

Exámenes

Tema 4. Docker. Múltiples componentes. Test de autoevaluación

[Volver a la Lista de Exámenes](#)

Parte 1 de 2 -

4.0 Puntos

Preguntas 1 de 5

2.0 Puntos. Puntos descontados por fallo: 0.67

¿De qué forma podemos resolver las dependencias entre múltiples componentes de una aplicación distribuida desplegada mediante contenedores sin necesidad de regenerar las imágenes Docker a utilizar en ese despliegue?

- ☐ Procesando los Dockerfile de esos componentes con "docker-compose" en lugar de hacerlo con "docker build".
- ☐ Utilizando opciones avanzadas de "docker run" con las que es posible modificar la configuración establecida en el Dockerfile que generó la imagen correspondiente a cada uno de los componentes.
- ☒ Utilizando variables de entorno en los Dockerfile y estableciendo sus valores mediante opciones de "docker run" o en un "docker-compose.yml".
- ☐ Mediante "docker-compose", independientemente de que los Dockerfile utilizados para generar las imágenes de los componentes permitan utilizar variables de entorno o no.

Respuesta correcta: C

Preguntas 2 de 5

2.0 Puntos. Puntos descontados por fallo: 0.67

¿Con qué orden podemos averiguar la dirección IP utilizada por un contenedor?

- ☐ docker run
- ☐ docker images
- ☐ docker build
- ☒ docker inspect

Respuesta correcta: D

Parte 2 de 2 -

6.0 Puntos

Dado el siguiente contenido de un Dockerfile:

```
FROM zmq
RUN mkdir /zmq
COPY ./worker.js /zmq/worker.js
WORKDIR /zmq
CMD node worker $BROKER_PORT
```

Si se genera una imagen a partir de él mediante la orden:

```
docker build -t worker .
```

¿Cuál de las siguientes afirmaciones es FALSA?

- ☐ Si se crea un contenedor a partir de la imagen worker, se ejecutará por omisión el programa indicado en la instrucción CMD.
- ☐ La imagen worker es una ampliación de la imagen zmq.
- ☒ Si se crea un contenedor a partir de la imagen worker, no se ejecutará ningún programa, pues debe indicarse con la instrucción ENTRYPOINT en lugar de CMD.
- ☐ El directorio de trabajo para la instrucción CMD es /zmq.

Respuesta correcta: C

Considerando este Dockerfile...

```
FROM ubuntu
RUN apt-get install -y nodejs
RUN apt-get install -y npm
RUN apt-get install -y make
RUN npm install zeromq
```

...se puede afirmar que:

- ☒ Este Dockerfile crea una nueva imagen basada en la imagen ubuntu del Docker Hub. La nueva imagen añade al menos tres paquetes Ubuntu a esa imagen base.
- ☐ El nombre de la imagen creada con este Dockerfile es zeromq.
- ☐ Este Dockerfile es incorrecto porque falla en su segunda línea. No existe ninguna instrucción apt-get en Docker.
- ☐ Este Dockerfile no tiene sentido porque no incluye ninguna instrucción CMD o ENTRYPOINT. No hace nada en absoluto.

Respuesta correcta: A

Asumiendo este Dockerfile...

```
FROM zmq
RUN mkdir /zmq
COPY ./broker.js /zmq/broker.js
WORKDIR /zmq
EXPOSE 8000 8001
CMD node broker.js
```

¿Cuál de las siguientes afirmaciones es FALSA?

- ☐ Este Dockerfile asume la existencia de una imagen llamada "zmq" con una instalación válida del intérprete de JavaScript "node".
- ☐ Por omisión, los contenedores generados a partir de este Dockerfile ejecutarán la orden "node broker.js".
- ☒ El programa a ejecutar en estos contenedores utiliza el puerto 8000 del contenedor y lo asocia al puerto 8001 del anfitrión.
- ☐ Necesitamos tener el fichero "broker.js" en el directorio del anfitrión en el que se encuentre este Dockerfile.

Respuesta correcta: C

- [PoliformaT](#)
- [UPV](#)
- [Powered by Sakai](#)
- Copyright 2003-2021 The Sakai Foundation. All rights reserved. Portions of Sakai are copyrighted by other parties as described in the Acknowledgments screen.