TSR	Segundo Parcial	Temas 3,4,5,6 Práctica 3	2023-01-23	D

Este examen tiene un valor de 10 puntos, y consta de 24 cuestiones de tipo test. Cada cuestión plantea 4 alternativas y tiene una única respuesta correcta. Cada respuesta correcta aporta 10/24 puntos, y cada error descuenta 10/72 puntos. Debe contestar en la hoja de respuestas.

1 *El patrón de comunicaciones DEALER-ROUTER presupone que el socket DEALER será utilizado por un proceso cliente y el socket ROUTER por un servidor. Si esos dos procesos todavía no han intercambiado ningún mensaje, ¿puede el servidor enviar el primer mensaje al cliente utilizando un socket ROUTER para ello?*

a Normalmente sí, aunque para ello el cliente habrá tenido que conectar previamente su socket DEALER a ese socket ROUTER.

b Sí, pues el socket ROUTER es bidireccional y asincrónico.

c Normalmente no, pues desconocerá la identidad a utilizar para comunicarse con el DEALER del cliente.

d No, pues el socket ROUTER únicamente podrá recibir mensajes.

2 *Entre otras tareas, el despliegue de un servicio distribuido comprende:*

a Todas las demás opciones son correctas.

b La escritura o desarrollo de los programas que compongan la aplicación distribuida cuya ejecución generará ese servicio.

c La parada de cada uno de los componentes del servicio, en la secuencia adecuada para evitar inconsistencias durante esa desactivación.

d La depuración de los programas que compongan la aplicación distribuida cuya ejecución generará ese servicio.

3 *Suponga que un proceso P1 está utilizando un socket DEALER llamado* `so1`*, que ya ha sido conectado correctamente a un socket* `so2` *de tipo ROUTER utilizado por un proceso P2. P1 utiliza esta instrucción para enviar una cadena a P2: so1.send('hola').*
¿Qué código necesitará P2 en el método on() de su socket `so2` *para escribir en pantalla el contenido ('hola') de ese mensaje?*

a so2.on('data', (m) => console.log(m+"))

b so2.on('message', (m) => console.log(m+"))

c so2.on('message', (i, j) => console.log(j+"))

d so2.on('message', (i, s, x) => console.log(x+"))

4 *En cierta actividad del Tema 3 se solicitaba desarrollar un servicio de chat con dos componentes: cliente y servidor. El cliente utilizaba un socket PUSH y un socket SUB. Por su parte, el servidor utilizaba un socket PULL y un PUB.*
¿Se podría desarrollar una implementación equivalente de este servicio utilizando un único socket ROUTER en el servidor y un único socket DEALER en cada cliente?

a No, pues el socket DEALER no puede emular simultáneamente el comportamiento de un socket PUSH y un socket SUB.

b No, pues los clientes estarían forzados a utilizar bind() en su DEALER y el servidor no podría conectar con todos ellos al desconocer sus direcciones.

c No, pues el servidor desconoce qué identidad utilizará el DEALER de cada cliente.

d Sí, pues ambos sockets son bidireccionalesy el servidor sabe cuántos clientes se han registrado en el servicio, así como la identidad utilizada en cada uno de sus DEALERs.

Considere estos ficheros Dockerfile en las próximas cuestiones relacionadas con Docker.

```
# Dockerfile A
FROM tsr2223/ubuntu-zmq
COPY ./mytsr.js tsr.js
RUN mkdir broker
WORKDIR broker
COPY ./broker.js mybroker.js
EXPOSE 9998 9999
CMD node mybroker 9998 9999
```

```
# Dockerfile B
FROM tsr2223/ubuntu-zmq
COPY ./tsr.js tsr.js
RUN mkdir worker
WORKDIR worker
COPY ./workerReq.js myworker.js
CMD node myworker $BROKER_HOST 9999
```

5 *Considere que el programa 'broker.js' mencionado en Dockerfile A utiliza su primer argumento como* **número de puerto** *para interactuar con procesos clientes. Mediante el Dockerfile A se ha generado la imagen 'broker'. Con esa imagen se iniciará un contenedor en un anfitrión cuya dirección IP es 148.52.2.200.*
Se pretende, además, ejecutar un proceso cliente en un ordenador cuya dirección IP es 148.52.2.145, que debe conectar con el contenedor mencionado mediante la dirección IP 148.52.2.200 y el puerto 9998.
¿Con qué orden Docker deberíamos iniciar ese contenedor basado en broker para que interactúe sin errores con el cliente mencionado?

