

SPRINT 2

Índex

Resum Sprint 1-----	3
Docker -----	3
Servidor Web en Docker -----	7
MongoDB en Docker -----	18
Servidor DNS amb Docker -----	28
Servidor DHCP en Docker -----	32
Cloud Computing -----	37
Fog -----	46
Burndown-----	49

Resum Sprint 1

L'Sprint anterior vam començar a implementar els serveis necessaris en màquines virtuals i es va veure un altre esquema de les comunicacions, tot i que es va veure un simplificat.

En fase el feedback del sprint anterior, en aquest Sprint s'han dockeritzat els serveis que havíem fet en VirtualBox. També hem proposar un esquema força més millorat i detallat de com serà la capa de cloud i fog. També s'ha començat a deixar preparar el cloud per donar accés als altres equips.

Docker

Comandes bàsiques del docker

1- Comandes per la gestió de contenidors:

- Construïm una imatge

`docker build`

Si no s'especifiquen paràmetres, buscarà un arxiu dockerfile i crearà la imatge a partir de les configuracions dintre d'aquest. Algunes de les formes típiques per crear el contenidor:

`docker build --build-arg -t <nombre_imagen> .`

`<nombre_imagen>`: nom de la imatge que posarem

`<.>`: directori on es crearà la imatge

- Executar un contenidor

`docker run -v <unidad_host>:<unidad_docker> -it --name <nombre_contenedor> -d <nombre_imagen>`

Amb aquesta comanda junt amb la de build, el nostre contenidor estarà en funcionament. També es pot fer anar una imatge amb un simple script de configuració que tenen algunes imatges.

- Llistar imatges en el nostre servidor:

`docker image ls`

- Contenidors actius, per veure'ls tots amb l'opció -a

`docker ps`

- Esborrar contenidors:

`docker rm -f <container ID> <container-name>`

- Esborrar imatges:

`docker image rm -f <container ID>`

- Esborrem contenidor

`docker rm -f <container ID>`

- Copia d'arxius

- Docker -> host
docker cp [nombre-contenedor]:[ruta/archivo.txt] .
- Host -> docker
docker cp [ruta/archivo.txt] [nombre-contenedor]:[ruta/]

2- Comandes de docker:

```
docker start/stop/kill <nombre_imagen>
```

Què és dockerfile

És un arxiu de text pla que conté una sèrie d'instruccions necessàries per crear una imatge de docker. Quan s'inicia una imatge al docker, si no s'indica el contrari, docker buscarà aquest fitxer per crear-la amb els paràmetres indicats.

Un fitxer dockerfile té les següents parts:

```
#Descarregar la imatge, indiquem la imatge base sobre la qual es crearà la imatge
FROM <imagen>
FROM <imagen>:<tag>
#Copiar fitxer de configuració
COPY <ruta>
#Definir la ruta per defecte del projecte
WORKDIR <ruta>
#Executar comandes només a l'inici del contenidor.
CMD ["ejecutable", "parámetro1", "parámetro2"]
```

També es poden indicar altres coses com:

- RUN(executa comanda en el contenidor)
- ENV(defineix variables d'entorn en el nostre contenidor)
- ADD(copiar arxius a un destí dintre del nostre contenidor)
- VOLUME(punt de muntatge dintre del contenidor, visible des de l'amfitrió)
- USER(determina l'usuari a utilitzar per executar totes les comandes)

Aquest són alguns dels paràmetres que es poden utilitzar en un dockerfile.

Per veure les imatges oficials es pot entrar a:

https://hub.docker.com/search?image_filter=official&type=image

Instal·lació de Docker

Per fer la instal·lació és recomanable estar logejat com a root amb la comanda ‘su root’ o escrivint ‘sudo’ davant de cada comanda

Primer de tot actualitzem els paquets del sistema amb ‘apt update’

```
root@seax:~# apt update
Obj:1 http://deb.debian.org/debian buster InRelease
Des:2 http://security.debian.org/debian-security buster/updates InRelease [65,4 kB]
Des:3 http://deb.debian.org/debian buster-updates InRelease [49,3 kB]
Des:4 http://security.debian.org/debian-security buster/updates/main Sources [108 kB]
Des:5 http://security.debian.org/debian-security buster/updates/main amd64 Packages [187 kB]
Des:6 http://security.debian.org/debian-security buster/updates/main Translation-en [100 kB]
Descargados 510 kB en 6s (84,6 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se puede actualizar 1 paquete. Ejecute «apt list --upgradable» para verlo.
root@seax:~#
```

