

Aquest examen val 10 punts i consta de 25 preguntes. Cada pregunta planteja 4 alternatives i només té una resposta correcta. Un cop descartada la pitjor resposta, cada resposta correcta aporta 10/24 punts, i cada error resta 10/72 punts. Has de respondre al full de respostes.

Considerem aquests programes JavaScript:

```
1 // Program: client1.js
2 const zmq = require('zeromq')
3 const rq = zmq.socket('req')
4 rq.connect('tcp://127.0.0.1:8888')
5 rq.send('Hello')
6 rq.on('message', (msg) => {
7   console.log('Response: ' + msg)
8   rq.close()
9 })
```

```
1 // Program: server1.js
2 const zmq = require('zeromq')
3 const rp = zmq.socket('rep')
4 rp.bindSync('tcp://127.0.0.1:8888')
5 rp.on('message', (msg) => {
6   console.log('Request: ' + msg)
7   rp.send('World')
8 })
```

Tingueu en compte que aquests programes utilitzen el patró de comunicació REQ-REP. Tenim previst canviar els dos programes per utilitzar el patró DEALER-ROUTER, proporcionant la mateixa funcionalitat i utilitzant un **nombre mínim de segments** als missatges

1 Els canvis que s'han d'aplicar al programa *client1.js* són:

- a const rq = zmq.socket('router') // Línia 3
- b const rq = zmq.socket('dealer') // Línia 3  
rq.send(['', 'Hello']) // Línia 5
- c const rq = zmq.socket('dealer') // Línia 3
- d Cap altra opció és correcta.

2 Els canvis que s'aplicaran al programa *server1.js* són:

- a const rp = zmq.socket('dealer') // Línia 3
- b const rp = zmq.socket('router') // Línia 3  
rp.on('message', (who, msg) => { // Línia 5  
rp.send([who, 'World']) // Línia 7
- c const rp = zmq.socket('router') // Línia 3  
rp.on('message', (who, sep, msg) => { // Línia 5  
rp.send([who, 'World']) // Línia 7
- d const rp = zmq.socket('router') // Línia 3

3 Considerem un sistema format per un client i dos servidors. Els servidors s'uneixen i utilitzen un socket REP cadascun, mentre que el client utilitza un socket REQ connectat als dos sockets REP. El client envia una sol·licitud per segon i no acaba mai. En aquest sistema, si un dels servidors s'atura, els altres dos processos es bloquegen i no se'ls transmet ni se'ls lliura cap altre missatge. Suposem que substituïm el socket REQ en el procés del client per un socket DEALER, i adaptem adequadament la gestió dels segments dels missatges per tal d'assegurar que tots els missatges enviats i rebuts es lliuraran. Amb aquesta nova configuració, es reinicien els tres processos. Què passa en aquest nou escenari quan es mata un dels servidors?

- a El mateix que en el primer sistema: la comunicació es bloqueja i no s'entrega cap altra sol·licitud al seu destí.
- b Totes les sol·licituds del client s'entreguen al servidor restant i no es perd cap missatge.
- c El client genera una excepció que normalment avorta la seua execució quan s'acaba el servidor.
- d La comunicació no es bloqueja i el servidor restant encara rebrà la meitat de les sol·licituds enviades pel client.

**4** Quina de les següents etapes del cicle de vida del programari no forma part del desplegament del programari?

- a Disseny de programari.
- b Eliminació del programari.
- c Instal·lació i configuració del programari.
- d Actualització del programari.

**5** Entre aquestes opcions, quina és la millor eina per desplegar un servei en diversos amfitrions?

- a Kubernetes
- b docker
- c docker-compose
- d Desplegament manual

**6** Per executar un contenidor Docker en un ordinador amfitrió determinat, necessitem...

- a Un hipervisor.
- b Un sistema operatiu convidat instal·lat i configurat adequadament a la imatge que s'ha d'executar.
- c Un sistema operatiu amfitrió diferent del que s'assumeix a la imatge que s'ha d'executar.
- d Cap altra opció és certa.

