

Introducción a los Contenedores, Docker y Registro de Contenedores en IBM Cloud

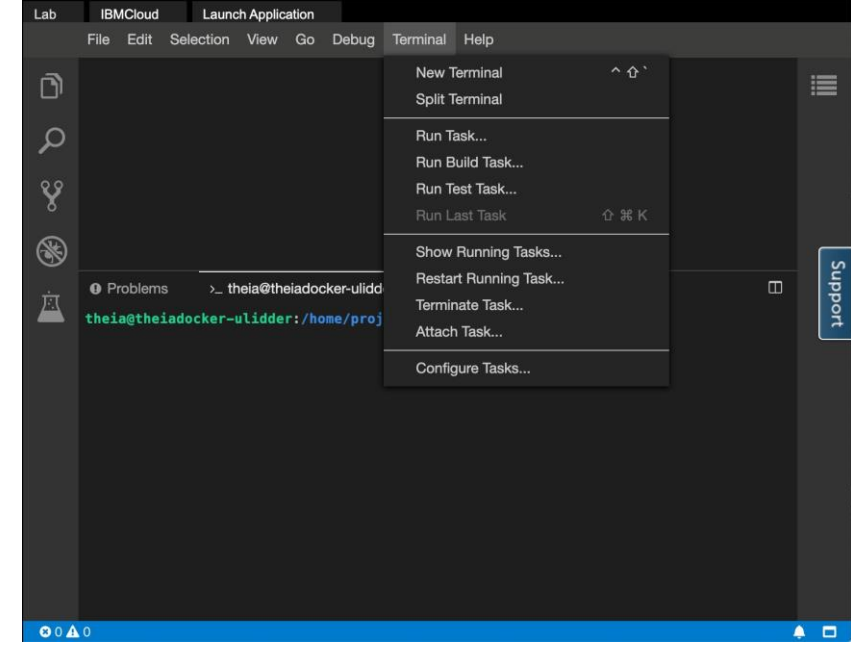
Objetivos

En este laboratorio, usted:

- Sacará una imagen del Docker Hub
- Ejecutará una imagen como un contenedor usando docker
- Construirá una imagen usando un Dockerfile
- Empujará una imagen en el registro de contenedores de IBM Cloud

Verificar el entorno y las herramientas de la línea de mando

1. Abra una ventana de terminal usando el menú del editor: Terminal > Nueva Terminal.



2. Verifique que el CLI Docker esté instalado.

```
docker --version
```

[:codeblock]

Debería ver una salida similar a esta, aunque la versión puede ser diferente:

```
docker version 18.09.7, build 2d0083d
```

3. Verifique que el CLI de ibmcloud esté instalado.

```
ibmcloud version
```

[:codeblock]

Debería ver una salida similar a esta, aunque la versión puede ser diferente:

```
ibmcloud version 1.0.0-908f90a-2020-03-30T06:37:22+00:00
```

3. Cambie a su carpeta de proyecto.

```
cd /home/project
```

[:codeblock]

4. Clonar el depósito de git que contiene los artefactos necesarios para este laboratorio.

```
git clone https://gitlab.com/ibm/sk111s-network/courses/cc201.git
```

[:codeblock]

5. Cambie al directorio de este laboratorio.

```
cd cc201/labs/_ContainersAndDocker/
```

[:codeblock]

6. Enumere el contenido de este directorio para ver los artefactos de este laboratorio.

```
ls
```

[:codeblock]

Sacar una imagen del Docker Hub y ejecutarla como un contenedor

1. Use el docker CLI para enlistar sus imágenes.

```
docker images
```

[:codeblock]

Debería ver una mesa vacía ya que no tiene ninguna imagen todavía.

2. Saque su primera imagen del Docker Hub.

```
docker pull hello-world
```

[:codeblock]

3. Enlistar las imágenes de nuevo para ver esta imagen en el entorno local.

```
docker images
```

[:codeblock]

Ahora debería ver la imagen de hola-mundo presente en la mesa.

4. Ejecute la imagen del hola-mundo como un contenedor.

```
docker run hello-world
```

[:codeblock]

Este comando imprime un simple mensaje de hola y alguna explicación de lo que Docker hizo para generar este mensaje.

5. Enumere los contenedores para ver que su contenedor se ejecutó y salió con éxito.

```
docker ps -a
```

[:codeblock]

Entre otras cosas, para este contenedor debería ver una identificación del contenedor, el nombre de la imagen (hola mundo), y un estado que indica que el contenedor salió con éxito.