Després s'instal·laran una sèrie de paquets que permet treballar amb HTTPS amb la comanda ‘apt install apt-transport-https ca-certificates curl gnupg2 software-properties-common’

```

root@seax:~# apt install apt-transport-https ca-certificates curl gnupg2 software-properties-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  dbus gir1.2-glib-2.0 gir1.2-packagekitglib-1.0 iso-codes krb5-locales libappstream4
  libcurl3-gnutls libcurl4 libdbus-1-3 libgirepository-1.0-1 libglib2.0-0 libglib2.0-bin
  libglib2.0-data libgssapi-krb5-2 libgstreamer1.0-0 libicu63 libk5crypto3 libkeyutils1 libkrb5-3
  libkrb5support0 libnnghttp2-14 libpackagekit-glib2-18 libpam-systemd libpolkit-agent-1-0
  libpolkit-backend-1-0 libpolkit-gobject-1-0 libpsl1 librtmp1 libssh2-1 libstemmer0d libxml2
  libyaml-0-2 openssl packagekit packagekit-tools policykit-1 publicsuffix python-apt-common
  python3-apt python3-dbus python3-distro-info python3-gi python3-pycurl
  python3-software-properties shared-mime-info unattended-upgrades xdg-user-dirs
Paquetes sugeridos:
  default-dbus-session-bus | dbus-session-bus isoquery krb5-doc krb5-user gstreamer1.0-tools
  appstream python3-apt-dbg python-apt-doc python-dbus-doc python3-dbus-dbg libcurl4-gnutls-dev
  python-pycurl-doc python3-pycurl-dbg bsd-mailx default-mta | mail-transport-agent needrestart
  powermgmt-base
Se instalarán los siguientes paquetes NUEVOS:
  apt-transport-https ca-certificates curl dbus gir1.2-glib-2.0 gir1.2-packagekitglib-1.0 gnupg2
  iso-codes krb5-locales libappstream4 libcurl3-gnutls libcurl4 libdbus-1-3 libgirepository-1.0-1
  libglib2.0-0 libglib2.0-bin libglib2.0-data libgssapi-krb5-2 libgstreamer1.0-0 libicu63
  libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 libnnghttp2-14 libpackagekit-glib2-18
  libpam-systemd libpolkit-agent-1-0 libpolkit-backend-1-0 libpolkit-gobject-1-0 libpsl1 librtmp1
  libssh2-1 libstemmer0d libxml2 libyaml-0-2 openssl packagekit packagekit-tools policykit-1
  publicsuffix python-apt-common python3-apt python3-dbus python3-distro-info python3-gi
  python3-pycurl python3-software-properties shared-mime-info software-properties-common
  unattended-upgrades xdg-user-dirs
0 actualizados, 52 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
Se necesita descargar 23,8 MB de archivos.
Se utilizarán 96,7 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s_

```

Després obtindrem una clau de Docker per poder accedir al repositori amb:

'curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -'

```

root@seax:~# curl -fsSL https://download.docker.com/linux/debian/gpg | apt-key add -
OK
root@seax:~

```

A continuació afegim el repositori de Docker amb:

'add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian
\$(lsb_release -cs) stable"'

Tornem a fer un update actualitzar els recursos

```

root@seax:~# add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"
root@seax:~# apt update
Obj:1 http://deb.debian.org/debian buster InRelease
Obj:2 http://deb.debian.org/debian buster-updates InRelease
Ign:3 https://download.docker.com/linux/debian bionic InRelease
Des:4 https://download.docker.com/linux/debian buster InRelease [44,4 kB]
Err:5 https://download.docker.com/linux/debian bionic Release

```

Ara ja es pot instal·lar Docker al sistema correctament i podem fer

' apt install docker-ce' i només caldrà esperar

Si la instal·lació s'ha fet correctament usant la comanda

'systemctl status docker' veurem que el servei està funcionant

```
root@seax:~# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2020-04-16 10:16:00 CEST; 48s ago
       Docs: https://docs.docker.com
      Main PID: 365 (dockerd)
        Tasks: 8
       Memory: 130.1M
      CGroup: /system.slice/docker.service
              └─365 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

abr 16 10:15:59 seax dockerd[365]: time="2020-04-16T10:15:59.033076704+02:00" level=warning msg="You
abr 16 10:15:59 seax dockerd[365]: time="2020-04-16T10:15:59.037333706+02:00" level=warning msg="You
abr 16 10:15:59 seax dockerd[365]: time="2020-04-16T10:15:59.037612058+02:00" level=warning msg="You
abr 16 10:15:59 seax dockerd[365]: time="2020-04-16T10:15:59.037913611+02:00" level=info msg="Loadin
abr 16 10:15:59 seax dockerd[365]: time="2020-04-16T10:15:59.779004492+02:00" level=info msg="Defaul
abr 16 10:15:59 seax dockerd[365]: time="2020-04-16T10:15:59.845184243+02:00" level=info msg="Loadin
abr 16 10:16:00 seax dockerd[365]: time="2020-04-16T10:16:00.236442515+02:00" level=info msg="Docker
abr 16 10:16:00 seax dockerd[365]: time="2020-04-16T10:16:00.236993816+02:00" level=info msg="Daemon
abr 16 10:16:00 seax systemd[1]: Started Docker Application Container Engine.
abr 16 10:16:00 seax dockerd[365]: time="2020-04-16T10:16:00.371196014+02:00" level=info msg="API li
lines 1-20/20 (END)
```