a

```
docker run broker node mybroker tcp://148.52.2.200:9998 9999
```

b No hay orden posible, ya que esa interacción no es factible: los contenedores Docker solo pueden comunicarse con otros contenedores en su mismo anfitrión.

c No hay ninguna orden Docker con ese propósito. Para desplegar componentes en más de un ordenador debe utilizarse Kubernetes, en lugar de Docker.

d

```
docker run -p 9998:9998 broker
```

6 *Si se pretendiera utilizar Dockerfile A para generar una imagen llamada 'broker', sería necesario…*

a Todas las demás opciones son correctas.

b Ejecutar la orden

```
docker build -t broker .
```

allí donde esté Dockerfile A.

c Tener un fichero `mybroker.js` allí donde esté Dockerfile A.

d Tener un fichero `tsr.js` allí donde esté Dockerfile A.

7 *Se va a utilizar el Dockerfile A para generar una imagen llamada 'broker' y el Dockerfile B para generar una imagen llamada 'worker'. No va a haber errores en estas dos operaciones. Si cuando se inicie un contenedor 'broker', su dirección IP es 192.168.1.5,*
¿se podrá iniciar posteriormente algún contenedor 'worker' que interactúe con ese contenedor 'broker'?

a Ninguna de las demás opciones es correcta.

b No, pues habríamos tenido que sustituir $BROKER_HOST por 192.168.1.5 antes de utilizar Dockerfile B para crear la imagen 'worker'.

c Sí. Por ejemplo, usando la orden docker-compose y un fichero docker-compose.yml en el que worker dependa de broker y se asigne valor a BROKER_HOST.

d No, pues ambos contenedores deberían iniciarse simultáneamente para que worker pueda tener algún valor correcto en su variable BROKER_HOST.

8 *Considere este fichero docker-compose.yml:*

```
version: '2'
services:
   one:
      image: zzz
      links:
         - two
      environment:
         - SERVER_IP=two
         - SERVER_PORT=8080
   two:
      image: yyy
      expose:
         - "8080"
         - "8443"
```

Determine cuál de las siguientes afirmaciones sobre ese fichero es FALSA:

a El programa a ejecutar en los contenedores que utilicen la imagen 'zzz' utiliza dos variables de entorno: SERVER_IP y SERVER_PORT.

b Podríamos iniciar cinco instancias del componente 'one' y una del componente 'two' si utilizáramos la orden

```
docker-compose up -d --scale one=5
```

allí donde esté el fichero.

c Al realizar un docker-compose up allí donde esté el fichero, se iniciará en primer lugar una instancia del componente 'two' y posteriormente una instancia de 'one'.

d Si las imágenes 'zzz' e 'yyy' no existieran en la máquina donde se utilizará este fichero, la orden docker-compose up se encargaría de construirlas localmente.

9 *En los servicios escalables:*

a Debe mantenerse la tolerancia al particionado y la consistencia fuerte.

b Se permite mantener la disponibilidad, la tolerancia al particionado y la consistencia fuerte.

c Debe mantenerse la disponibilidad y la consistencia fuerte.

d Debe mantenerse la disponibilidad y la tolerancia al particionado.

10 *Se ha desarrollado una aplicación distribuida cuyas operaciones requieren, generalmente, un breve intervalo de cómputo (menos de 10 ms), pero cuando modifican el estado cada una de ellas suele sobreescribir muchos datos (100 MB, por término medio).*
¿Qué modelo de replicación interesaría utilizar para optimizar el rendimiento y la disponibilidad de este servicio evitando inconsistencias?

a Ninguno. En un servicio con esas características se obtendrá un rendimiento óptimo con una sola instancia. No merece la pena replicar nada.

b El activo, pues no necesita transferencias de estado entre las réplicas.

c El multi-máster, pues puede retrasar las transferencias de estado tanto como nos interese.

d El pasivo, pues su reconfiguración es inmediata en caso de fallo, y eso compensa sobradamente cualquier tipo de pérdida de rendimiento.

