Sesión # 17 Componente Práctico

Manejo de contenedores

Conociendo Docker como una poderosa plataforma para desarrollar, lanzar y ejecutar aplicaciones, que permite separar las aplicaciones desarrolladas de la infraestructura donde se desarrollan y trabajar así de una manera más cómoda y rápida. Vamos a realizar un ejercicio de crear nuestra propia máquina y configuración de contenedor.

Instalación WSL2

1. Ten en cuenta que tu máquina Windows debe cumplir con los siguientes requisitos para instalar con éxito Docker Desktop.

    - Windows 11 de 64 bits: Home o Pro versión 21H2 o superior, o Enterprise o Education versión 21H2 o superior.
    - Windows 10 de 64 bits: Home o Pro 2004 (compilación 19041) o superior, o Enterprise o Education 1909 (compilación 18363) o superior.
    - Procesador de 64 bits con traducción de direcciones de segundo nivel (SLAT)
    - RAM del sistema de 4GB
    - El soporte de virtualización de hardware a nivel de BIOS debe estar habilitado en la configuración de BIOS. Para obtener más información, consulte Virtualización .

    Para versiones anteriores ver este enlace.

2. Descargue el paquete de actualización del kernel de Linux y ejecute el paquete descargado. (Haga doble clic para ejecutarlo. Se le pedirán permisos elevados. Seleccione "Sí" para aprobar esta instalación).

3. Ahora puede instalar todo lo que necesita para ejecutar el Subsistema de Windows para Linux (WSL), ingresando este comando en un PowerShell con permisos de administrador o en el Símbolo del sistema de Windows y luego reiniciando su máquina.

   ```
   wsl --install
   ```

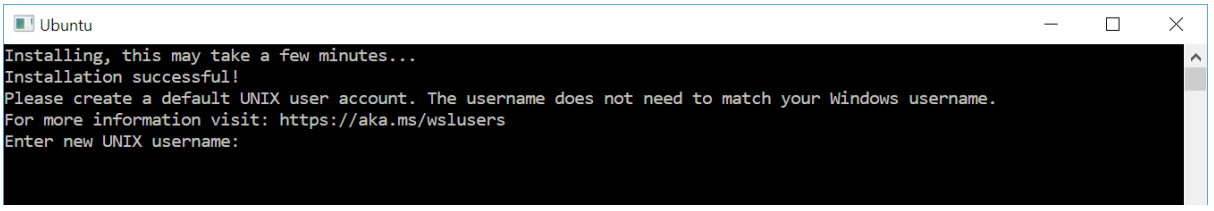
   Este comando habilita los componentes opcionales requeridos.

4. Abra PowerShell y ejecute este comando para establecer WSL 2 como versión predeterminada al instalar una nueva distribución de Linux:

   ```
   wsl --set-default-version 2
   ```

5. Ahora, acceda a la tienda de aplicaciones Windows y descargue Ubuntu. En la página de la distribución, selecciona "Obtener".

6. La primera vez que inicies una distribución de Linux recién instalada, se abrirá una ventana de la consola y se te pedirá que esperes uno o dos minutos para que los archivos se descompriman y almacenen en tu equipo. Todos los inicios posteriores deberían tardar menos de un segundo en completarse. Espera y crea un usuario y contraseña.



7. Revise las buenas prácticas para instalación.

8. ¡Listo! Ha instalado y configurado correctamente una distribución de Linux completamente integrada con el sistema operativo Windows.

Instalación Docker

1. Instalaremos Docker Desktop, el cual nos permitirá abrir el panel de Docker, ejecutar la Guía de inicio rápido, configurar los ajustes de Docker como la instalación, las actualizaciones, los canales de versión, el inicio de sesión de Docker Hub, entre otras cosas.

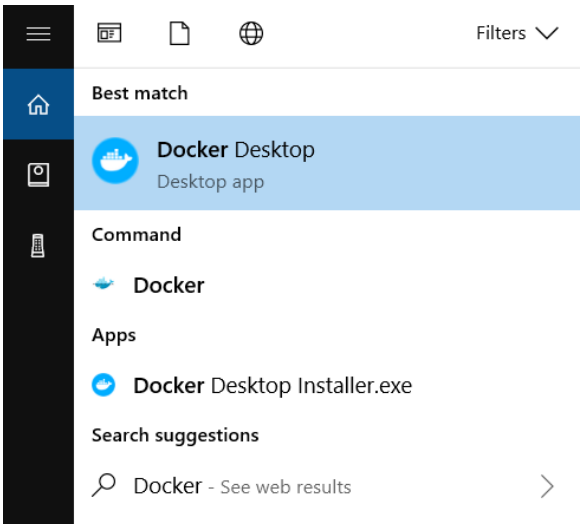
   Haga doble clic en Docker Desktop Installer.exe para ejecutar el instalador.