Servidor Web (Apache i PHP) a Docker

Instal·lació d'una imatge d'apache en docker

1- Creem l'esquelet de directoris per tenir tota la instal·lació en un mateix lloc.

```
root@seax:/# mkdir /docker
root@seax:/# mkdir docker/apache
root@seax:/#
```

2-Fer la instal·lació d'apache a través de docker.

El que fa és construir i arrancar el servei sobre docker. En aquest cas no utilitzarem un dockerfile, per fer les primeres proves, utilitzarem els scripts per defecte que ens aporta aquesta imatge, tot es crearà en la carpeta on estem executant la comanda.

```
root@seax:/docker/apache# docker run -dit --name my-apache-app -p 8080:80 -v "$PWD":/usr/local/apache2/htdocs/ httpd:2.4
Unable to find image 'httpd:2.4' locally
2.4: Pulling from library/httpd
c499e6d256d6: Pull complete
76155f771be0: Pull complete
48b718b71719: Pull complete
d65ae7a4c211: Pull complete
8d17dee838ad: Pull complete
Digest: sha256:13aa010584cb3d79d66adf52444494ae5db67faa28d65a1a25e6ddc57f7c0e2a
Status: Downloaded newer image for httpd:2.4
71f4c77ffc36b983d2695cb4547adc7e737f57f7e1d32f9084cf8304de449fb
root@seax:/docker/apache#
```

Comprovem que la nostra imatge s'està executant:

```
root@seax:/docker/apache# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
71f4c77ffc36        httpd:2.4          "httpd-foreground"   20 minutes ago   Up 20 minutes      0.0.0.0:8080->80/tcp   my-apache-app
```

Comprovacions

Es pot comprovar que el servei funciona amb un navegador web accedint pel port pel qual hem executat el servei en la comanda del run.



← → C ⌂ ⓘ No es seguro | 192.168.1.52:8080

Aplicaciones Campus Virtual UP...

It works!

Configuració de apache

De la mateixa forma que ens descarreguem la imatge, si volem configurar-la al nostre gust, també hem de baixar-nos tots els fitxers de configuració pertinents, ara ens descarreguem el fitxer principal de configuració d'apache:

```
root@seax:/home/entel# docker run --rm httpd:2.4 cat /usr/local/apache2/conf/httpd.conf > my-httpd.conf
root@seax:/home/entel#
```

```

GNU nano 3.2                                my-htpd.conf

#
# This is the main Apache HTTP server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
#
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path. If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so "logs/access_log"
# with ServerRoot set to "/usr/local/apache2" will be interpreted by the
# server as "/usr/local/apache2/logs/access_log", whereas "/logs/access_log"
# will be interpreted as '/logs/access_log'.
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used. If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/usr/local/apache2"

```

En aquest fitxer podrem fer totes les modificacions que faríem amb el fitxer normal d'apache. Ara mateix no entraré més en detall amb la configuració d'apache, ja que aquest servei ara no està vinculat amb php i després haurem de descarregar-nos una altra imatge que incorporarà aquest 2 serveis junts, però com a primera toma de contacte amb el docker i provar les diferents comandes, s'ha provat apache.

Després de cada canvi en el fitxer, s'ha de fer un restart del servei.

```

root@seax:/docker/apache# docker restart my-apache-app
my-apache-app

```

Comprovacions

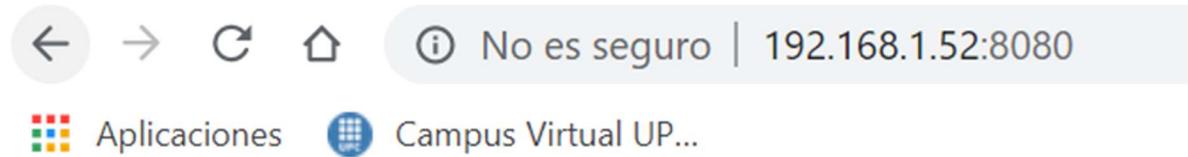
Al directori on hem creat la imatge, creem un index.html per provar com ficar la pàgina web dintre del servidor.

```

GNU nano 3.2                                index.html

<html>
PROVA DE FUNCIONAMET DEL GRUP A2
</html>

```



PROVA DE FUNCIONAMET DEL GRUP A2

Instal·lació d'una imatge conjunta d'apache i php en docker

1- Creem els directoris on crearem el contenidor del docker i la carpeta on serà l'arrel per guardar la pàgina web si no s'especifica el contrari després en la configuració.