6. Anote la identificación del contenedor del paso anterior, y retire el contenedor ahora que lo hemos ejecutado.

```
docker container rm <container_id>
```

[:codeblock]

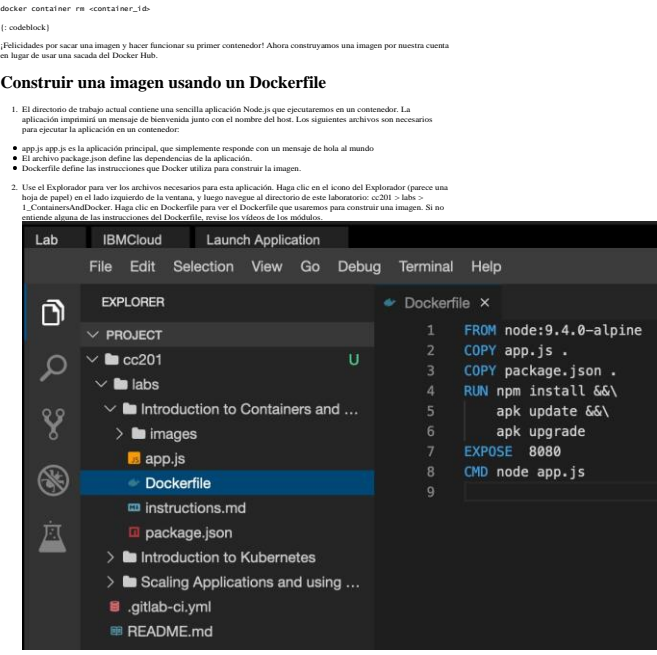
¡Felicidades por sacar una imagen y hacer funcionar su primer contenedor! Ahora construyamos una imagen por nuestra cuenta en lugar de usar una sacada del Docker Hub.

Construir una imagen usando un Dockerfile

1. El directorio de trabajo actual contiene una sencilla aplicación Node.js que ejecutaremos en un contenedor. La aplicación imprimirá un mensaje de bienvenida junto con el nombre del host. Los siguientes archivos son necesarios para ejecutar la aplicación en un contenedor:

- app.js app.js es la aplicación principal, que simplemente responde con un mensaje de hola al mundo
- El archivo package.json define las dependencias de la aplicación.
- Dockerfile define las instrucciones que Docker utiliza para construir la imagen.

2. Use el Explorador para ver los archivos necesarios para esta aplicación. Haga clic en el icono del Explorador (parece una hoja de papel) en el lado izquierdo de la ventana, y luego navegue al directorio de este laboratorio: cc201 > labs > 1_ContainersAndDocker. Haga clic en Dockerfile para ver el Dockerfile que usaremos para construir una imagen. Si no entiende alguna de las instrucciones del Dockerfile, revise los videos de los módulos.



3. Ejecute el siguiente comando para construir la imagen:

```
docker build . -t myimage:v1
```

[:codeblock]

Como se ve en los videos del módulo, la salida debe incluir un paso para cada instrucción en el Dockerfile. Cada paso crea una nueva capa en la imagen.

4. Listar las imágenes para ver su imagen etiquetada como myimage:v1 en la tabla.

```
docker images
```

[:codeblock]

Note que comparada con la imagen del hola-mundo, esta imagen tiene un ID de imagen diferente. Esto significa que las dos imágenes consisten en capas diferentes, en otras palabras, no son la misma imagen.

También debería ver una imagen de nodo en la salida de las imágenes. Esto se debe a que el comando de construcción del acoplador tiró del nodo:9.4.0-alpine para usarla como la imagen base de la imagen que construyó.

Ejecute la imagen como un contenedor

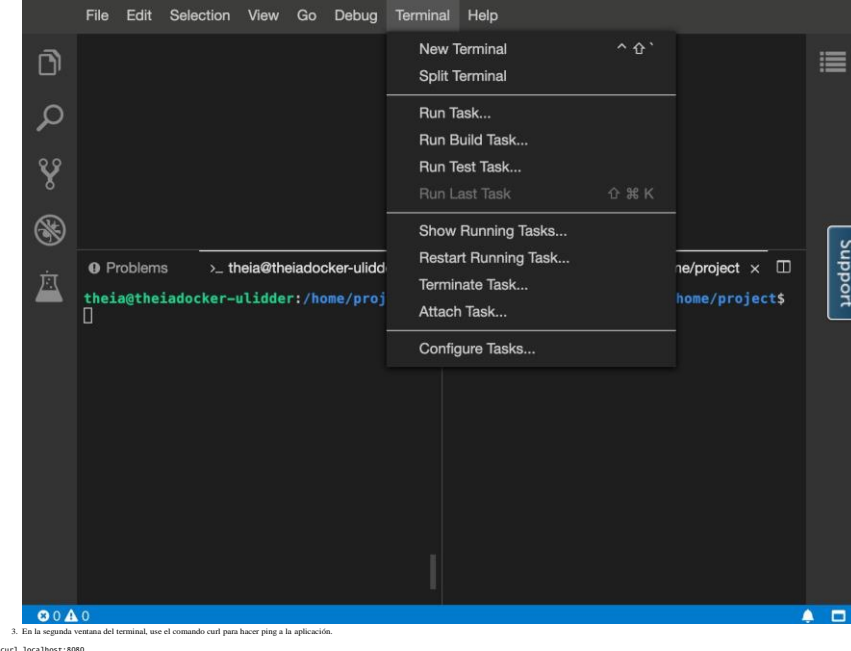
1. Ahora que tu imagen está construida, ejecútala como un contenedor con el siguiente comando:

```
docker run -p 8080:8080 myimage:v1
```

