

Sistemas Operativos

Práctica 4 (II) - Docker

1. Utilizando sus palabras, describa qué es Docker y enumere al menos dos beneficios que encuentre para el concepto de contenedores.
2. ¿Qué es una imagen? ¿Y un contenedor? ¿Cuál es la principal diferencia entre ambos?
3. ¿Qué es Union Filesystem? ¿Cómo lo utiliza Docker?
4. ¿Qué rango de direcciones IP utilizan los contenedores cuando se crean? ¿De dónde la obtiene?
5. ¿De qué manera puede lograrse que los datos sean persistentes en Docker? ¿Qué dos maneras hay de hacerlo? ¿Cuáles son las diferencias entre ellas?

Taller:

El siguiente taller le guiará paso a paso para la construcción de una imagen Docker utilizando dos mecanismos distintos para los cuales deberá investigar y documentar qué comandos y argumentos utiliza para cada caso.

1. Instale Docker CE (Community Edition) en su sistema operativo. Ayuda: seguir las instrucciones de la página de Docker. La instalación más simple para distribuciones de GNU/Linux basadas en Debian es usando los repositorios.
2. Usando las herramientas (comandos) provistas por Docker realice las siguientes tareas:
 - a. Obtener una imagen de la última versión de Ubuntu disponible. ¿Cuál es el tamaño en disco de la imagen obtenida? ¿Ya puede ser considerada un contenedor? ¿Qué significa lo siguiente: *Using default tag: latest*?
 - b. De la imagen obtenida en el punto anterior iniciar un contenedor que simplemente ejecute el comando `ls -l`.
 - c. ¿Qué sucede si ejecuta el comando `docker [container] run ubuntu /bin/bash1`? ¿Puede utilizar la shell Bash del contenedor?
 - i. Modifique el comando utilizado para que el contenedor se inicie con una terminal interactiva y ejecutarlo. ¿Ahora puede utilizar la shell Bash del contenedor? ¿Por qué?
 - ii. ¿Cuál es el PID del proceso bash en el contenedor? ¿Y fuera de éste?
 - iii. Ejecutar el comando `lsns`. ¿Qué puede decir de los namespaces?
 - iv. Dentro del contenedor cree un archivo con nombre `sistemas-operativos` en el directorio raíz del filesystem y luego salga del contenedor (finalice la sesión de Bash utilizando las teclas `Ctrl + D` o el comando `exit`).
 - v. Corrobore si el archivo creado existe en el directorio raíz del sistema operativo anfitrión (host). ¿Existe? ¿Por qué?
 - d. Vuelva a iniciar el contenedor anterior utilizando el mismo comando (con una terminal interactiva). ¿Existe el archivo creado en el contenedor? ¿Por qué?
 - e. Obtenga el identificador del contenedor (`container_id`) donde se creó el archivo y utilícelo para iniciar con el comando `docker start -ia container_id` el contenedor en el cual se creó el archivo.

¹ Los corchetes indican que el argumento `container` es opcional, pero no son parte del comando a ejecutar.

