UNIVERSITAT DE BARCELONA

Sistemes operatius II

Informe pràctica 2

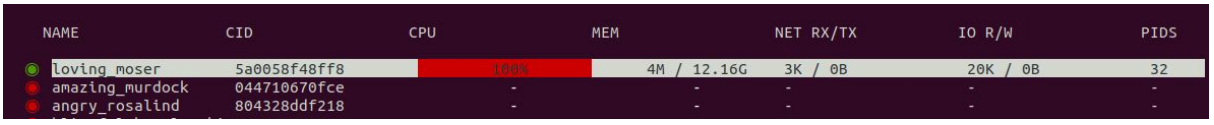
Arnau Gris
Eric Duque

Q1) Com es pot executar un contenidor que ha sigut “aturat” prèviament?

- docker start -a <id del contenidor>

Q2) Quants fork-bomb està executant dins del contenidor? Com ho compteu? Podeu fer servir instruccions de la línia de comandes per saber-ho?

- 32 fork bomb perquè està utilitzant 32 PIDS, ho indica al ctop.



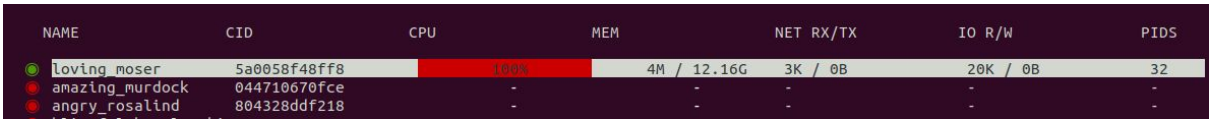
NAME	CID	CPU	MEM	NET RX/TX	IO R/W	PIDS
loving_moser	5a0058f48ff8	100%	4M / 12.16G	3K / 0B	20K / 0B	32
amazing_murdock	044710670fce	-	-	-	-	-
angry_rosalind	804328ddf218	-	-	-	-	-

- Amb la comanda “top” ens mostra el PID de cada fork bomb per separat.

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
10919	500	20	0	2140	68	0	R	4,3	0,0	0:03.42	fork-bomb
10920	500	20	0	2140	68	0	R	4,3	0,0	0:02.17	fork-bomb
10925	500	20	0	2140	68	0	R	4,3	0,0	0:02.36	fork-bomb
10927	500	20	0	2140	68	0	R	4,3	0,0	0:02.32	fork-bomb
10930	500	20	0	2140	68	0	R	4,3	0,0	0:02.14	fork-bomb
10943	500	20	0	2140	68	0	R	4,3	0,0	0:01.74	fork-bomb
10944	500	20	0	2140	68	0	R	4,3	0,0	0:06.23	fork-bomb
10945	500	20	0	2140	68	0	R	4,3	0,0	0:01.89	fork-bomb
10924	500	20	0	2140	68	0	R	4,0	0,0	0:01.82	fork-bomb
10929	500	20	0	2140	68	0	R	4,0	0,0	0:02.14	fork-bomb
10934	500	20	0	2140	68	0	R	4,0	0,0	0:02.50	fork-bomb
10872	500	20	0	2140	732	672	R	3,7	0,0	0:05.72	fork-bomb
10926	500	20	0	2140	68	0	R	3,3	0,0	0:02.21	fork-bomb
10933	500	20	0	2140	68	0	R	3,3	0,0	0:02.16	fork-bomb
10937	500	20	0	2140	68	0	R	3,3	0,0	0:02.03	fork-bomb
10932	500	20	0	2140	68	0	R	3,0	0,0	0:02.17	fork-bomb
10939	500	20	0	2140	68	0	R	3,0	0,0	0:02.03	fork-bomb
10947	500	20	0	2140	68	0	R	3,0	0,0	0:02.16	fork-bomb
10949	500	20	0	2140	68	0	R	3,0	0,0	0:01.88	fork-bomb
2027	oslab	20	0	4864388	414532	179384	S	2,7	3,3	1:39.21	gnome-shel
10923	500	20	0	2140	68	0	R	2,7	0,0	0:02.11	fork-bomb
10922	500	20	0	2140	68	0	R	2,3	0,0	0:03.17	fork-bomb
10931	500	20	0	2140	68	0	R	2,3	0,0	0:01.82	fork-bomb
10948	500	20	0	2140	68	0	R	2,3	0,0	0:01.79	fork-bomb
10921	500	20	0	2140	68	0	R	2,0	0,0	0:02.24	fork-bomb
10928	500	20	0	2140	68	0	R	2,0	0,0	0:01.75	fork-bomb
10935	500	20	0	2140	68	0	R	2,0	0,0	0:02.10	fork-bomb
10936	500	20	0	2140	68	0	R	2,0	0,0	0:01.76	fork-bomb
10938	500	20	0	2140	68	0	R	2,0	0,0	0:01.64	fork-bomb
10940	500	20	0	2140	68	0	R	2,0	0,0	0:03.32	fork-bomb
10941	500	20	0	2140	68	0	R	2,0	0,0	0:03.32	fork-bomb
10942	500	20	0	2140	68	0	R	2,0	0,0	0:01.78	fork-bomb
10946	500	20	0	2140	68	0	R	2,0	0,0	0:01.92	fork-bomb

