Crear una Imagen Personalizada

En este ejercicio, construiremos una imagen personalizada usando un Dockerfile que agrega un archivo test.txt a la imagen base alpine. Exploraremos los archivos generados en el sistema de Docker y veremos cómo se almacenan y estructuran las capas de la imagen. Además, realizaremos un push de la imagen a un repositorio y observaremos cómo cambian los IDs al hacer pull en otro entorno.

Pasos para el Ejercicio en Play with Docker

1. **Crear el Dockerfile:**
    - Abre un nuevo entorno en Play with Docker y crea un archivo Dockerfile con el siguiente contenido:

```
FROM alpine:latest

ADD test.txt /
```

    - Crea también un archivo test.txt en el mismo directorio, que será añadido a la imagen:

```
echo "Este es un archivo de prueba." > test.txt
```

2. **Construir la Imagen:**
    - Construye la imagen personalizada usando el siguiente comando:

```
docker build -t myimage .
```

    - Docker mostrará el progreso de la construcción, indicando los IMAGE IDs de cada paso. Anota estos IDs para facilitar el seguimiento.

3. **Verificar la Imagen Creada:**
    - Ejecuta docker images para ver las imágenes disponibles, incluyendo la imagen myimage que acabamos de construir:

```
docker images
```

    - Verás myimage junto con el IMAGE ID, la TAG, y el SIZE.

4. **Explorar los Archivos Generados:**
    - Ahora, navega al directorio donde Docker guarda la configuración de las imágenes:

```
cd /var/lib/docker/image/overlay2
```

    - Examina el archivo repositories.json, que muestra las asociaciones entre repositorios y IMAGE IDs:

```
cat repositories.json | jq
```

    - Verás que myimage no tiene un digest del manifest, ya que aún no se ha hecho push en un registro.

5. **Revisar el Archivo de Configuración de la Imagen:**

- Explora el archivo de configuración de la imagen myimage en imagedb/content/sha256/, donde 09c0177179d4... es el IMAGE ID generado durante la construcción:

```
cat /var/lib/docker/image/overlay2/imagedb/content/sha256/09c0177179d4... | jq
```

- Aquí, puedes ver la configuración de la imagen, que incluye la capa base de alpine y una capa adicional para test.txt.

6. **Verificación del Diff ID y Chain ID:**
   - Navega al directorio layerdb/sha256/ para ver cómo Docker organiza las capas utilizando el Chain ID. Los nombres de las carpetas en layerdb/sha256/ no son diff_ids (digest de la capa de diferencias), sino chainIDs, que representan el digest de cada capa compuesta junto con sus ancestros.
   - Calcula el Chain ID de la capa adicional (siguiendo la estructura de ancestros) y verifícalo en el sistema:

```
echo -n "sha256:<digest_de_capa_base> sha256:<digest_capa_adicional>" | sha256sum
```

   - Compara el Chain ID con los archivos en layerdb/sha256/.

7. **Push de la Imagen al Repositorio:**
   - Antes de realizar push, cambia el tag de la imagen para asociarla con un repositorio en Docker Hub:

```
docker tag myimage <tu_usuario>/myimage
```

   - Realiza push al repositorio:

```
docker push <tu_usuario>/myimage
```

   - Este proceso genera un digest de manifest en el repositorio, que no estaba disponible cuando la imagen estaba solo en local.

8. **Pull en Otro Entorno y Comparación de IDs:**
   - Si cambias de entorno (otro sistema o sesión en Docker), realiza un pull de la imagen:

```
docker pull <tu_usuario>/myimage
```

   - Observa que el IMAGE ID se mantiene, mientras que el digest refleja el manifest del repositorio. Además, el digest de las capas es distinto entre pull y push, ya que al hacer pull se descarga la imagen comprimida, mientras que al hacer push, Docker muestra el DiffID descomprimido.

Resumen

1. **IMAGE ID y Digest:** El IMAGE ID corresponde a la configuración de la imagen, mientras que el digest del manifest solo se genera en el repositorio.

2. **Differences in Digest for Compressed and Uncompressed Layers:** El digest de la capa comprimida y descomprimida difieren, afectando el DiffID en local y el digest en el repositorio.

Este ejercicio muestra cómo Docker organiza, almacena y estructura las capas y configuraciones de las imágenes.