

Aquest examen té un valor de 10 punts, i consta de 24 qüestions de tipus test. Cada qüestió planteja 4 alternatives i té una única resposta correcta. Cada resposta correcta aporta 10/24 punts, i cada error descompta 10/72 punts. Ha de contestar en la fulla de respostes.

- 1** El patró de comunicacions DEALER-ROUTER pressuposa que el socket DEALER serà utilitzat per un procés client i el socket ROUTER per un servidor. Si aqueixos dos processos encara no han intercanviat cap missatge, pot el servidor enviar el primer missatge al client utilitzant un socket ROUTER?
- a Sí, perquè el socket ROUTER és bidireccional i asincrònic.
 - b Normalment no, perquè desconexirà la identitat a utilitzar per a comunicar-se amb el DEALER del client.
 - c No, perquè el socket ROUTER únicament podrà rebre missatges.
 - d Normalment sí, encara que per a això el client haurà hagut de connectar prèviament el seu socket DEALER a aqueix socket ROUTER.
- 2** Suppose que un procés P1 està utilitzant un socket DEALER anomenat `so1`, que ja ha sigut connectat correctament a un socket `so2` de tipus ROUTER utilitzat per un procés P2. P1 utilitza aquesta instrucció per a enviar una cadena a P2: `so1.send('hola')`. Quin codi necessitarà P2 en el mètode `on()` del seu socket `so2` per a escriure en pantalla el contingut ('hola') d'aqueix missatge?
- a `so2.on('message', (m) => console.log(m+'))`
 - b `so2.on('message', (i, j) => console.log(j+'))`
 - c `so2.on('message', (i, s, x) => console.log(x+'))`
 - d `so2.on('data', (m) => console.log(m+'))`
- 3** En una certa activitat del Tema 3 se sol·licitava desenvolupar un servei de xat amb dos components: client i servidor. El client utilitzava un socket PUSH i un socket SUB. Per la seua part, el servidor utilitzava un socket PULL i un PUB. Es podria desenvolupar una implementació equivalent d'aquest servei utilitzant un únic socket ROUTER en el servidor i un únic socket DEALER en cada client?
- a No, perquè els clients estarien forçats a utilitzar `bind()` en el seu DEALER i el servidor no podria connectar amb tots ells en desconèixer les seues adreces.
 - b No, perquè el servidor desconexirà quina identitat utilitzarà el DEALER de cada client.
 - c Sí, perquè tots dos sockets són bidireccionals i el servidor sap quants clients s'han registrat en el servei, així com la identitat utilitzada en cadascun dels seus DEALERS.
 - d No, perquè el socket DEALER no pot emular simultàniament el comportament d'un socket PUSH i un socket SUB.
- 4** Entre altres tasques, el desplegament d'un servei distribuït comprèn:
- a L'escriptura o desenvolupament dels programes que componen l'aplicació distribuïda i l'execució de la qual generarà aqueix servei.
 - b La parada de cadascun dels components del servei, en la seqüència adequada per a evitar inconsistències durant aqueixa desactivació.
 - c La depuració dels programes que componen l'aplicació distribuïda i l'execució de la qual generarà aqueix servei.
 - d Totes les altres opcions són correctes.

Considere aquests fitxers Dockerfile en les pròximes qüestions relacionades amb 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 S'utilitzarà el Dockerfile A per a generar una imatge anomenada 'broker' i el Dockerfile B per a generar una imatge anomenada 'worker'. No hi haurà errors en aquestes dues operacions. Si quan s'iniciï un contenidor 'broker', la seua adreça IP és 192.168.1.5, es podrà iniciar posteriorment algun contenidor 'worker' que interaccione amb aqueix contenidor 'broker'?

- a No, perquè hauríem hagut de substituir \$BROKER_HOST per 192.168.1.5 abans d'utilitzar Dockerfile B per a crear la imatge 'worker'.
- b Sí. Per exemple, usant l'ordre docker-compose i un fitxer docker-compose.yml en el qual worker depenga de broker i s'assigne valor a BROKER_HOST.
- c No, perquè tots dos contenidors haurien d'iniciar-se simultàniament perquè worker pugui tenir algun valor correcte en la seua variable BROKER_HOST.
- d Cap de les altres opcions és correcta.