- i. ¿Cómo obtuvo el `container_id` para para este comando?
 - ii. Chequee nuevamente si el archivo creado anteriormente existe. ¿Cuál es el resultado en este caso? ¿Puede encontrar el archivo creado?
 - f. ¿Cuántos contenedores están actualmente en ejecución? ¿En qué estado se encuentra cada uno de los que se han ejecutado hasta el momento?
 - g. Elimine todos los contenedores creados hasta el momento. Indique el o los comandos utilizados.
3. Creación de una imagen a partir de un contenedor. Siguiendo los pasos indicados a continuación genere una imagen de Docker a partir de un contenedor:
- a. Inicie un contenedor a partir de la imagen de Ubuntu descargada anteriormente ejecutando una consola interactiva de Bash.
 - b. Instale el servidor web Nginx, <https://nginx.org/en/>, en el contenedor utilizando los siguientes comandos²:

```
export DEBIAN_FRONTEND=noninteractive
export TZ=America/Buenos_Aires
apt update -qq
apt install -y --no-install-recommends nginx
```
 - c. Salga del contenedor y genere una imagen Docker a partir de éste. ¿Con qué nombre se genera si no se especifica uno?
 - d. Cambie el nombre de la imagen creada de manera que en la columna Repository aparezca `nginx-so` y en la columna Tag aparezca `v1`.
 - e. Ejecute un contenedor a partir de la imagen `nginx-so:v1` que corra el servidor web nginx atendiendo conexiones en el puerto 8080 del host, y sirviendo una página web para corroborar su correcto funcionamiento. Para esto:
 - I. En el Sistema Operativo anfitrión (host) sobre el cual se ejecuta Docker crear un directorio que se utilizará para este taller. Éste puede ser el directorio `nginx-so` dentro de su directorio personal o cualquier otro directorio - para los fines de este enunciado haremos referencia a éste como `/home/so/nginx-so`, por lo que en los lugares donde se mencione esta ruta usted deberá reemplazarla por la ruta absoluta al directorio que haya decidido crear en este paso.
 - II. Dentro de ese directorio, cree un archivo llamado `index.html` que contenga el código HTML de este gist de GitHub: <https://gist.github.com/ncuesta/5b959fce1c7d2ed4e5a06e84e5a7efc8>.
 - III. Cree un contenedor a partir de la imagen `nginx-so:v1` montando el directorio del host (`/home/so/nginx-so`) sobre el directorio `/var/www/html` del contenedor, mapeando el puerto 80 del contenedor al puerto 8080 del host, y ejecutando el servidor nginx en primer plano³. Indique el comando utilizado.
 - f. Verifique que el contenedor esté ejecutándose correctamente abriendo un navegador web y visitando la URL <http://localhost:8080>.
 - g. Modifique el archivo `index.html` agregándole un párrafo con su nombre y número de alumno. ¿Es necesario reiniciar el contenedor para ver los cambios?

² Los dos primeros comandos exportan dos variables de ambiente para que la instalación de una de las dependencias de nginx (el paquete `tzdata`) no requiera que interactivamente se respondan preguntas sobre la ubicación geográfica a utilizar

³ Para iniciar el servidor nginx en primer plano utilice el comando `nginx -g 'daemon off;'`

- h. Analice: ¿por qué es necesario que el proceso nginx se ejecute en primer plano? ¿Qué ocurre si lo ejecuta sin `-g 'daemon off;'`?
4. Creación de una imagen Docker a partir de un archivo Dockerfile. Siguiendo los pasos indicados a continuación, genere una nueva imagen a partir de los pasos descritos en un Dockerfile.
- En el directorio del host creado en el punto anterior (`/home/so/nginx-so`), cree un archivo Dockerfile que realice los siguientes pasos:
 - Comenzar en base a la imagen oficial de Ubuntu.
 - Exponer el puerto 80 del contenedor.
 - Instalar el servidor web nginx.
 - Copiar el archivo `index.html` del mismo directorio del host al directorio `/var/www/html` de la imagen.
 - Indicar el comando que se utilizará cuando se inicie un contenedor a partir de esta imagen para ejecutar el servidor nginx en primer plano: `nginx -g 'daemon off;'`. Use la forma `exec`⁴ para definir el comando, de manera que todas las señales que reciba el contenedor sean enviadas directamente al proceso de nginx.
Ayuda: las instrucciones necesarias para definir los pasos en el Dockerfile son FROM, EXPOSE, RUN, COPY y CMD.
 - Utilizando el Dockerfile que generó en el punto anterior construya una nueva imagen Docker guardándola localmente con el nombre `nginx-so:v2`.
 - Ejecute un contenedor a partir de la nueva imagen creada con las opciones adecuadas para que pueda acceder desde su navegador web a la página a través del puerto 8090 del host. Verifique que puede visualizar correctamente la página accediendo a `http://localhost:8090`.
 - Modifique el archivo `index.html` del host agregando un párrafo con la fecha actual y recargue la página en su navegador web. ¿Se ven reflejados los cambios que hizo en el archivo? ¿Por qué?
 - Termine el contenedor iniciado antes y cree uno nuevo utilizando el mismo comando. Recargue la página en su navegador web. ¿Se ven ahora reflejados los cambios realizados en el archivo HTML? ¿Por qué?
 - Vuelva a construir una imagen Docker a partir del Dockerfile creado anteriormente, pero esta vez dándole el nombre `nginx-so:v3`. Cree un contenedor a partir de ésta y acceda a la página en su navegador web. ¿Se ven reflejados los cambios realizados en el archivo HTML? ¿Por qué?

⁴ La documentación oficial de Docker describe las tres formas posibles para indicar el comando principal de una imagen: <https://docs.docker.com/engine/reference/builder/#cmd>.