2. Cuando se le solicite, asegúrese de que la opción Habilitar las características de Hyper-V Windows o Instalar los componentes requeridos de Windows para WSL 2 esté seleccionada en la página Configuración.

3. Siga las instrucciones del asistente de instalación para autorizar al instalador y continuar con la instalación.

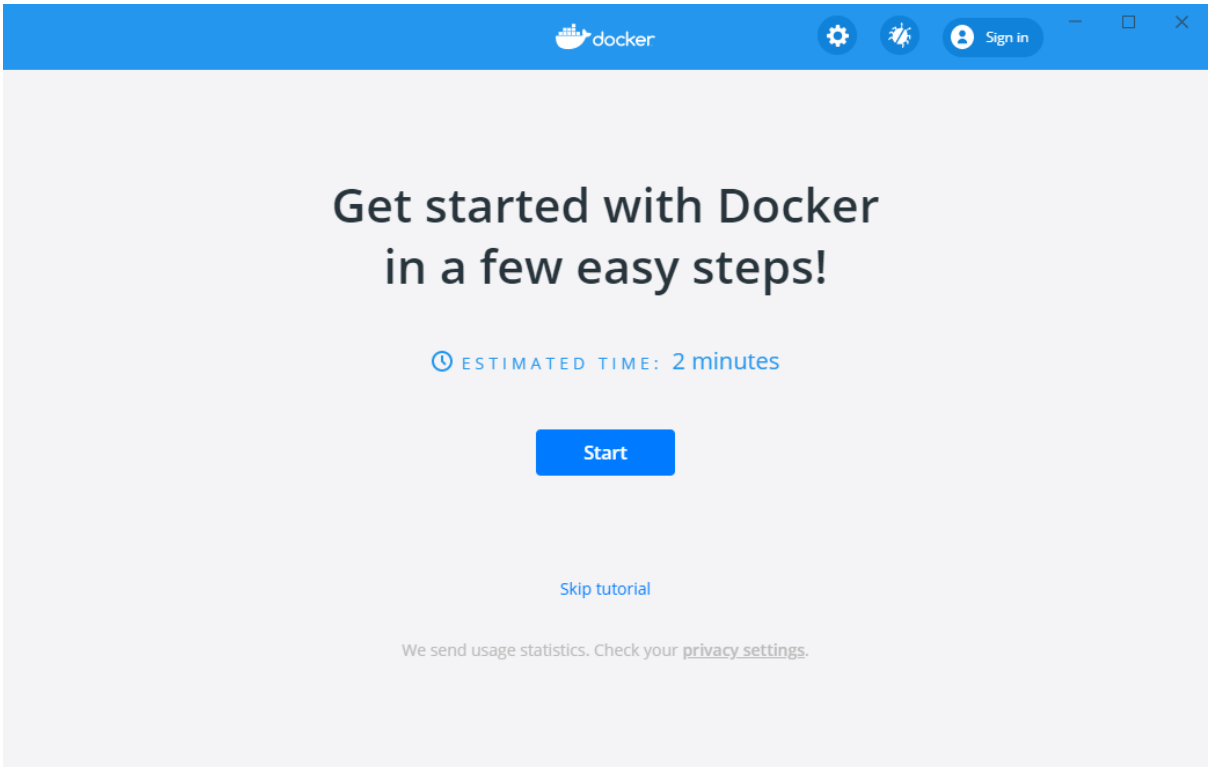
4. Cuando la instalación sea exitosa, haga clic en Cerrar para completar el proceso de instalación.

5. Busque Docker y seleccione Docker Desktop en los resultados de la búsqueda.



6. Haga clic en la casilla de verificación para indicar que acepta los términos actualizados y luego haga clic en Aceptar para continuar. Docker Desktop comienza después de que acepta los términos.

7. Cuando se completa la inicialización, Docker Desktop inicia la Guía de inicio rápido. Este tutorial incluye un ejercicio simple para crear una imagen de Docker de ejemplo, ejecutarla como un contenedor, empujar y guardar la imagen en Docker Hub.


docker
Sign in
Get started with Docker
in a few easy steps!
ESTIMATED TIME: 2 minutes
Start
Skip tutorial
We send usage statistics. Check your privacy settings.

8. Listo!

Creación de un contenedor

A continuación aprenderemos a crear e implementar aplicaciones de Docker en Windows o Mac usando Visual Studio Code, incluido el uso de varios contenedores con una base de datos y el uso de Docker Compose.

