Universidad de Barcelona

Arthur Font Gouveia 20222613

Marc Colominas Casabella 20081692

Sistemes Operatius II

Pràctica 1

Barcelona

2021

Índex

I bash dóna el següent missatge de error:

bash: fork: retry: Resource temporarily unavailable

Això passa perquè hem limitat el nombre de processos del contenidor a 32, aleshores no n'hi ha prou recursos per executar un altre procés (comanda).

Secció - Docker: connexió via socket

Pregunta: Proveu d'executar els dos servidors en dos terminals diferents, sense fer servir cap contenidor. Què és el que succeeix en intentar fer-ho? Per què succeeix?

No ens permet executar els dos servidors perquè el port 5000 ja està ocupat, aleshores no el podem utilitzar. L'error es mostra a continuació:

```
oslab:~/Desktop/experiments/socket-wo-docker> ./socker_server
bind() ha fallat. Prova un altre port.
```

Pregunta: Observeu, al README, que per fer el mapat de ports es fa servir l'opció "-p" per executar el contenidor. Descriviu breument, fent servir la documentació oficial del Docker, què és el que permet fer l'opció "-p". Indiqueu també què passa si no es fa servir l'opció "-p" per executar el contenidor. Quina és la pàgina web del Docker on està descrit el funcionament d'aquesta opció?

El que ens permet fer l'opció *-p* és assignar-li un port extern al contenidor i mapejar-lo a un port intern al contenidor. Si no fem servir l'opció -p ambdós servidors intentant escoltar al port 5000 (està definit al codi) del docker host, el que no es pot fer-ho.
Pàgina web: https://docs.docker.com/config/containers/container-networking

Pregunta: Per què, actualment, es fan servir els contenidors per executar els serveis associats a una aplicació més gran en contenidors diferents? Quines avantatges aporta?

Es fan servir contenidors perquè garanteixen l'aïllament entre els serveis, proporcionant millor estabilitat, seguretat i assignació dels recursos.

3. Exercicis: pautes per al desenvolupament d'aplicacions

Secció - Etapa de desenvolupament: compilació i execució en un contenidor

Exercici: Crear un contenidor que tingui el compilador (gcc). En executar el contenidor es farà un “bind mount” de forma que internament hi ha un director /home/appuser/practica4 des del qual es pot accedir a un directori extern al Docker el qual contindrà el codi font i les dades de la pràctica 4. Provar de compilar i executar el codi des de l’interior del contenidor. Comproveu que els executables així com les dades associades que es llegeixen i s’escriuen en executar les aplicacions són fora del contenidor.

Per crear el contenidor hem utilitzat la comanda: **docker build -t exercici1 .**

Un cop tenim el contenidor creat, executem la següent comanda: **docker run --volume /media/sf_shareVM/practica4:/home/appuser/practica4 -ti exercici1**

No es va poder compilar el codi perquè el contenidor no tenia permís de lectura i escriptura en la carpeta.

Pregunta: Observar que per poder escriure al directori extern al Docker cal permisos per poder-ho fer. Com ho solucioneu?

Per donar permisos al docker hem executat la següent comanda dins del host: **chmod -R a+rwx practica4/**

Secció - Etapa de producció: execució en un contenidor petit

Exercici: L’objectiu és utilitzar una imatge que contingui el mínim necessari per poder executar l’aplicació (accedint a les dades externes amb el “bind mount”). Quin és el Dockerfile corresponent? Com heu arribat a trobar-lo?

El Dockerfile corresponent és el següent: **FROM** frolvlad/alpine-glibc

Aquesta imatge està disponible en https://hub.docker.com/r/frolvlad/alpine-glibc/ i és basat en Alpine Linux Image.

El codi va ser compilat de manera que requereix *dynamic libraries* per executar. Aleshores necessitem d'un contenidor que tingui les *dynamic libraries* necessàries per executar el binari. Amb la imatge Alpine només no es pot executar el binario perquè Alpine és una distribució Linux basat en musl libc.

Un altra solució utilitzant la imatge de Alpine seria construir el binari dins d'un container Alpine i executar-ho en un altre contenidor Alpine. D'aquest manera tindrem un contenidor més petit, vist que Alpine és una distribució de Linux molt compacta (5Mb).

Secció - Etapa de producció: execució en un contenidor fent servir volums

Pregunta: Què és el que fan cadascuna de les instruccions anteriors?

La comanda *docker volume create vol-practica4* crea un volum.
La comanda *docker volume ls* lista los volumes coneguts al Docker.
La comanda *docker container create* crea un nou contenidor.
La comanda *docker cp* copia el contingut del primer paràmetre al segon paràmetre.
La comanda *docker rm* esborra el container.

Aquest conjunt d'instruccions el que fa és utilitzar un contenidor temporal per copiar el contingut al volum.

Exercici: Comprovem que s'ha copiat tot correctament. Per això es demana executar el contenidor petit muntant el volum vol-practica4 en el directori /home/appuser/practica4. Què hi ha en aquest directori? Quin és l'usuari propietari dels arxius? Es pot executar el codi que hi ha?

En el directori hi hauriem de trobar els arxius copiats amb la comanda *docker cp* que son els que tenim a la carpeta practica4 . L'usuari propietari es appuser. I si que es pot executar el codi.

El dockerfile utilitzat en aquest exercici es el mateix que al primer exercici.

4. Conclusions

Hem pogut acabar la pràctica correctament i fent funcionar tot el que se'ns ha demanat.

Hem après com funciona el docker i algunes de les seves aplicacions. Primer experimentant amb els contenidors i finalment posan en pràctica el desenvolupament de aplicacions.