/webphp → directori principal del contenidor docker
/webphp/web on anirà la pàgina web

```
root@seax:/# mkdir webphp
root@seax:/# mkdir webphp/web
```

2- Creem un fitxer docker a partir del qual s'iniciarà la nostra imatge en el contenidor docker que crearem. Ha d'estar en el directori on es farà el build del nostre contenidor.

```
GNU nano 3.2                                     webphp/Dockerfile

FROM php:7.0-apache
COPY web /var/www/html
EXPOSE 80
```

També crearem un fitxer index.php per després fer les proves de connectivitat:

```
GNU nano 3.2                                     webphp/web/index.php

<?php
echo "Hola Mundo"
?>
```

3- Construïm el contenidor, totes les dades les extreu del fitxer Dockerfile, es pot observar com hi han 3 passos.

- 1- Obtenir les imatges necessàries
- 2- Copiem el que hi ha en la nostra carpeta web dintre de la imatge nova en la carpeta "/var/www/html"
- 3- Compartim el port 80

```
root@seax:/# cd webphp/
root@seax:/webphp# docker build -t imagen_web_php .
Sending build context to Docker daemon 3.584kB
Step 1/3 : FROM php:7.0-apache
7.0-apache: Pulling from library/php
177e7ef0df69: Pull complete
9bf89f2eda24: Pull complete
350207dcf1b7: Pull complete
a8a33d96b4e7: Pull complete
c0421d5b63d6: Pull complete
f76e300fbe72: Pull complete
af9ff1b9ce5b: Pull complete
d9f072d61771: Pull complete
37007e292198: Pull complete
8ba923990f24: Pull complete
98af8902979a: Pull complete
f1548c2cd376: Pull complete
e1062fd0605a: Pull complete
Digest: sha256:1d34b2e491a02ba7a8d26478132015e197a5ffea37f0a93b42621d11cf042cc
Status: Downloaded newer image for php:7.0-apache
--> aa67a9c9814f
Step 2/3 : COPY web /var/www/html
--> 4fc870f5d704
Step 3/3 : EXPOSE 80
--> Running in 46276eff75d5
Removing intermediate container 46276eff75d5
--> dd06d897c0c6
Successfully built dd06d897c0c6
Successfully tagged imagen_web_php:latest
root@seax:/webphp#
```

4- Veure les imatges que s'han descarregat per tenir el servidor web

```
root@seax:/webphp# docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
imagen_web_php     latest   932524b82593  2 minutes ago  368MB
my-php-app          latest   b42b9448a962  30 minutes ago  405MB
php                 7.4-cli  76a5dae0c884  2 weeks ago   405MB
httpd               2.4     8326be82abe6  2 weeks ago   166MB
php                 7.0-apache  aa67a9c9814f  15 months ago  368MB
root@seax:/webphp#
```

5- Ara arranquem el servidor de forma interactiva per veure els logs i compartim el port 80 de la màquina amfitrió amb el contenidor. D'aquesta forma al connectar-nos directament amb la ip de la màquina real, es redirigirà cap al port 80 del contenidor docker i encara que el docker estigui en una xarxa pròpia d'aquest servei, podrem connectar-nos al servidor apache.

```

root@seax:/webphp# docker run -it -p 80:80 imagen_web_php
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[Wed Apr 15 15:23:35.494227 2020] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.25 (Debian) PHP/7.0.33 configured -- resuming normal operations
[Wed Apr 15 15:23:35.495695 2020] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
192.168.1.42 - - [15/Apr/2020:15:26:03 +0000] "GET / HTTP/1.1" 200 240 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.92 Safari/537.36"
192.168.1.42 - - [15/Apr/2020:15:26:03 +0000] "GET /favicon.ico HTTP/1.1" 404 503 "http://192.168.1.52/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.92 Safari/537.36"

```

Comprovacions

← → ⏪ ⌂ ⓘ No es seguro | 192.168.1.52

Aplicaciones Campus Virtual UP...

Hola mundo

Arrancar/parar el servei

Un cop tot funciona, ja podem arrancar i parar el servei:

1- Mirar la informació de la imatge

```

root@seax:/webphp# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
6a3622f29936        imagen_web_php   "docker-php-entrypoi..."   19 minutes ago    Exited (0) 8 minutes ago           stupefied_ptolemy
root@seax:/webphp#

```

2- Fer el start/stop

```

root@seax:/webphp# docker start 6a3622f29936
6a3622f29936

```

3- Comprovar els processos actius:

```

root@seax:/webphp# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
6a3622f29936        imagen_web_php   "docker-php-entrypoi..."   25 minutes ago    Up About a minute   0.0.0.0:80->80/tcp   stupefied_ptolemy
root@seax:/webphp#

```

Configuració i com funciona el servei

Cada cop que es tingui una versió estable de la pàgina web s'ha d'executar el següent script per passar els arxius locals a la imatge local. Això ho fem forçant a esborrar i tornar a crear la imatge.