Los contenedores son entornos virtualizados compactos, como máquinas virtuales (VM), que proporcionan una plataforma para crear y ejecutar aplicaciones, pero sin el tamaño completo y la sobrecarga del sistema operativo completo. Docker Desktop se ejecuta en su máquina y administra sus contenedores locales. Las herramientas de desarrollo como Visual Studio y VS Code ofrecen extensiones que le permiten trabajar con un servicio Docker Desktop instalado localmente para crear aplicaciones en contenedores, implementar aplicaciones en contenedores y depurar aplicaciones que se ejecutan en sus contenedores.

1. Abra un símbolo del sistema o una ventana bash y ejecute el comando:
   `docker run -d -p 80:80 docker/getting-started`

   Notará que se utilizan algunas banderas. Aquí hay más información sobre ellos:

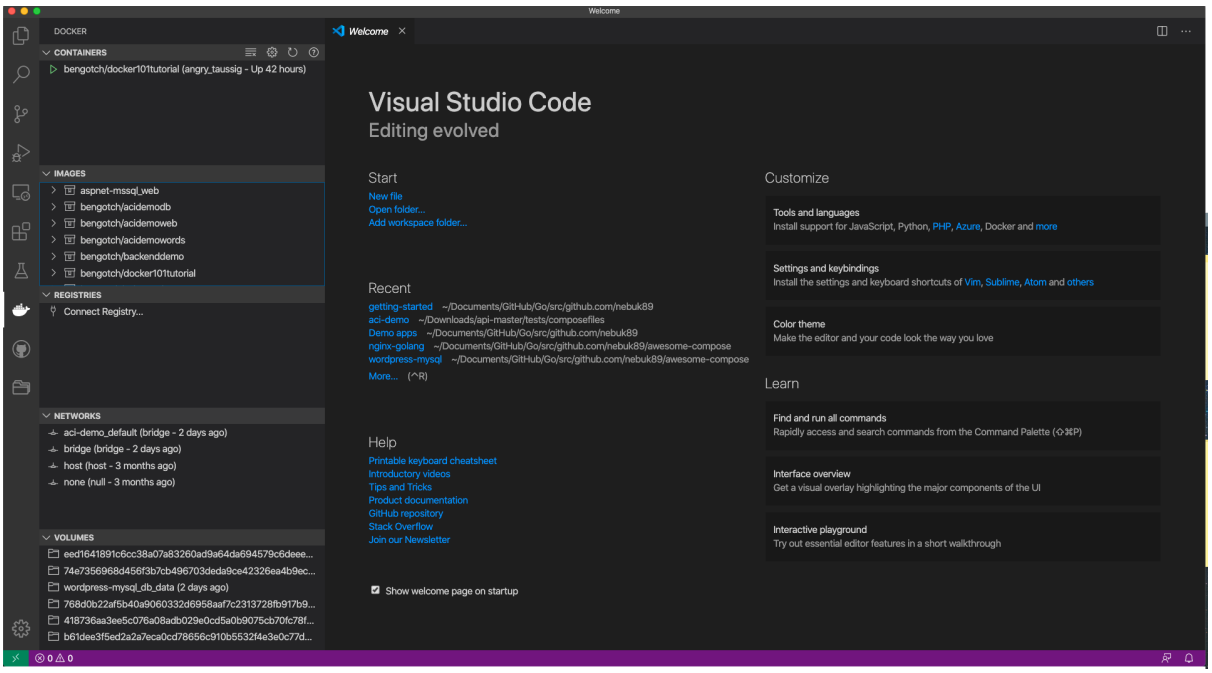
   -d - ejecutar el contenedor en modo separado (en segundo plano)