11 *En un sistema hay dos procesos P1 y P2 que replican una misma variable 'y'. Cada acceso utiliza la notación operaciónProceso(variable)valor. ¿Cuál de las siguientes ejecuciones respetaría el modelo de consistencia secuencial, pero no respetaría la consistencia FIFO?*

a Ninguna, pues toda ejecución que cumpla el modelo secuencial también respetará el modelo FIFO.

b

```
W1(y)4, R2(y)4, W2(y)3, R1(y)3
```

c

```
W1(y)2, W1(y)5, R2(y)2, R2(y)5
```

d

```
W1(y)4, R2(y)4, W2(y)3, W2(y)7, R1(y)7, R1(y)3
```

12 *¿Cuál de las siguientes afirmaciones sobre el modelo de replicación pasivo es CIERTA?*

a Este modelo tolera operaciones no deterministas porque cada operación es ejecutada por una sola réplica y sus efectos son normalmente transferidos a las demás.

b Los fallos en la réplica primaria no requieren ninguna reconfiguración del servicio replicado.

c El fallo de una réplica secundaria puede ser detectado fácilmente por los clientes del servicio replicado.

d Los clientes de un servicio replicado bajo este modelo recibirán normalmente más de una respuesta para cada una de las peticiones realizadas.

13 *Si en un servicio distribuido se decide gestionar los posibles fallos de conectividad (es decir, particiones de la red) mediante el modelo de partición primaria, entonces…*

a La consistencia causal no se podrá mantener cuando haya particiones en la red, pues lo que se haga en un grupo aislado de nodos no se podrá transmitir a los demás.

b En caso de particionado de la red, todos los clientes de ese servicio seguirán utilizándolo y observarán que el servicio sigue disponible.

c Se podrá mantener una consistencia fuerte, como pueda ser la secuencial, en el subgrupo de nodos mayoritario, en caso de que exista.

d El despliegue de ese servicio no se podrá realizar, pues no hay manera de detectar si la comunicación con otros nodos resulta posible o no.

14 *La replicación multi-máster …*

a Las restantes cláusulas son todas ciertas.

b Presenta mínima sobrecarga.

c No admite operaciones indeterministas.

d No permite que puedan surgir inconsistencias.

15 *Los contenedores permiten desplegar un servicio más rápido que las máquinas virtuales porque…*

a Un gestor de contenedores limita el uso de recursos para los procesos del sistema anfitrión, mientras un hipervisor no puede hacer eso.

b Están específicamente diseñados para ser desplegados sobre sistemas en la nube.

c No disponen de sistemas de ficheros, por lo que requieren menor espacio que una máquina virtual.

d No necesitan un sistema operativo 'huésped', por lo que completan mucho antes su arranque.

16 *En MongoDB:*

a Las restantes cláusulas son todas falsas.

b Los datos se encuentran en los procesos mongos.

c Los procesos directores mongod encaminan las peticiones.

d Los servidores de configuración son procesos mongos.

17 *Para que un servicio sea elástico se necesita…*

a Ninguna de las demás opciones es correcta.

b … administradores expertos en su despliegue que estén diariamente revisando su configuración.

c … que su proveedor contrate un número fijo y grande de máquinas virtuales para que pueda llegar sin problemas a su mayor grado de escalabilidad.

d …que todos sus componentes estén programados utilizando algún lenguaje orientado a objetos, pues así se facilita su monitorización.

18 *En la segunda parte de la primera sesión de la Práctica 3, se indica que podríamos ejecutar 2 clientes, 5 trabajadores y 1 broker mediante la orden:*

```
docker-compose up -d --scale cli=2 --scale wor=5
```

¿Con qué orden, sin otros argumentos adicionales, podríamos parar y eliminar todos esos contenedores?

a Ninguna de las demás opciones es correcta.

b docker-compose down

c docker-compose kill

d docker-compose rm

19 *En la segunda sesión de la práctica 3 se añadió un componente 'logger' al sistema CBW desplegado en la primera sesión. Eso implicó la adición de cierto fragmento al fichero 'docker-compose.yml' correspondiente, similar al siguiente:*

```
log:
   image: logger
   build: ./logger/
   expose:
      - "9995"
   volumes:
      - /tmp/logger.log:/tmp/cbwlog
   environment:
      - LOGGER_DIR=/tmp/cbwlog
```

¿Fue necesaria alguna ampliación más, sobre el fichero docker-compose.yml utilizado en la primera sesión de la práctica 3?

a Todas las demás opciones son correctas.

b Sí, tanto el componente 'cli' como el componente 'wor' necesitan una cláusula 'links:' que indique que ambos dependen de 'log'.

c Sí, el componente 'bro' necesita añadir una cláusula 'volumes:' con un contenido idéntico al mostrado aquí para 'log'.

d Sí, el componente 'bro' necesita cláusulas 'links:' y 'environment:' para indicar que depende de 'log' y asignar valores a sus variables de entorno, respectivamente.