[:codeblock]

La salida debería indicar que su aplicación está escuchando en el puerto 8080. Este comando continuará ejecutándose hasta que se cierre, ya que el contenedor ejecuta una aplicación web que escucha continuamente las peticiones. Para consultar la aplicación, necesitamos abrir otra ventana de terminal.

2. Para dividir la terminal, haga clic en Terminal > Split Terminal.



3. En la segunda ventana de terminal, use el comando curl para hacer ping a la aplicación.

```
curl localhost:8080
```

[:codeblock]

La salida debería indicar que su aplicación está en funcionamiento.

4. En la segunda ventana de la terminal, detenga el contenedor. El siguiente comando usa el docker ps -q para pasar la lista de todos los contenedores en ejecución:

```
docker stop $(docker ps -q)
```

[:codeblock]

5. Cierre la segunda ventana terminal, ya que no es necesaria.

```
exit
```

[:codeblock]

En la ventana de la terminal original, el comando de ejecución de la plataforma ha salido y puedes volver a escribir comandos en esa ventana de la terminal.

Empuje la imagen al Registro de Contenedores de IBM Cloud

1. El medio ambiente ya debería haberle registrado en una cuenta de IBM Cloud. El siguiente comando le dará información sobre la cuenta a la que se dirige:

```
ibmcloud target
```

[:codeblock]

2. El entorno también creó un espacio de nombres de registro de contenedores en la IBM Cloud para usted. Dado que el Registro de Contenedores es multi-inquilino, los espacios de nombres se utilizan para repartir el registro entre varios usuarios. Utilice el siguiente comando para ver su espacio de nombres:

```
ibmcloud cr namespaces
```

[:codeblock]

3. Sin embargo, ya que creó su propio espacio de nombres de registro de contenedores de IBM Cloud en el laboratorio anterior, usaremos ese para este laboratorio. Para ello, inicie sesión en su cuenta de IBM Cloud. Si tiene un ID federado, use la opción --sso. Utilice la URL proporcionada en la salida de su CLI para recuperar su código de acceso de una sola vez. Se sabe que tiene una ID federada cuando el inicio de sesión falla sin la opción --sso y tiene éxito con la opción --sso.

```
ibmcloud login [--sso]
```

[:codeblock]

4. Asegúrese de que está apuntando a la región del sur de EE.UU. donde creó su espacio de nombres.

```
ibmcloud cr region-set us-south
```

[:codeblock]

5. Registre su daemon local de Docker en el registro de contenedores de IBM Cloud para que pueda empujar y tirar del registro.

```
ibmcloud cr login
```

[:codeblock]

6. Exporte su espacio de nombres como una variable de entorno para que pueda ser utilizado en comandos posteriores. Asegúrese de sustituir su espacio de nombres después del signo de igualdad. Si no recuerda su namespace, ejecute ibmcloud cr namespaces. Dado que ahora está conectado a su cuenta personal, debería ver su namespace personal en lugar del de la cuenta de laboratorio.

```
export MY_NAMESPACE=my_namespace
```

[:codeblock]

7. Etiquete su imagen para que pueda ser empujada al Registro de Contenedores de IBM Cloud.

```
docker tag myimage:v1 us.icr.io/$MY_NAMESPACE/hello-world:1
```

[:codeblock]

8. Empuje la imagen recién etiquetada al Registro de Contenedores de IBM Cloud.

```
docker push us.icr.io/$MY_NAMESPACE/hello-world:1
```

[:codeblock]

9. Verifique que la imagen fue empujada con éxito listando las imágenes en el Registro de Contenedores.

```
ibmcloud cr images
```

[:codeblock]

Debería ver el nombre de su imagen en la salida. Recordemos los videos del módulo que discutimos sobre el Asesor de Vulnerabilidad, que escanea imágenes en el Registro de Contenedores de IBM Cloud para detectar vulnerabilidades y exposiciones comunes. En la última columna de la salida, observe que Vulnerability Advisor está escaneando su imagen o ha proporcionado un estado de seguridad, dependiendo de la rapidez con la que se listan las imágenes y el tiempo que toma el escaneo.

10. Dado que en futuros laboratorios utilizará el espacio de nombres en la cuenta del laboratorio, vuelva a entrar en la cuenta del laboratorio.

```
ibmcloud login --apikey $IBM_CLOUD_API_KEY
```

[:codeblock]

¡Felicidades! Ha completado el segundo laboratorio para el primer módulo de este curso.