Dintre del directori /webphp/

```
GNU nano 3.2                                         script_web.sh

#!/bin/bash
ID_DOCKER=$(docker ps -a | grep imagen_web_php | awk '{print $1}')
docker stop $ID_DOCKER
docker rm $ID_DOCKER
docker rmi imagen_web_php --force
docker build -t imagen_web_php .
docker run -it -p 80:80 imagen_web_php
exit 0
```

Donem permisos al script:

```
root@seax:/# chmod +x webphp/script_web.sh
```

Comprovacions

Fem modificacions al fitxer index.php

```
GNU nano 3.2                                         web/index.php

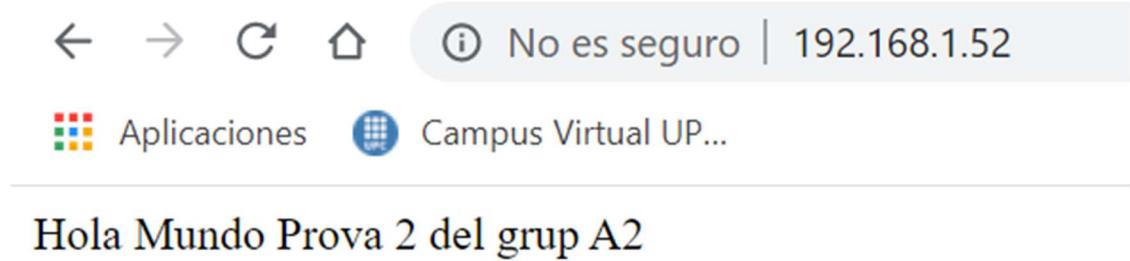
<?php
echo "Hola Mundo \n";
echo "Prova 2 del grup A2";
?>
```

Executem el script

```
root@seax:/webphp# ./script_web.sh
6a3622f29936
Untagged: imagen_web_php:latest
Deleted: sha256:2c34505b5069c6076951cc26c0ccle5e9cc50096eb468c20bc94285f6334d964
Deleted: sha256:163c4edc925c44563d93a4733ba0d951d666b1647e9fe5e08accccd6c3d0314c
Sending build context to Docker daemon 4.608kB
Step 1/3 : FROM php:7.0-apache
--> aa6749c9814f
Step 2/3 : COPY web /var/www/html
--> 041fce698a24
Step 3/3 : EXPOSE 80
--> Running in 2a89dafddd4
Removing intermediate container 42a09dafddd4
--> b78637a26462
Successfully built b78637a26462
Successfully tagged imagen_web_php:latest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[Wed Apr 15 16:39:09.418232 2020] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.25 (Debian) PHP/7.0.33 configured -- resuming normal operations
[Wed Apr 15 16:39:09.418716 2020] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
```

Ara comprovem si s'han guardat els canvis tornant a fer la connexió:

Comprovacions:



The screenshot shows a web browser window with the following details:

- Address bar: No es seguro | 192.168.1.52
- Icons: back, forward, refresh, home.
- Links: Aplicaciones, Campus Virtual UP...
- Content: Hola Mundo Prova 2 del grup A2

Configurar apache i php

La configuració del servei apache es va explicar en la passada documentació. L'única diferència és que ara no tenim els arxius en la nostra màquina, ara tenim tota l'estructura de fitxers en la imatge. Per fer-nos una idea la imatge té la següent estructura:

Nombre	Tamaño	Comprimido	Tipo	Modificado	CRC32
..			Carpeta de archivos		
bin			Carpeta de archivos	29/12/2018 5:32	
boot			Carpeta de archivos	20/10/2018 12:40	
dev			Carpeta de archivos	15/04/2020 19:03	
etc			Carpeta de archivos	15/04/2020 19:03	
home			Carpeta de archivos	20/10/2018 12:40	
lib			Carpeta de archivos	29/12/2018 5:23	
lib64			Carpeta de archivos	26/12/2018 1:00	
media			Carpeta de archivos	26/12/2018 1:00	
mnt			Carpeta de archivos	26/12/2018 1:00	
opt			Carpeta de archivos	26/12/2018 1:00	
proc			Carpeta de archivos	20/10/2018 12:40	
root			Carpeta de archivos	29/12/2018 7:51	
run			Carpeta de archivos	29/12/2018 5:32	
sbin			Carpeta de archivos	29/12/2018 5:32	
srv			Carpeta de archivos	26/12/2018 1:00	
sys			Carpeta de archivos	20/10/2018 12:40	
tmp			Carpeta de archivos	29/12/2018 7:51	
usr			Carpeta de archivos	26/12/2018 1:00	
var			Carpeta de archivos	29/12/2018 5:32	
.dockerenv	0	0	Archivo DOCKERENV	15/04/2020 19:03	