   -p 80:80 - mapear el puerto 80 del host al puerto 80 en el contenedor

   docker/getting-started - la imagen a utilizar

2. Abra Visual Studio Code e instale la extensión Docker

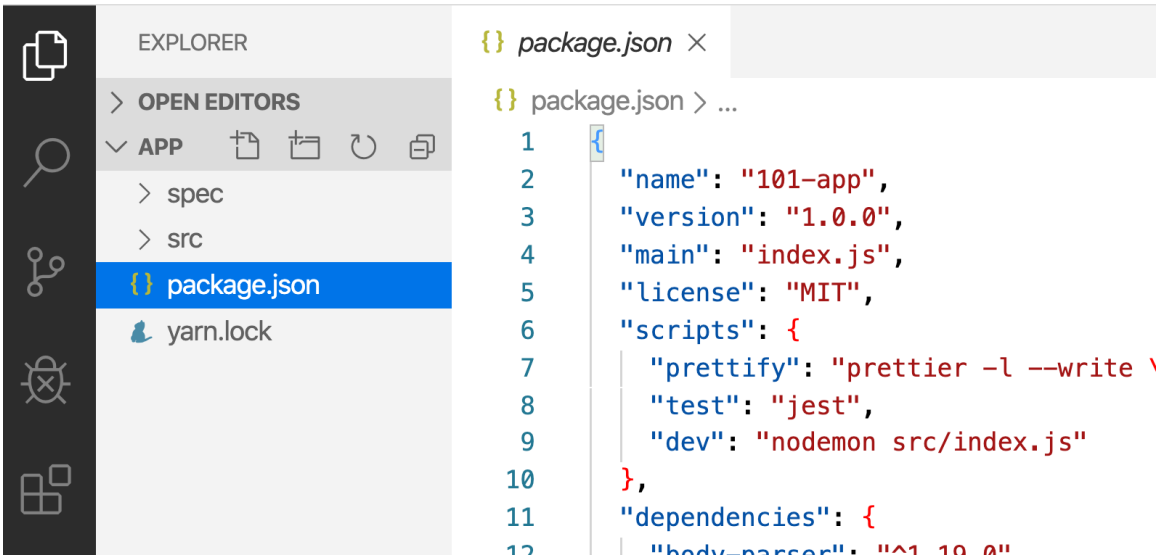
3. Reinicie VSCode y utilice el icono de Docker a la izquierda para abrir la vista de Docker. Si abre la extensión ahora, verá este tutorial en ejecución. El nombre del contenedor (angry_taussigabajo) es un nombre creado al azar. Por lo tanto, lo más probable es que tenga un nombre diferente.



4. Ahora, trabajaremos como proyecto con un administrador de listas de tareas pendientes que se ejecuta en Node.js.

5. Descargue la fuente de la aplicación o clone desde el repositorio de Docker .

6. Abra el proyecto y deberá poder ver los archivos package.js y dos subdirectorios (src y spec).



7. Para construir la aplicación, se necesita usar un Dockerfile. Un Dockerfile es simplemente un script de instrucciones basado en texto que se usa para crear una imagen de contenedor. Para ello, cree un archivo con el nombre Dockerfile en la misma carpeta que el archivo package.json con el siguiente contenido.

```
FROM node:12-alpine

WORKDIR /app

COPY . .

RUN yarn install --production

CMD ["node", "/app/src/index.js"]
```

8. Si aún no lo ha hecho, abra una terminal y vaya al app directorio con el Dockerfile. Ahora construya la imagen del contenedor usando el comando docker build.

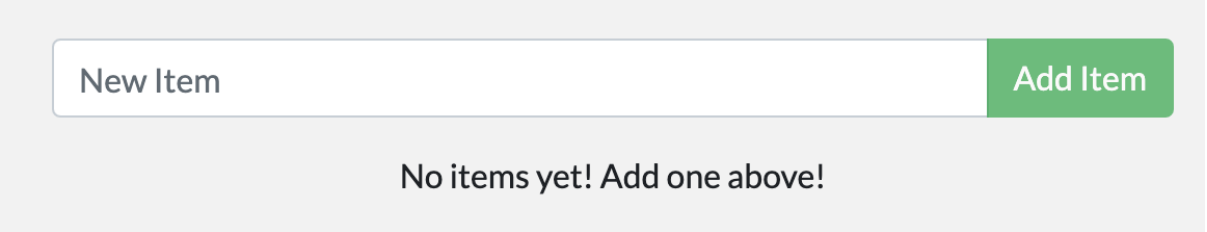
   docker build -t getting-started .

9. Ahora, sí revisas el Docker Desktop verás que este comando usó el Dockerfile para construir una nueva imagen de contenedor. Es posible que haya notado que se descargaron muchas "capas". Esto se debe a que le indicó al constructor que deseaba comenzar la imagen. **node:12-alpine.** Pero, como no tenía eso en su máquina, esa imagen necesitaba ser descargada.

10. Ahora que tienes una imagen, ¡ejecuta la aplicación! Para hacerlo, Inicie su contenedor usando el comando docker run y especifique el nombre de la imagen que acaba de crear:

    docker run -dp 3000:3000 getting-started

11. Después de unos segundos, abra su navegador web en la dirección http://localhost:3000. ¡Deberías ver la aplicación!



12. Continúe y agregue uno o dos elementos y compruebe que funcionan como se espera. Puede marcar elementos como completos y eliminar elementos. ¡Tu interfaz está almacenando elementos con éxito en el backend! Bastante rápido y fácil.

Referencias

- https://docs.microsoft.com/en-us/visualstudio/docker/tutorials/your-application
- https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-docker
- https://docs.microsoft.com/en-us/visualstudio/docker/tutorials/docker-tutorial
- https://docs.docker.com/desktop/windows/

- https://docs.microsoft.com/en-us/windows/wsl/install-manual
- https://docs.microsoft.com/es-es/windows/wsl/install