6 Si es pretenguera utilitzar Dockerfile A per a generar una imatge anomenada 'broker', seria necessari...

a Executar l'ordre

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

allí on estiga Dockerfile A.

- b Tenir un fitxer mybroker.js allí on estiga Dockerfile A.
- c Tenir un fitxer tsr.js allí on estiga Dockerfile A.
- d Totes les altres opcions són correctes.

7 Considere que el programa 'broker.js' esmentat en Dockerfile A utilitza el seu primer argument com a **número de port** per a interactuar amb processos clients. Mitjançant el Dockerfile A s'ha generat la imatge 'broker'. Amb aqueixa imatge s'iniciarà un contenidor en un amfitrió l'adreça IP del qual és 148.52.2.200.

Es pretén, a més, executar un procés client en un ordinador l'adreça IP del qual és 148.52.2.145, que ha de connectar amb el contenidor esmentat mitjançant l'adreça IP 148.52.2.200 i el port 9998.

Amb quina ordre Docker hauríem d'iniciar aqueix contenidor basat en broker perquè interaccione sense errors amb el client esmentat?

- a No hi ha ordre possible, ja que aqueixa interacció no és factible: els contenidors Docker només poden comunicar-se amb altres contenidors en el seu mateix amfitrió.
- b No hi ha cap ordre Docker amb aqueix propòsit. Per a desplegar components en més d'un ordinador ha d'utilitzar-se Kubernetes, en lloc de Docker.

c

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

d

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

8 Considere aquest fitxer `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 quina de les següents afirmacions sobre aqueix fitxer és FALSA:

- a** Podríem iniciar cinc instàncies del component 'one' i una del component 'two' si utilitzàrem l'ordre

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

allí on estiga el fitxer.

- b** En realitzar un `docker-compose up` allí on estiga el fitxer, s'iniciarà en primer lloc una instància del component 'two' i posteriorment una instància de 'one'.
- c** Si les imatges 'zzz' i 'yyy' no existiren en la màquina on s'utilitzarà aquest fitxer, l'ordre `docker-compose up` s'encarregaria de construir-les localment.
- d** El programa a executar en els contenidors que utilitzen la imatge 'zzz' utilitza dues variables d'entorn: `SERVER_IP` i `SERVER_PORT`.

9 Els contenidors permeten desplegar un servei més ràpid que les màquines virtuals perquè...

- a** Estan específicament dissenyats per a ser desplegats sobre sistemes en el núvol.
- b** No disposen de sistemes de fitxers, per la qual cosa requereixen menor espai que una màquina virtual.
- c** No necessiten un sistema operatiu 'hoste' (és a dir, diferent al del amfitrió), per la qual cosa completen molt abans la seua arrancada.
- d** Un gestor de contenidors limita l'ús de recursos per als processos del sistema amfitrió, mentre un hipervisor no pot fer això.

10 Si en un servei distribuït es decideix gestionar les possibles fallades de connectivitat (és a dir, particions de la xarxa) mitjançant el model de partició primària, llavors...

- a** En cas de particionat de la xarxa, tots els clients d'aqueix servei continuaran utilitzant-lo i observaran que el servei continua disponible.
- b** Es podrà mantenir una consistència forta, com puga ser la seqüencial, en el subgrup de nodes majoritari, en cas que existisca.
- c** El desplegament d'aqueix servei no es podrà realitzar, perquè no hi ha manera de detectar si la comunicació amb altres nodes resulta possible o no.
- d** La consistència causal no es podrà mantenir quan hi haja particions en la xarxa, perquè el que es faça en un grup aïllat de nodes no es podrà transmetre als altres.

11 *Quina de les següents afirmacions sobre el model de replicació passiu és CERTA?*

- a** Les fallades en la rèplica primària no requereixen cap reconfiguració del servei replicat.
- b** La fallada d'una rèplica secundària pot ser detectada fàcilment pels clients del servei replicat.
- c** Els clients d'un servei replicat amb aquest model rebran normalment més d'una resposta per a cadascuna de les peticions realitzades.
- d** Aquest model tolera operacions no deterministes perquè cada operació és executada per una única rèplica i els seus efectes són normalment transferits a les altres.