I les carpetes que ens interessen són les mateixes que en el servei normal d'apache:

/etc/apache2

Nombre	Tamaño	Comprimido	Tipo	Modificado	CRC32
[..]			Carpeta de archivos		
conf-available			Carpeta de archivos	29/12/2018 5:32	
conf-enabled			Carpeta de archivos	29/12/2018 5:32	
mods-available			Carpeta de archivos	29/12/2018 7:51	
mods-enabled			Carpeta de archivos	29/12/2018 7:51	
sites-available			Carpeta de archivos	29/12/2018 5:32	
sites-enabled			Carpeta de archivos	29/12/2018 5:32	
apache2.conf	7.224	7.224	Archivo CONF	03/11/2018 19:46	
envvars	1.924	1.924	Archivo	29/12/2018 5:32	
magic	31.063	31.063	Archivo	03/11/2018 12:34	
ports.conf	320	320	Archivo CONF	03/11/2018 12:34	

/var/www/html

Nombre	Tamaño	Comprimido	Tipo	Modificado	CRC32
[..]			Carpeta de archivos		
index.php	59	59	Archivo PHP	15/04/2020 18:38	

Per fer qualsevol canvi, ens baixem l'arxiu que volem amb les comandes mencionades al començament, fer els canvis, pujar-les a la imatge i recarregar la imatge perquè es vegin els canvis reflectits. Les configuracions s'haurien de tractar com fem per pujar contingut nou a la imatge.

Exportar/importar el contenidor docker

Per exportar un servei:

- 1- Mirem el id del contenidor a exportar
- 2- Exportem el contenidor a un .tar

```
root@seax:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
              PORTS     NAMES
799f777e47c2        imagen_web_php   "docker-php-entrypoi..."   39 hours ago       Exited (0) 39 h
ago
root@seax:~# docker export 799f777e47c2 > apache-php.tar
root@seax:~# ls
apache-php.tar
root@seax:~#
```

- 3- Un cop tenim el .tar, ara hem de pasar-la a una altra màquina utilitzant un USB,ssh o una carpeta compartida o el que ens sigui més còmode.

A la màquina on importarem el servei:

1- Creem l'estructura de directoris:

```
root@seax:~# mkdir webphp
root@seax:~# mkdir webphp/web
root@seax:~# cd webphp/
root@seax:~/webphp#
```

2- Obtenim el tar

3- Ara importem la imatge

```
root@seax:~/webphp# cat /root/apache-php.tar | docker import - imagen_web_php
sha256:96ccce2f4bb6732b2216f365aaaf5b0f4fba8d89f93f37f7f919b4780e3f72a92
root@seax:~/webphp# _
```

4- Comprovem que la imatge s'ha importat bé:

```
root@seax:~/webphp# docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
imagen_web_php     latest   96ccce2f4bb6   4 minutes ago  361MB
```

5- Ara ens falta obtenir el Dockerfile, el script i la nostra pàgina web per acabar de fer que tot funcioni correctament:

```
root@seax:~/webphp# ls
Dockerfile  script_web.sh  web
root@seax:~/webphp# ls web/
index.php
root@seax:~/webphp# _
```

6- Construïm el contingut

```
root@seax:~/webphp# docker build -t imagen_web_php .
Sending build context to Docker daemon 4.608kB
Step 1/3 : FROM php:7.0-apache
7.0-apache: Pulling from library/php
177e7ef0df69: Pull complete
9bf89f2eda24: Pull complete
350207dcf1b7: Pull complete
a8a33d96b4e7: Pull complete
c0421d5b63d6: Pull complete
f76e300fbe72: Pull complete
af9ff1b9ce5b: Pull complete
d9f072d61771: Pull complete
37007e292198: Pull complete
8ba923990f24: Pull complete
98af8902979a: Pull complete
f1548c2cd376: Pull complete
e1062fd0605a: Pull complete
Digest: sha256:1d34b2e491a02ba7a8d26478132015e197a5ffea37f0a93b42621d11cf042cc
Status: Downloaded newer image for php:7.0-apache
--> aa67a9c9814f
Step 2/3 : COPY web /var/www/html
--> fcd0f4016cfe
Step 3/3 : EXPOSE 80
--> Running in 237b82d2339f
Removing intermediate container 237b82d2339f
--> b962cea6f125
Successfully built b962cea6f125
Successfully tagged imagen_web_php:latest
root@seax:~/webphp# _
```

