

*Esta prueba tiene un valor de 1 punto, y consta de 10 cuestiones tipo test. Cada cuestión plantea 4 alternativas y tiene una única respuesta correcta. Cada respuesta correcta aporta 0.1 puntos, y cada error descuenta 0.033 puntos. Debe contestar en la hoja de respuestas.*

**1** Si deseamos conocer la dirección IP de un contenedor docker:

- a** Utilizaremos la orden `docker ps`
- b** Utilizaremos la orden `docker images`
- c** Utilizaremos la orden `docker inspect`
- d** No se puede lograr esa información, pues no es una propiedad de la imagen docker.

**2** La orden `docker-compose`:

- a** Toma como entrada únicamente un fichero `Dockerfile` donde se especifican las relaciones entre las imágenes docker.
- b** Permite ejecutar contenedores en máquinas remotas. Para ello se incluyen directivas para la conexión remota dentro del fichero de configuración.
- c** Permite ejecutar aplicaciones docker multi-contenedor.
- d** Todas las afirmaciones son ciertas.

**3** En la práctica 3 de laboratorio se ha construido una aplicación docker, donde una de las imágenes se encarga de registrar eventos en un fichero de log. A esta componente la hemos llamado `logger`. Señale la opción correcta.

- a** El fichero de log que se genera puede corromperse debido a condiciones de carrera en el `logger`.
- b** Mediante `docker-compose` hemos conectado el resto de componentes al `logger`, haciendo que cada componente tenga acceso al fichero de log, de forma que todas las componentes pueden escribir en el mismo fichero concurrentemente.
- c** El fichero de log será visible en la máquina `host` dado que así lo hemos indicado en la configuración.
- d** La componente `logger` utiliza un socket `rep` para recibir los mensajes de log.

**4** Si el componente `logger` se encontrara en un anfitrión diferente al del resto del sistema `cbw`, indíquese qué enunciado sería cierto:

- a** El archivo de despliegue `yaml` deberá dividirse en dos, de manera que la parte referente al `logger` formará un nuevo archivo (añadiendo el resto de instrucciones necesarias)
- b** Las herramientas empleadas en el laboratorio 3 no permitirían resolver este despliegue de forma automatizada
- c** El despliegue automatizado será posible si el anfitrión de `cbw_ftcl` permite conexiones desde otros equipos que encamina a los contenedores adecuados
- d** El despliegue automatizado será posible si el anfitrión del nuevo `logger` permite conexiones desde otros equipos que encamina al contenedor adecuado

- 5** **Se han admitido dos soluciones correctas** Al final del apartado 2 se sugiere reflexionar ante la posibilidad de desplegar `cbw_ftcl` en dos escenarios concretos. Selecciona la única afirmación cierta.
- a** Un despliegue con dos brokers no es posible porque ambos realizan bind en la misma URL
  - b** En un despliegue con dos brokers, únicamente uno ellos será capaz de interactuar con workers, clients y logger
  - c** En un despliegue con dos loggers, los mensajes del broker se repartirán alternativamente (round-robin) entre dichos loggers
  - d** En un despliegue con dos loggers, docker-compose encontrará dificultades para asignar valores a la variable `LOGGER_URL` porque depende de `log`
- 6** En la práctica 3, en cuanto al cliente externo, no se puede desplegar conjuntamente con `cbw_ftcl` porque...
- a** En los equipos de escritorio LINUX no hay Docker instalado
  - b** El despliegue con docker-compose no puede afectar a varios anfitriones
  - c** El cliente externo es incompatible con los clientes normales
  - d** El enunciado es falso: sí se puede desplegar conjuntamente
- 7** Imagina que acabas de ejecutar sin errores la orden `docker build -t provisional .`, indica qué afirmación es cierta:
- a** En el directorio actual, ahora existe un archivo llamado `provisional`
  - b** Al ejecutar ahora `docker inspect provisional` puedo consultar su IP (casi al final de la información devuelta) entre otras cosas
  - c** Aunque cambie de directorio, es posible ejecutar `docker run provisional`
  - d** El Dockerfile empleado seguro que incluye una orden `CMD` o `ENTRYPOINT`
- 8** En el sistema en el que consideramos un cliente externo:
- a** El acceso al contenedor broker se realiza a través de un puerto del anfitrión
  - b** Los clientes externos pueden interactuar con el logger
  - c** Para redirigir la petición del cliente al broker se realiza una entrada `ROUTE ip_localhost:ip_container` en el Dockerfile del componente broker
  - d** La IP del broker es fija y visible desde el nodo donde está ejecutándose el cliente externo
- 9** Sobre el sistema con un componente `worcli`, el código suministrado acepta como parámetros:
- a** Un url del broker
  - b** Dos url de dos brokers y el tipo de trabajo a procesar
  - c** Dos url de dos brokers y un retardo
  - d** Dos url de dos brokers, un retardo y el tipo de trabajo a procesar
- 10** **ANULADA** Considerando el escenario del cliente externo (`4_CBW_FTCL_CEXT`) y el `docker-compose.yml` suministrado
- a** El `docker-compose.yml` suministrado permite el acceso de un cliente externo al broker
  - b** Eliminando la sección `expose:` del `docker-compose.yml` es posible conectar al broker desde un cliente externo
  - c** Es necesario añadir una sección `ports:` mapeando el puerto de clientes desde el anfitrión al contenedor (`9999:9999`)
  - d** Es necesario eliminar la sección `expose:`