

# Comparación de IaaS vs Virtualización

Comparación: Docker, WSL y Amazon EC2.

## Índice

- [Comparación de IaaS vs Virtualización](#)
- [Índice](#)
  - [Docker](#)
  - [WSL \(Subsistema de Windows para Linux\)](#)
  - [Amazon EC2](#)
- [Cuál opción es mejor y por qué \(Pros y Contras\)](#)
- [Cómo instalar o configurar cada herramienta](#)
  - [Docker](#)
  - [WSL \(Subsistema de Windows para Linux\)](#)
  - [Amazon EC2](#)
- [Conclusión](#)

## Docker

### Resumen:

- **Tipo:** Virtualización a nivel de sistema operativo
- **Uso:** Ejecuta contenedores que encapsulan una aplicación y sus dependencias.

### Instalación y Configuración:

- **Facilidad de Instalación:** Proceso de instalación simple en la mayoría de los sistemas operativos.
- **Complejidad de Configuración:** Requiere la instalación de Docker Engine y familiaridad con los comandos de Docker.
- **Tiempo para Desplegar:** Rápido; iniciar un contenedor es casi instantáneo.

### Desarrollo y Pruebas:

- **Caso de Uso:** Ideal para entornos de desarrollo consistentes, pipelines de integración continua/despliegue continuo (CI/CD) y arquitectura de microservicios.
- **Mi Experiencia:** Usado como una fase de pruebas final para asegurar que la aplicación funcione correctamente en un entorno estable. Ayuda a identificar problemas del ecosistema antes del despliegue.

### Gestión de Recursos:

- **Eficiencia:** Ligero en comparación con las máquinas virtuales completas, compartiendo el núcleo del sistema operativo host.
- **Escalabilidad:** Fácilmente escalable a través de diferentes entornos.

## WSL (Subsistema de Windows para Linux)

### Resumen:

- **Tipo:** Capa de compatibilidad a nivel de sistema operativo
- **Uso:** Ejecuta distribuciones de Linux de forma nativa en Windows.

### Instalación y Configuración:

- **Facilidad de Instalación:** Fácil de habilitar a través de las características de Windows y descargando una distribución de Linux desde la Microsoft Store.
- **Complejidad de Configuración:** Mínima; configuración directa.
- **Tiempo para Desplegar:** Muy rápido, casi inmediato después de la configuración inicial.

### Desarrollo y Pruebas:

- **Caso de Uso:** Útil para desarrollar y probar aplicaciones de Linux directamente desde un entorno Windows.
- **Mi Experiencia:** Usado para probar rápidamente errores en el código, especialmente en Python y C++. Principalmente usado para probar errores de compilación y asegurar la compatibilidad multiplataforma.

### Gestión de Recursos:

- **Eficiencia:** Muy eficiente; usa menos recursos ya que comparte el núcleo del sistema operativo de Windows.
- **Escalabilidad:** No está diseñado para despliegues a gran escala, pero es excelente para desarrollo y pruebas individuales.

## Amazon EC2

### Resumen:

- **Tipo:** IaaS (Infraestructura como Servicio)
- **Uso:** Proporciona capacidad de computación escalable en la nube.

### Instalación y Configuración:

- **Facilidad de Instalación:** No requiere instalación en máquinas locales; la configuración se realiza a través de la Consola de Gestión de AWS.
- **Complejidad de Configuración:** Moderada; implica crear una cuenta de AWS, configurar grupos de seguridad y lanzar instancias.
- **Tiempo para Desplegar:** Puede tardar varios minutos en lanzar y configurar una instancia.

### Desarrollo y Pruebas:

- **Caso de Uso:** Adecuado para desplegar aplicaciones en un entorno de producción, manejar aplicaciones a gran escala y ejecutar máquinas virtuales en la nube.
- **Mi Experiencia:** Generalmente utilizado para despliegues escalables y confiables que requieren alta disponibilidad.

### Gestión de Recursos:

- **Eficiencia:** Mayor sobrecarga en comparación con los contenedores debido a la virtualización completa del sistema operativo.
- **Escalabilidad:** Altamente escalable; puede manejar aplicaciones grandes y complejas con facilidad.

## Cuál opción es mejor y por qué (Pros y Contras)

### WSL y Docker:

- **Pros:**
  - Rápidos y fáciles de configurar en una máquina local.
  - Eficientes y ligeros, con mínima sobrecarga.
  - Ideales para entornos de desarrollo y pruebas.
  - Tu experiencia personal muestra que son herramientas efectivas para identificar y resolver rápidamente errores en el código.
- **Contras:**
  - Escalabilidad limitada en comparación con soluciones en la nube.
  - No son adecuados para ejecutar aplicaciones de producción a gran escala.

