explicado paso por paso anteriormente.