12 *S'ha desenvolupat una aplicació distribuïda les operacions de la qual requereixen, generalment, un breu interval de còmput (menys de 10 ms), però quan modifiquen l'estat cadascuna d'elles sol sobreescriure moltes dades (100 MB, de mitjana). Quin model de replicació interessaria utilitzar per a optimitzar el rendiment i la disponibilitat d'aquest servei evitant inconsistències?*

- a** L'actiu, perquè no necessita transferències d'estat entre les rèpliques.
- b** El multi-màster, perquè pot retardar les transferències d'estat tant com ens interesse.
- c** El passiu, perquè la seua reconfiguració és immediata en cas de fallada, i això compensa àmpliament qualsevol tipus de pèrdua de rendiment.
- d** Cap. En un servei amb aqueixes característiques s'obtindrà un rendiment òptim amb una sola instància. No val la pena replicar res.

13 *En un sistema hi ha dos processos P1 i P2 que repliquen una mateixa variable 'i'. Cada accés utilitza la notació operacióProcés(variable)valor. Quina de les següents execucions respectaria el model de consistència seqüencial, però no respectaria la consistència FIFO?*

a

W1(i)4, R2(i)4, W2(i)3, R1(i)3

b

W1(i)2, W1(i)5, R2(i)2, R2(i)5

c

W1(i)4, R2(i)4, W2(i)3, W2(i)7, R1(i)7, R1(i)3

d Cap, perquè tota execució que complisca el model de consistència seqüencial també respectarà el model FIFO.

14 *En els serveis escalables:*

- a** Es permet mantenir la disponibilitat, la tolerància al particionat i la consistència forta.
- b** Ha de mantenir-se la disponibilitat i la consistència forta.
- c** Ha de mantenir-se la disponibilitat i la tolerància al particionat.
- d** Ha de mantenir-se la tolerància al particionat i la consistència forta.

15 *La replicació multi-màster...*

- a** Presenta mínima sobrecàrrega.
- b** No admet operacions indeterministes.
- c** No permet que puguin sorgir inconsistències.
- d** Les restants clàusules són totes certes.

16 *Per que un servei siga elàstic es necessita...*

- a** ...administradors experts en el seu desplegament que estiguen diàriament revisant la seua configuració.
- b** ...que el seu proveïdor contracte un número fix i gran de màquines virtuals perquè puga arribar sense problemes al seu major grau d'escalabilitat.
- c** ...que tots els seus components estiguen programats utilitzant algun llenguatge orientat a objectes, perquè així es facilita el seu monitoratge.
- d** Cap de les altres opcions és correcta.

17 *En MongoDB:*

- a** Les dades es troben en els processos mongos.
- b** Els processos directors mongod encaminen les peticions.
- c** Els servidors de configuració són processos mongos.
- d** Les restants clàusules són totes falses.