### Amazon EC2:

- **Pros:**
  - Altamente escalable y confiable.
  - Adecuado para despliegues de producción a gran escala.
  - Proporciona infraestructura robusta y características de seguridad.
- **Contras:**
  - Más complejo de configurar y gestionar.
  - Mayor sobrecarga debido a la virtualización completa del sistema operativo.
  - Tarda más en desplegar instancias en comparación con iniciar contenedores Docker o usar WSL.

## Cómo instalar o configurar cada herramienta

### Docker

#### Pasos de Instalación:

1. **Descargar Docker Desktop:** Ve a la página de descarga de [Docker Desktop](#) y descarga el instalador para tu sistema operativo.
2. **Ejecutar el Instalador:** Sigue las instrucciones en pantalla para completar la instalación. Asegúrate de habilitar el backend de WSL 2 durante la instalación si se te solicita.
3. **Iniciar Docker Desktop:** Una vez instalado, lanza [Docker Desktop](#) desde el menú de inicio o la carpeta de aplicaciones. Docker comenzará a ejecutarse y verás el icono de Docker en tu bandeja del sistema.
4. **Verificar Instalación:** Abre una terminal y ejecuta el siguiente comando para verificar que Docker esté instalado correctamente:

```
docker --version
```

### WSL (Subsistema de Windows para Linux)

#### Pasos de Instalación:

1. **Habilitar WSL:** Abre PowerShell como administrador y ejecuta el siguiente comando para habilitar WSL:

```
wsl --install
```

2. **Listar Distribuciones Disponibles:** Para ver una lista de distribuciones de Linux disponibles, ejecuta:

```
wsl.exe -l -o
```

3. **Instalar una Distribución Específica:** Reemplaza `<Distribution Name>` con el nombre de la distribución que deseas instalar (por ejemplo, `Ubuntu-22.04`):

```
wsl.exe --install -d <Distribution Name>
```

4. **Configuración:** Una vez que la instalación esté completa, WSL te pedirá que crees una cuenta de usuario y contraseña para la distribución de Linux.

### Ejemplo:

Para instalar Ubuntu:

```
wsl.exe --install -d Ubuntu-22.04
```

Después de ejecutar el comando, WSL instalará Ubuntu y lo configurará para que lo uses.

## Amazon EC2

### Pasos de Configuración:

1. **Crear una Cuenta de AWS:** Si aún no tienes una cuenta de AWS, crea una en [aws.amazon.com](https://aws.amazon.com).
2. **Lanzar una Instancia EC2:**
  - Inicia sesión en la Consola de Gestión de AWS.
  - Navega al Panel de EC2.
  - Haz clic en "Launch Instance".
  - Selecciona una Imagen de Máquina de Amazon (AMI).
  - Elige un tipo de instancia (por ejemplo, t2.micro para el nivel gratuito).
  - Configura los detalles de la instancia, añade almacenamiento y añade etiquetas si es necesario.
  - Configura los grupos de seguridad (abre los puertos necesarios, como SSH).
  - Revisa y lanza la instancia.
  - Selecciona o crea un par de claves para el acceso SSH.

### 3. Conectarse a la Instancia:

- Una vez que la instancia esté en ejecución, selecciónala desde el Panel de EC2.
- Haz clic en "Connect" y sigue las instrucciones para conectarte vía SSH usando el par de claves.

## Conclusión

### Instalación de WSL:

```
wsl --install  
wsl.exe --install -d Ubuntu
```

### Instalación de Docker:

1. Descarga e instala Docker Desktop desde el sitio web de Docker.
2. Sigue las instrucciones de instalación y habilita el backend de WSL 2 si se te solicita.
3. Verifica la instalación:

```
docker --version
```

### Configuración de Amazon EC2:

1. Crea una cuenta de AWS.
2. Lanza una instancia de EC2 desde la Consola de Gestión de AWS.
3. Conéctate a la instancia usando SSH:

```
ssh -i /path/to/your-key-pair.pem ec2-user@your-ec2-instance-public-dns
```

### Conclusión:

- **Para Desarrollo y Pruebas:** WSL y Docker son mejores debido a su facilidad de uso, configuración rápida y eficiencia. Proporcionan un entorno flexible para desarrollar y probar aplicaciones, especialmente en un contexto multiplataforma.
- **Para Despliegues de Producción:** Amazon EC2 es mejor debido a su escalabilidad, fiabilidad e infraestructura robusta. Es adecuado para ejecutar aplicaciones a gran escala que requieren alta disponibilidad y rendimiento.