Suposem que la imatge 'tsr-zmq' existeix i té el contingut i la funcionalitat que s'explica al Tema 4 i al Laboratori 3. Suposem també que aquest Dockerfile (que s'anomenarà 'Dockerfile A', tot i que el seu nom real és 'Dockerfile') s'ha desat al directori /home/user/docker/config:

```
FROM tsr-zmq
COPY ./tsr.js tsr.js
RUN mkdir broker
WORKDIR broker
COPY ./broker.js mybroker.js
EXPOSE 9998 9999
CMD node mybroker 9998 9999
```

**7** Considerem el Dockerfile A. Quina ordre s'hauria d'utilitzar per crear una imatge anomenada 'broker' si el directori de treball actual del nostre shell és /home/user/docker?

- a docker commit config broker
- b docker run broker
- c docker build -t broker config
- d docker build -t broker

**8** Considerem el Dockerfile A. On s'han de col·locar la imatge 'tsr-zmq' i els fitxers 'tsr.js' i 'broker.js' per tal d'utilitzar amb èxit aquest Dockerfile per crear una imatge nova?

- a Tots els elements necessaris (tsr-zmq, tsr.js i broker.js) s'han de col·locar a /home/user/docker/config
- b La imatge tsr-zmq hauria d'estar al repositori local o estar present a hub.docker.com, mentre que els fitxers haurien d'estar a /home/user/docker/config
- c Tots els elements necessaris (tsr-zmq, tsr.js i broker.js) haurien d'estar en algun lloc de l'amfitrió local, ja que el motor Docker els cercarà i els trobarà sense cap problema.
- d Cap altra opció és correcta.

**9** Considerem el Dockerfile A. Quina ordre executarà l'interpret 'bash' en un contenidor que utilitza la imatge 'broker1' generada amb aquest Dockerfile A?

- a Cap ordre docker pot assolir aquest objectiu.
- b docker run bash
- c docker run -i -t broker1 bash
- d docker-compose up bash

- 10** Considerem que aquest Dockerfile s'ha utilitzat per crear una imatge anomenada 'broker2':

```
FROM tsr-zmq
COPY ./tsr.js tsr.js
RUN mkdir broker
WORKDIR broker
COPY ./broker.js mybroker.js
EXPOSE 9998 9999
ENTRYPOINT [ "/usr/bin/node", "mybroker" ]
CMD [ "9998", "9999" ]
```

*Podem utilitzar 'broker2' per executar l'interpret d'ordres 'bash' en un contenidor?*

- a** No, ja que la imatge broker2 no es pot crear, perquè un Dockerfile no pot combinar ENTRYPOINT i CMD.
- b** Per omissió, no, ja que el programa que s'executa en contenidors generat a partir de la imatge broker2 és node mybroker
- c** Sí, amb aquesta ordre:  
docker run -i -t broker2 bash
- d** No, ja que quan combinem ENTRYPOINT i CMD al mateix Dockerfile només es considera l'últim d'ells al fitxer, i en aquest exemple CMD té valors no vàlids.

- 11** La primera sessió del laboratori 3 comença amb un desplegament manual d'un sistema broker-treballador-client. En aquest sistema, el broker s'inicia primer i els altres dos components han de conèixer l'adreça IP del contenidor del broker. L'ordre (o conjunt d'ordres) que proporciona aquesta informació és...

- a** docker images
- b** docker inspect id  
Suposant que l'identificador del contenidor del broker s'ha trobat anteriorment.
- c** docker-compose
- d** Cap altra opció és correcta.

- 12** Considerem que aquest docker-compose.yml es troba al directori /home/user/docker

```
version: '2'
services:
  svca:
    image: imga
    links:
      - svcb
    environment:
      - B_HOST=svcb
  svcb:
    image: imgb
    links:
      - svcc
    environment:
      - C_HOST=svcc
    expose:
      - "9999"
  svcc:
    image: imgc
    expose:
      - "9999"
```

*Trieu la frase FALSA sobre el servei que es desplegarà amb aquest fitxer.*

- a** Els components d'aquest servei s'iniciaran en aquest ordre: svcc, svcb, svca.
- b** Aquest servei es pot desplegar, però els seus components svcc i svcb escoltaran el mateix port amfitrió (9999) alhora i per això generaran un conflicte.
- c** Podem desplegar una instància dels components svcc i svcb i sis instàncies del component svca utilitzant aquesta ordre a /home/user/docker:  
docker-compose up -d --scale svca=6
- d** Un cop desplegat el servei, podem aturar i eliminar tots els contenidors iniciats mitjançant aquesta ordre a /home/user/docker:  
docker-compose down

