TALLER DOCKER CON TOMCAT

Objetivo

Al final de este taller, deberán poder:

- Comprender los conceptos básicos de los contenedores Docker.
- Ejecutar contenedores con imágenes de Docker
- Hacer que el servidor Tomcat se ejecute en un contenedor
- Implementar la aplicación web en el servidor Tomcat
- Cree sus propias imágenes de Docker con Dockerfile

Código Fuente

Todo el código y la aplicación de muestra necesarios para este taller están disponibles en:

https://github.com/andresado/tomcat-taller

Clónalo en tu computadora local

$ git clone

$ ls

Imagenes

Las imágenes se crean con el comando build y producirán un contenedor cuando se inicien con run

$ docker search tomcat

Al comprobar la documentación de la página de Tomcat, vemos que Tomcat construido en la versión **alpina de Linux** es más pequeño e ideal para la experimentación.

Una vez que hayamos identificado la imagen a usar, podemos escribir un Dockerfile especificando la imagen base a usar, la aplicación web a usar, etc.

Puede crear imágenes leyendo las instrucciones de un Dockerfile. Dockerfile contiene todos los comandos que un usuario puede llamar en la línea de comandos para ensamblar una imagen. Al utilizar la compilación de Docker, los usuarios pueden crear una compilación automatizada que ejecute varias instrucciones de línea de comandos en sucesión.

Dentro de dicho Dockerfile debemos llevar las siguientes instrucciones

- La instrucción FROM que inicializa una nueva etapa de compilación y establece la imagen base para instrucciones posteriores.
- La instrucción ADD copia nuevos archivos, directorios o URL de archivos remotos desde <src> y los agrega al sistema de archivos de la imagen en la ruta <dest>.
- La instrucción EXPOSE informa a Docker que el contenedor escucha en los puertos de red especificados en tiempo de ejecución.
- La instrucción CMD especifica qué ejecutar cuando se ejecuta el contenedor (no la imagen). En nuestro caso, el servidor TomCat se inicia ejecutando el script de shell que inicia el contenedor web. Solo puede haber una instrucción CMD en un Dockerfile.

Build

En el siguiente comando, se usa el Dockerfile que creamos anteriormente y la imagen recién construida se etiqueta como mywebapp.

$ docker build -t mywebapp.

Debería salir un mensaje final con la siguiente información, “Mywebapp etiquetado con éxito: último”

Ahora puede verificar que la imagen de mywebapp esté creada y lista para usarse. Tenga en cuenta que en este punto, solo ha creado la imagen, no hay un contenedor en ejecución.

$ docker image ls

Para ejecutar realmente el contenedor usando la imagen, siga los pasos a continuación

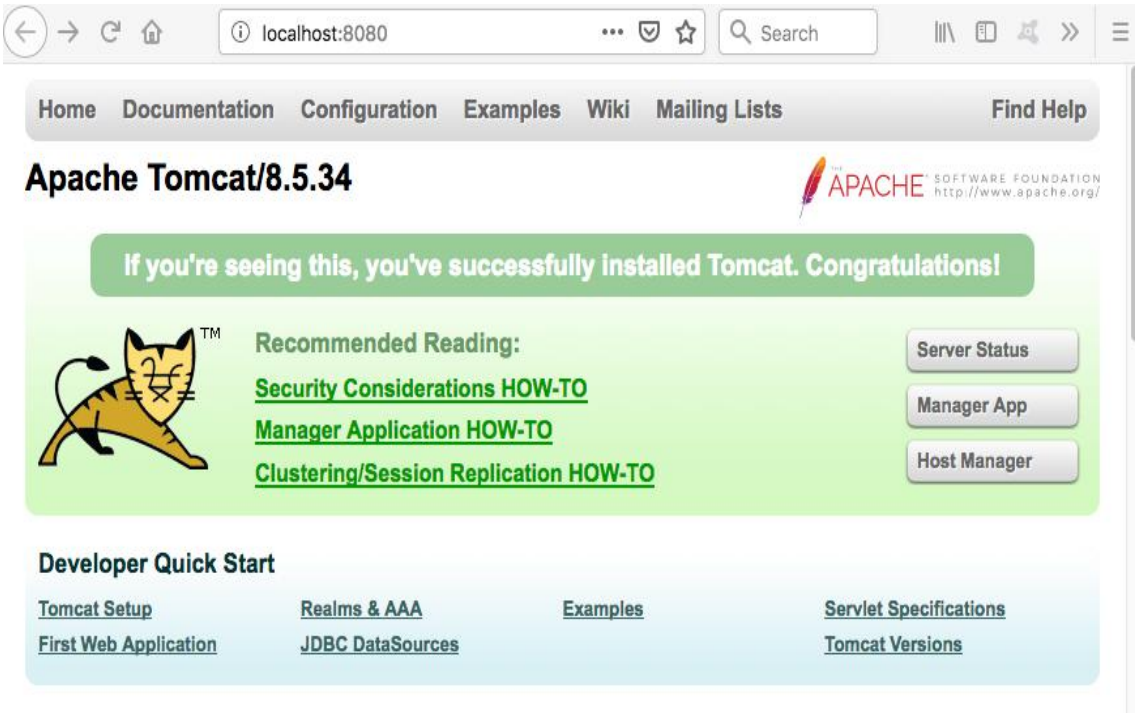
Ejecutar el Contenedor

La CLI tiene un comando llamado RUN que iniciará un contenedor basado en una imagen de Docker. La estructura es Docker run <options> <image-name>.

Como mencionamos antes, la instrucción EXPOSE en el Dockerfile en realidad no publica el puerto. Para que cuando ejecute el contenedor, use la marca -p en la ventana acoplable para publicar y mapear uno o más puertos. Entonces, para mapear el puerto contenedor 8080 para la imagen de mywebapp al puerto 80 en la máquina host, ejecutamos:

$ docker run -p 80:8080 mywebapp

Abrimos http://localhost:8080/ en un navegador web para ver la aplicación web de muestra en ejecución.



Comandos Adicionales Utiles de Docker

De forma predeterminada, ejecutará un comando en primer plano. Para ejecutarse en segundo plano, es necesario especificar la opción -d. Al hacerlo, lo cual se generará el ID del contenedor que podría usarse para otros comandos como se muestra a continuación.

$ docker ejecutar -d -p 80: 8080 mywebapp

El comando docker inspect <container-id> proporciona más detalles sobre un contenedor en ejecución, como la dirección IP.

El comando docker logs <container-id> mostrará los mensajes que el contenedor ha escrito en error estándar o salida estándar.

El comando docker ps enumera todos los contenedores en ejecución.