18 *En la primera sessió de la pràctica 3 es va utilitzar un docker-compose.yml similar al següent per a realitzar el desplegament del servei 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"
```

Quins valors havien d'utilitzar-se en ell, en lloc dels A, B, C i D que hem usat aquí?

- a** A=cli, B=wor, C=CLIENT, D=WORKER
- b** A=broker, B=broker, C=BROKER, D=BROKER
- c** A=wor, B=cli, C=WORKER, D=CLIENT
- d** A=bro, B=bro, C=BROKER, D=BROKER

- 19** En la segona part de la primera sessió de la Pràctica 3, s'indica que podríem executar 2 clients, 5 treballadors i 1 broker mitjançant l'ordre:

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

Amb quina ordre, sense altres arguments addicionals, podríem parar i eliminar tots aqueixos contenidors?

- a** docker-compose down
- b** docker-compose kill
- c** docker-compose rm
- d** Cap de les altres opcions és correcta.

- 20** En la segona sessió de la Pràctica 3 es va afegir un component 'logger' al sistema CBW desplegat en la primera sessió. Això va implicar l'addició d'un cert fragment al fitxer 'docker-compose.yml' corresponent, similar al següent:

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

Va ser necessària alguna ampliació més, sobre el fitxer docker-compose.yml utilitzat en la primera sessió de la Pràctica 3?

- a** Sí, tant el component 'cli' com el component 'wor' necessiten una clàusula 'links:' que indiqui que tots dos depenen de 'log'.
- b** Sí, el component 'bro' necessita afegir una clàusula 'volumes:' amb un contingut idèntic al mostrat ací per a 'log'.
- c** Sí, el component 'bro' necessita clàusules 'links:' i 'environment:' per a indicar que depèn de 'log' i assignar valors a les seues variables d'entorn, respectivament.
- d** Totes les altres opcions són correctes.

- 21** En la segona sessió de la Pràctica 3 es necessitava que alguns components enviaren els seus missatges de traça a un nou component, el logger. Perquè això fora possible es necessitava:

- a** Afegir un socket de tipus ROUTER en cadascun d'ells i enviar amb aqueix socket una còpia de cada missatge de traça al logger.
- b** Cap de les altres alternatives és correcta.
- c** Afegir un socket de tipus PUB en cadascun d'ells i enviar amb aqueix socket una còpia de cada missatge de traça al logger.
- d** Afegir un socket de tipus PUSH en aqueixos components i enviar amb aqueix socket una còpia de cada missatge de traça al logger.

- 22** En la tercera sessió de la Pràctica 3 es va desplegar un servei amb dos components, MariaDB i WordPress, en un mateix ordinador. Per a completar aqueix desplegament, havíem de:

- a** Cercar les imatges Docker més recents d'aqueixos components, modificar els seus Dockerfile per a facilitar la seua interacció i iniciar manualment cada contenidor.
- b** Desplegar tots dos components conjuntament, emprant per a això una solució basada en Kubernetes, ja descrita en l'enunciat.
- c** Descarregar el fitxer docker-compose.yml indicat en l'enunciat i sol·licitar el seu desplegament amb l'ordre docker-compose up.
- d** Descarregar el fitxer docker-compose.yml indicat en l'enunciat, modificar algunes de les seues clàusules i iniciar el seu desplegament amb docker-compose start.

23 *En la tercera sessió de la pràctica 3 es va arribar a modificar la pàgina de mostra oferida per Word-Press i es va comprovar que els canvis es mantien després d'haver parat el servei i haver-lo iniciat de nou.*

Com s'aconseguia aqueixa persistència en els canvis aplicats?

- a** Modificant els fitxers de configuració corresponents, per a afegir clàusules 'volume:' en el component 'mariadb'.
- b** Modificant els fitxers de configuració corresponents, per a afegir clàusules 'volume:' en el component 'wordpress'.
- c** Eliminant una línia de configuració que assignava el valor 'false' a una certa variable d'entorn o canviant el seu valor perquè fora 'true'.
- d** No era necessari aplicar cap canvi, ja que la configuració original ja utilitzava les seccions 'volume:' necessàries, amb els seus valors adequats.

24 *Per a automatitzar el desplegament d'un servei distribuït es necessita:*

- a** Tots els components del servei s'hauran programat en el mateix llenguatge i usaran una configuració similar.
- b** Totes les altres opcions són correctes.
- c** Disposar d'una eina i d'un pla de desplegament que configure adequadament tots els components.
- d** Tenir un bon administrador del sistema en cada màquina. Ells supervisaran acuradament la configuració i avanç de cada etapa del desplegament.



DNI		NIE		PASSAPORT	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
0	0	0	0	0	0
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
1	1	1	1	1	1
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2	2	2	2	2	2
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
3	3	3	3	3	3
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
4	4	4	4	4	4
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
5	5	5	5	5	5
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
6	6	6	6	6	6
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
7	7	7	7	7	7
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
8	8	8	8	8	8
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
9	9	9	9	9	9
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

ETSINF - TSR

Segon Parcial - 23/01/2023

Cognoms

Nom

Marque així

Així NO marque



NO ESBORRAR, corregir amb corrector

Segon Parcial

	a	b	c	d
1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
3	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
4	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
5	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
6	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
7	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
8	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
9	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
10	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
11	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
12	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
13	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
14	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
15	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
16	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
17	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
18	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
19	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
20	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
21	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
22	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
23	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
24	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>