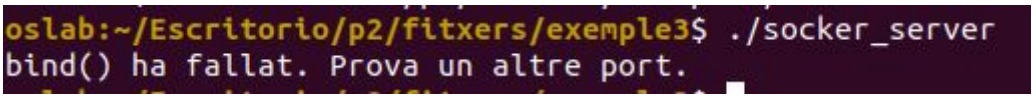
Q3) Quanta CPU està utilitzant el contenidor? Està utilitzant 1, 2 o més CPUs? Per mirar-ho tingueu en compte que en un ordinador amb 4 CPUs el màxim de CPU que es pot ocupar és d'un 400%.

- 100% del que te assignat.
- Está utilitzant només 1 CPU



Q4) Proveu d'executar els dos servidors des de dos terminals diferents, sense fer servir cap contenidor. Què és el que succeeix en intentar fer-ho? Per què succeeix?

- Que ens diu que el port 5000 està en ús, i per tant no el podem fer servir.



Q5) 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 4 el contenidor.

- L'opció -p permet assignar un port específic del host i mapejar-lo a un altre del contenidor.

Q6) Quina és la pàgina web del Docker on està descrit el funcionament d'aquesta opció?

- Documentació oficial

Q7) Per què, actualment, es fan servir els contenidors per executar els serveis associats a una aplicació més gran en contenidors diferents? Quines avantatges aporta?

- Perquè aporta el benefici de independencia entre serveis, de manera que cada servei està aïllat amb els altres. Permet crear entorns predicibles i és menys probable que generin errors que comprometin tota l'aplicació. Els contenidors poden incloure dependències de software que els serveis necessiten, com biblioteques. En estar aïllades garanteixen un entorn d'execució independent, coherent i segur.

- Els contenidors permeten empaquetar les aplicacions i les seves dependències en un arxiu de manifest petit on les seves versions es poden controlar. D'aquesta forma les aplicacions poden ser replicades fàcilment entre els desenvolupadors de l'equip.

- L'entorn on els desenvolupadors han de treballar es converteix en quelcom més petit, el que comporta major agilitat i productivitat. Tot això facilita el desenvolupament, les proves, el desplegament i l'administració dels serveis i aplicacions.

- Beneficis dels contenidors

Exercici

1) Per construir el contenidor fem servir la comanda:

```
docker build -t exemple1
```

2) Per copiar el codi font abans de construir el contenidor incluim al dockerfile:

```
COPY --chown=appuser:appgroup codi/* /home/appuser/
```

Aquesta comanda copia el contingut de la carpeta local "codi" a dins del contenidor específicament al directori "/home/appuser/"

3) Per compartir la carpeta fem servir la següent comanda:

```
docker run -v /home/oslab/Escritorio/p2/fitxers/exercici/dades/:/home/appuser/shared/:rw -it exemple1
```

Aquesta comanda permet crear un accés a la carpeta "dades" dins del contenidor, aquesta carpeta dins del contenidor s'anomena "shared" i tot el que es modifiqui en aquesta es guardarà a la carpeta "dades", això es pot fer gràcies al fet que li donem els permisos r i w (read i write).
Compartir data entre host i contenidor

4) Per guardar la sortida de statistics al host farem servir la següent comanda:

```
./statistics > shared/output.txt
```

La sortida del programa es guardarà a la carpeta "dades".