- 13** Aquest és un esquelet del fitxer `docker-compose.yml` a utilitzar a la segona meitat de la primera sessió de la Pràctica 3 per automatitzar el desplegament d'un sistema CBW:

```
version: '2'
services:
  cli:
    image: client
    build: ./client/
    links:
      - W
    environment:
      - BROKER_HOST=X
      - BROKER_PORT=9998
  wor:
    image: worker
    build: ./worker/
    links:
      - Y
    environment:
      - BROKER_HOST=Z
      - BROKER_PORT=9999
  bro:
    image: broker
    build: ./broker/
    expose:
      - "9998"
      - "9999"
```

*Els valors necessaris perquè W, X, Y i Z omplin aquest fitxer `docker-compose.yml` amb un contingut mínim per gestionar correctament aquest desplegament (és a dir, per garantir un ordre d'inici i una resolució de dependències adequats) són :*

- a** X=bro, Z=bro i les clàusules `links` no són necessàries i s'han d'eliminar, de manera que W i Y no necessiten cap valor.
- b** W=wor, X=wor, mentre que les clàusules `links` i `environment` de wor s'han d'eliminar, de manera que Y i Z no necessiten cap valor.
- c** No calen clàusules `links` i `environment` per automatitzar aquest desplegament. Es poden eliminar totes aquestes seccions i el desplegament resultant es comportarà correctament. Per tant, no cal cap valor per a W, X, Y i Z.
- d** W=bro, X=bro, Y=bro, Z=bro.

- 14** La segona sessió de Lab 3 introdueix un nou component `logger` al sistema CBW. Quins altres components CBW interactuen amb aquest nou `logger`?

- a** Els clients i els treballadors envien els seus missatges de traça al `logger`.
- b** Només el broker envia els seus missatges de traça al `logger`.
- c** Els clients, els treballadors i el broker envien els seus missatges de seguiment al `logger`.
- d** Només els treballadors envien els seus missatges de traça al `logger`.

- 15** La segona sessió del Lab 3 proposa el desplegament d'un altre tipus de client (un client extern) que s'executarà en un altre ordinador, diferent de l'ordinador amfitrió on `docker` i `docker-compose` gestionen els contenidors CBW. Què cal a la configuració del component del broker per tal d'habilitar aquest client extern?

- a** Es necessita una clàusula `'ports:'` a la secció `'bro'` del fitxer `docker-compose.yml`, per tal de fer correspondre el port 9998 del contenidor del broker amb el port 9998 a l'amfitrió.
- b** Es necessita una clàusula `'ports:'` al `Dockerfile` de la imatge del broker, per tal de fer correspondre el port 9998 del contenidor del broker amb el port 9998 a l'amfitrió.
- c** No cal res especial, només cal esbrinar la IP del component del broker, utilitzant `docker ps` i `docker inspect` per a fer això.
- d** Es necessita una clàusula `'export:'` al `Dockerfile` de la imatge del broker, per tal d'indicar quin port del contenidor del broker hauria d'utilitzar els clients externs.

- 16** El `docker-compose.yml` utilitzat a la tercera sessió de Lab 3 conté una línia com aquesta:

```
image: docker.io/bitnami/mariadb:11.1
```

*Quina és la conseqüència de substituir la part '11.1' d'aquesta línia per 'latest' en aquest desplegament?*

- a** Que, potser, d'aquí a dos anys el fitxer '`docker-compose.yml`' resultant esdevindrà inútil.
  - b** Cap. El desplegament resultant es comporta i es comportarà correctament, independentment d'aquest canvi.
  - c** Un error ja que la paraula 'latest' no pot formar part del nom d'una imatge de Docker.
  - d** Un error ja que 'latest' no pot formar part d'una ruta del sistema.
- 17** Si comparem la replicació activa i la passiva, el model passiu és el model de replicació preferit quan les actualitzacions d'operacions només modifiquen una petita part de l'estat del servei perquè...:
- a** El model passiu ha d'enviar aquestes modificacions a les rèpliques secundàries i aquestes últimes les haurien d'aplicar, mentre que en el model actiu no es necessita cap transferència d'actualització.
  - b** Quan aquestes actualitzacions són petites, les rèpliques secundàries no les necessiten.
  - c** Quan aquestes actualitzacions són petites, es poden transferir de manera asincrònica i sense cap efecte sobre la consistència entre rèpliques resultant.
  - d** Totes les altres opcions són correctes.
- 18** Quant als errors de connectivitat, quan s'assumeix un model de partició primària aleshores:
- a** Tots els nodes del sistema continuen, i la consistència del sistema resultant és molt relaxada.
  - b** La disponibilitat del sistema està assegurada.
  - c** Els processos en subgrups menors s'aturen.
  - d** Els serveis utilitzen replicació passiva.

