
Actividad 3

Fundamentos de docker

Mikroways



Registries

1. Utilizando la **imagen docker oficial de una registry**, instanciar localmente una registry.
2. Pushear una imagen a la nueva registry:
 - Crear una imagen a partir de nginx que cambie el index.html
 - Pushear en la nueva imagen a la registry instanciada en el ejercicio 1.
3. Eliminar del docker engine local la imagen antes pusheada

```
1 docker rmi <IMAGEN>
```

Descargue desde la registry creada en el ejercicio 1, la imagen eliminada recientemente.

Volúmenes

1. Crear un contenedor con un **volumen anonimo**.
 - Crear un archivo dentro del directorio definido como un volumen.
 - Verificar el estado del contenedor y los volúmenes creados.
 - Realizar un inspect del volumen y obtener el directorio en donde se encuentra almacenado.
 - ¿Existe el archivo creado anteriormente en el directorio obtenido con el inspect?
 2. Repetir los pasos del ejercicio 1.
 - ¿Existen los archivos creados anteriormente? Justifique su respuesta.
- Utilizar **docker volume ls** y **docker volume inspect**
3. Instanciar nuevamente un contenedor con un volumen anónimo añadiendo la opción **--rm** y verificar el comportamiento de los contenedores y volúmenes al finalizar. ¿Cómo es el ciclo de vida del volumen en este caso?
 4. Crear un contenedor con un **volumen nombrado**.
 - Crear un archivo dentro del directorio definido como volumen.
 - Verificar el estado del contenedor y los volúmenes creados.
 - Realizar un inspect del volumen y obtener el directorio en donde se encuentra almacenado.
 - ¿Existe el archivo creado anteriormente en el directorio obtenido con el inspect?
 5. Repetir el ejercicio 4.
 - ¿Existen los archivos creados anteriormente? Justifique su respuesta.



Utilizar `docker volume ls` y `docker volume inspect`

6. Reutilizar el volumen creado en el ejercicio 4.

- Utilizar una imagen diferente y un punto de montaje diferente al utilizado en el punto 4.
- Al iniciar el contenedor verificar si existen los archivos creados anteriormente en el nuevo punto de montaje.

7. Crear un contenedor con un **volumen bind mount**.

- Crear un directorio en su máquina.
- Montar a este directorio como un volumen.
- Crear archivos en el directorio creado.
 - ¿Puede ver estos nuevos archivos dentro del contenedor?
- En el contenedor creado moverse al directorio donde se encuentra montado el volumen y agregar nuevos archivos.
 - ¿Es posible ver estos nuevos archivos en el directorio de su SO?

8. Crear un contenedor que utilice un volúmen anónimo.

- Crear un segundo contenedor que utilice el mismo volumen que el contenedor creado anteriormente.

Utilizar `--volumes-from`

9. Crear una instancia de postgres

- Descargue la imagen de postgres versión 12
- Utilice `docker inspect postgres:12` para observar que volúmenes exporta la imagen.
- Instancie un contenedor basado en la imagen descargada, utilice un volumen nombrado para persistir la base de datos.

Recomendamos dar un nombre al contenedor. Además setear la password y mapeo de puertos:

```
docker run --name=psql-docker -e POSTGRES_PASSWORD=psql-docker  
-p 5432:5432 ...
```

- Compruebe el volumen que creó.
- Conectar a la base de datos utilizando pgadmin4 como cliente:



```
1 docker run -p 80:80 \  
2   -e 'PGADMIN_DEFAULT_EMAIL=docker@mikroways.net' \  
3   -e 'PGADMIN_DEFAULT_PASSWORD=mikroways' \  
4   --link psql-docker:db -d dpage/pgadmin4
```

- Utilizando un navegador ingresar a localhost y acceder con el usuario: **docker@mikroways.net** y la contraseña **mikroways**.
- Conectar a una base de datos:
 - Server: **db**
 - User: **postgres**
 - Password: **psql-docker**
- Cree una base de datos “prueba”.
- Termine la ejecución del contenedor y bórralo.
- ¿El volumen nombrado continua existiendo?
- Instancie un nuevo contenedor con características similares al anterior
 - Utilice el mismo volumen nombrado.
 - ¿Continua existiendo la base de datos creada?
- Dentro de la imagen de postgres existe la utilidad psql.
 - Utilícela en combinación con docker exec para acceder a la base de datos.

10. Verifiqué el espacio utilizado por docker: `docker system df`

11. Libere todo el espacio utilizado por volúmenes docker.

- Pare todos los contenedores
- Elimine todos los volúmenes en desuso con **prune**

Entregables

Los ejercicios a entregar son sobre volúmenes y únicamente 2, 3 y 9. Las entregas deben responder a las preguntas de cada ejercicio o los comandos usados para completarlos.