20 *En la primera sesión de la práctica 3 se utilizó un docker-compose.yml similar al siguiente para realizar el despliegue del servicio CBW:*

```
version: '2'
services:
   cli:
      image: client
      build: ./client/
      links:
        - A
      environment:
        - C_HOST=A
        - C_PORT=9998
   wor:
      image: worker
      build: ./worker/
      links:
        - B
      environment:
        - D_HOST=B
        - D_PORT=9999
   bro:
      image: broker
      build: ./broker/
      expose:
        - "9998"
        - "9999"
```

¿Qué valores debían utilizarse en él, en lugar de los A, B, C y D que hemos usado aquí?

a A=bro, B=bro, C=BROKER, D=BROKER

b A=cli, B=wor, C=CLIENT, D=WORKER

c A=broker, B=broker, C=BROKER, D=BROKER

d A=wor, B=cli, C=WORKER, D=CLIENT

21 *En la segunda sesión de la práctica 3 se necesitabaque algunos componentes enviaran sus mensajes de traza a un nuevo componente,el logger.Para que eso fuera posible se necesitaba:*

a Añadir un socket de tipo PUSH en esos componentes y enviar con ese socket una copia de cada mensaje de traza al logger.

b Añadir un socket de tipo ROUTER en cada uno de ellos y enviar con ese socket una copia de cada mensaje de traza al logger.

c Ninguna de las demás alternativas es correcta.

d Añadir un socket de tipo PUB en cada uno de ellos y enviar con ese socket una copia de cada mensaje de traza al logger.

22 *En la tercera sesión de la práctica 3 se llegó a modificar la página de muestra ofrecida por WordPress y se comprobó que los cambios se mantenían tras haber parado el servicio y haberlo iniciado de nuevo.*
¿Cómo se lograba esa persistencia en los cambios aplicados?

a No era necesario aplicar ningún cambio, ya que la configuración original ya utilizaba las secciones 'volume:' necesarias, con sus valores adecuados.

b Modificando los ficheros de configuración correspondientes, para añadir cláusulas 'volume:' en el componente 'mariadb'.

c Modificando los ficheros de configuración correspondientes, para añadir cláusulas 'volume:' en el componente 'wordpress'.

d Eliminando una línea de configuración que asignaba el valor 'false' a cierta variable de entorno o cambiando su valor para que fuera 'true'.

23 *Para automatizar el despliegue de un servicio distribuido se necesita:*

a Tener un buen administrador del sistema en cada máquina. Ellos supervisarán cuidadosamente la configuración y avance de cada etapa del despliegue.

b Todos los componentes del servicio se habrán programado en el mismo lenguaje y usarán una configuración similar.

c Todas las demás opciones son correctas.

d Disponer de una herramienta y de un plan de despliegue que configure adecuadamente todos los componentes.

24 *En la tercera sesión de la práctica 3 se desplegó un servicio con dos componentes, MariaDB y WordPress, en un mismo ordenador. Para completar ese despliegue, debíamos:*

a Descargar el fichero docker-compose.yml indicado en el enunciado, modificar algunas de sus clásulas e iniciar su despliegue con docker-compose start.

b Buscar las imágenes Docker más recientes de esos componentes, modificar sus Dockerfile para facilitar su interacción e iniciar manualmente cada contenedor.

c Desplegar ambos componentes conjuntamente, empleando para ello una solución basada en Kubernetes, ya descrita en el enunciado.

d Descargar el fichero docker-compose.yml indicado en el enunciado y solicitar su despliegue con la orden docker-compose up.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

D

DNI NIE PASAPORTE

0 0 0 0 0 0 0 0 0

1 1 1 1 1 1 1 1 1

2 2 2 2 2 2 2 2 2

3 3 3 3 3 3 3 3 3

4 4 4 4 4 4 4 4 4

5 5 5 5 5 5 5 5 5

6 6 6 6 6 6 6 6 6

7 7 7 7 7 7 7 7 7

8 8 8 8 8 8 8 8 8

9 9 9 9 9 9 9 9 9

ETSINF - TSR

Segundo Parcial - 23/01/2023

Apellidos .....

Nombre .....

Marque así Así NO marque

NO BORRAR, corregir con corrector

Segundo Parcial

	a	b	c	d
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				