- 19** Aquest és el fitxer `docker-compose.yml` utilitzat a l'última sessió de Lab 3:

```
version: '2'
services:
  mariadb:
    image: docker.io/bitnami/mariadb:11.1
    volumes:
      - 'mariadb_data:/bitnami/mariadb'
    environment:
      - ALLOW_EMPTY_PASSWORD=yes
      - MARIADB_USER=bn_wordpress
      - MARIADB_DATABASE=bitnami_wordpress
  wordpress:
    image: docker.io/bitnami/wordpress:6
    ports:
      - '80:8080'
      - '443:8443'
    volumes:
      - 'wordpress_data:/bitnami/wordpress'
    depends_on:
      - mariadb
    environment:
      - ALLOW_EMPTY_PASSWORD=yes
      - WORDPRESS_DATABASE_HOST=mariadb
      - WORDPRESS_DATABASE_PORT_NUMBER=3306
      - WORDPRESS_DATABASE_USER=bn_wordpress
      - WORDPRESS_DATABASE_NAME=bitnami_wordpress
    volumes:
      mariadb_data:
        driver: local
      wordpress_data:
        driver: local
```

*Com podem desplegar manualment, amb l'ajuda de l'ordre `docker-compose`, els components `mariadb` i `wordpress` en dos ordinadors diferents?*

- a** No és possible desplegar el sistema, ja que desconexim el contingut dels Dockerfiles que van generar les imatges a utilitzar.
- b** No és possible el desplegament en dos ordinadors, ja que aquest fitxer `docker-compose.yml` ja no va funcionar com es pretenia amb un únic ordinador amfitrió.
- c** Aquest desplegament és possible sense aplicar cap modificació a aquest fitxer.
- d** Dividir el fitxer en dos, un per component, i afegint una secció 'ports:' a la part `mariadb` i un bon valor a `WORDPRESS_DATABASE_HOST` a l'altra.

- 20** *Si comparem els models de consistència causal i de 'cache', quin d'ells és més relaxat que l'altre?*
- a** Causal
  - b** Cache
  - c** Sobre el seu grau de relaxació, són equivalents.
  - d** No es pot comparar el seu grau de relaxació.
- 21** *Si considerem el teorema CAP, quin d'aquests models de consistència es pot respectar quan es necessita disponibilitat en un sistema particionat?*
- a** Estricte
  - b** Causal
  - c** Seqüencial
  - d** Cap
- 22** *Quina d'aquestes alternatives és una diferència correcta entre els models de replicació multi-master i passiu?*
- a** El model multi-master pot utilitzar una rèplica de processament diferent (és a dir, el mestre) per sol·licitud, mentre que el model passiu sempre utilitza el mateix primari.
  - b** En el model passiu cada sol·licitud només s'envia a la rèplica primària, mentre que en el model multi-master cada sol·licitud és emesa pel client a cada rèplica.
  - c** El model multi-master pot gestionar el model de fallades arbitrari, mentre que el model passiu no.
  - d** Totes les altres opcions són certes.
- 23** *Node.js proporciona el seu mòdul 'cluster' per a:*
- a** Executar un programa determinat en un conjunt de processos, compartint alguns recursos (per exemple, l'accés a un socket) per desplegar un servei escalable localment.
  - b** Totes les altres opcions són certes.
  - c** Iniciar diversos fils d'execució en un únic procés.
  - d** Desplegar un servei Node.js en un clúster d'ordinadors.
- 24** *MongoDB utilitza aquest model de replicació:*
- a** Replicació activa
  - b** Replicació passiva
  - c** Replicació multi-master
  - d** Sense replicació
- 25** *Per millorar la seua escalabilitat, MongoDB utilitza:*
- a** Rèpliques secundàries llegibles.
  - b** Particionament de bases de dades horitzontal.
  - c** Totes les altres opcions són correctes.
  - d** Una memòria cau de configuració als seus components mongos.



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 - 23/01/2024

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"/>
25	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>