7- Arranquem el sistema tal com hem fet fins ara:

```
root@seax:~/webphp# docker run -it -p 80:80 imagen_web_php
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[Fri Apr 17 08:38:54.055801 2020] [mpm_prefork:notice]  [pid 1] AH00163: Apache/2.4.25 (Debian) PHP/7.0.33 configured -- resuming normal operations
[Fri Apr 17 08:38:54.057429 2020] [core:notice]  [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
192.168.1.42 - - [17/Apr/2020:08:39:04 +0000] "GET / HTTP/1.1" 200 261 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.113 Safari/537.36"
192.168.1.42 - - [17/Apr/2020:08:39:04 +0000] "GET /favicon.ico HTTP/1.1" 404 503 "http://192.168.1.54/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.113 Safari/537.36"
```

No oblidem que si arranquem així el contingut i tot funciona, és un run interactiu per veure tots els moviments que es generen en el nostre servidor. Si volem arrancar-ho de forma normal, seria amb docker start.

```
root@seax:~/webphp# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
9b6407214d50        imagen_web_php   "docker-php-entrypoi..."   3 minutes ago      Exited (0) About a minute ago
root@seax:~/webphp# docker start 9b64072114d50
Error response from daemon: No such container: 9b64072114d50
Error: failed to start containers: 9b64072114d50
root@seax:~/webphp# docker start 9b6407214d50
9b6407214d50
root@seax:~/webphp# _
```

Comprovacions:

← → C ⌂ ⓘ No es seguro | 192.168.1.54

Aplicaciones Campus Virtual UP...

Hola Mundo Prova 2 del grup A2

A partir d'ara tot funciona com en l'altre servidor.

Base de Dades

MongoDB

Instal·lació

1. Ens baixem la imatge:

```
# docker pull mongo
```

Si no especificuem res, per defecte ens baixarà l'última versió (latest).

2. Comprovem que tenim la imatge:

```
# docker images | grep mongo
```

```
root@seax:~# docker images | grep mongo
mongo           latest            c5e5843d9f5f      2 weeks ago     387MB
```

3. Creem un directori on tindrem el volum:

```
# mkdir mongo-docker
# cd mongo-docker
```

4. Fem una instància de la imatge:

```
# docker run -d -p 2717:27017 -v ~/mongo-docker:/data/db
--name mymongo mongo
```

```
root@seax:~/mongo-docker# docker run -d -p 2717:27017 -v ~/mongo-docker:/data
/db --name mymongo mongo
eed9b8c64f0ac94abdfab78ab038b71f930832c09db6fe0908f45e26fb19c13f
```

Mirem el contingut del directori:

```
root@seax:~/mongo-docker# ls
collection-0-8818174410448091278.wt  journal          WiredTiger
collection-2-8818174410448091278.wt  _mdb_catalog.wt  WiredTigerLAS.wt
collection-4-8818174410448091278.wt  mongod.lock    WiredTiger.lock
index-1-8818174410448091278.wt     sizeStorer.wt   WiredTiger.turtle
index-3-8818174410448091278.wt     storage.bson  WiredTiger.wt
```

5. Fem una instància de MongoDB amb la comanda:

```
# docker run --name some-mongo -d mongo:tag
```

some-mongo → Nom que assignem al contenidor

tag → Especificar la versió de MongoDB que volem

Llavors adecuem els paràmetres i fem:

```
# docker run --name my-mongo-db -d mongo
```

Al no posar cap *tag* obtindrem la versió més recent de MongoDB
Aquest contenidor s'anomenarà *my-mongo-db*.

```
root@seax:~# docker run --name my-mongo-db -d mongo
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
5bed26d33875: Pull complete
f11b29a9c730: Pull complete
930bda195c84: Pull complete
78bf9a5ad49e: Pull complete
3d7fb3809884: Pull complete
a7237292ff8a: Pull complete
c936e28b5159: Pull complete
3fb56b127f30: Pull complete
c54a0478af29: Pull complete
ffc3a4dd1cdc: Pull complete
236ffdb10499: Pull complete
c33afe03b109: Pull complete
1d3609ce2ac9: Pull complete
Digest: sha256:1e38093260855e83baee0237e29947e243818c58a1d37b1022909e227f624d7a
Status: Downloaded newer image for mongo:latest
abbe6037e49d6f11b98cbd10bd011c530ef9af66d2b75a5e90b941a67b86e466
```

Configuració

Accedir al contenidor

1. Per veure els contenidors disponibles:

```
# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	
eed9b8c64f0a	mongo	"docker-entrypoint.s..."	2 minutes ago
Up 2 minutes	0.0.0.0:2717->27017/tcp	mymongo	

2. Fem un start del contenidor:

```
# docker start mymongo
```

```
root@seax:~/mongo-docker# docker start mymongo
mymongo
```

3. Per accedir al shell del contenidor:

```
# docker exec -it some-mongo bash
```

En el nostre cas:

```
# docker exec -it mymongo bash
```

```
root@seax:~/mongo-docker# docker exec -it mymongo bash
root@eed9b8c64f0a:/# 
```

4. Ara des d'aquí dintre podrem accedir a mongo:

```
# mongo
```

```
root@eed9b8c64f0a:/# mongo
MongoDB shell version v4.2.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("96e9bad3-24e0-4184-8212-a54f04c273b8")
" } }
MongoDB server version: 4.2.5
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
    http://docs.mongodb.org/
Questions? Try the support group
    http://groups.google.com/group/mongodb-user
Server has startup warnings:
2020-04-18T10:23:10.446+0000 I STORAGE [initandlisten]
2020-04-18T10:23:10.446+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2020-04-18T10:23:10.446+0000 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten]
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten]
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten]
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten] ** We suggest setting it to 'never'
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).
The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.
To enable free monitoring, run the following command: db.enableFreeMonitoring()
```

Creació d'usuaris i bases de dades

1. Per llistar les bases de dades:

```
# show dbs
```

```
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
```

2. Creem una base de dades per les proves de més endavant:

```
# use test
```

Una vegada inserim un element, la base de dades quedarà guardada.

```
# db.test.insert({ "name": "prova" })
```

```
> use test
switched to db test
> db.test.insert({ "name": "prova" })
WriteResult({ "nInserted" : 1 })
> db.test.find({})
{ "_id" : ObjectId("5e9ad72f42f6592e3048704d"), "name" : "prova" }
```

Comprovem que s'ha creat:

```
# show dbs
```

```
> show dbs
admin 0.000GB
config 0.000GB
local 0.000GB
test 0.000GB
```

3. Creem l'usuari admin:

S'ha de crear a la base de dades 'admin'.

```
> db.createUser({
    user: "admin",
    pwd: "admin",
    roles: [
        { role: "userAdminAnyDatabase", db: "admin" },
        { role: "readWriteAnyDatabase", db: "admin" },
        { role: "dbAdminAnyDatabase", db: "admin" }
    ]
});
```

```

ccbc 0.8.0-SNAPSHOT
> db.createUser({
...     user: "admin",
...     pwd: "admin",
...     roles: [
...         { role: "userAdminAnyDatabase", db: "admin" },
...         { role: "readWriteAnyDatabase", db: "admin" },
...         { role: "dbAdminAnyDatabase", db: "admin" }
...     ]
... });
Successfully added user: {
    "user" : "admin",
    "roles" : [
        {
            "role" : "userAdminAnyDatabase",
            "db" : "admin"
        },
        {
            "role" : "readWriteAnyDatabase",
            "db" : "admin"
        },
        {
            "role" : "dbAdminAnyDatabase",
            "db" : "admin"
        }
    ]
}
>

```

Comprovem que s'ha afegit correctament amb:

```

# show users
> show users
{
    "_id" : "test.admin",
    "userId" : UUID("4cd059d8-7a56-490d-9419-55ca520f95e4"),
    "user" : "admin",
    "db" : "test",
    "roles" : [
        {
            "role" : "userAdminAnyDatabase",
            "db" : "admin"
        },
        {
            "role" : "readWriteAnyDatabase",
            "db" : "admin"
        },
        {
            "role" : "dbAdminAnyDatabase",
            "db" : "admin"
        }
    ],
    "mechanisms" : [
        "SCRAM-SHA-1",
        "SCRAM-SHA-256"
    ]
}
> □

```

4. Creem l'usuari user1:

El crearem a la base de dades 'prova'.

```
> db.createUser({  
    user: "user1",  
    pwd: "user1",  
    roles: [  
        { role: "userAdmin", db: "test" },  
        { role: "dbAdmin", db: "test" },  
        { role: "readWrite", db: "test" }  
    ]  
}) ;
```

```
> db.createUser({  
...     user: "user1",  
...     pwd: "user1",  
...     roles: [  
...         { role: "userAdmin", db: "test" },  
...         { role: "dbAdmin", db: "test" },  
...         { role: "readWrite", db: "test" }  
...     ]  
...});  
Successfully added user: {  
    "user" : "user1",  
    "roles" : [  
        {  
            "role" : "userAdmin",  
            "db" : "test"  
        },  
        {  
            "role" : "dbAdmin",  
            "db" : "test"  
        },  
        {  
            "role" : "readWrite",  
            "db" : "test"  
        }  
    ]  
}  
> 
```

Comprovem que s'ha afegit correctament amb:

```
# show users
```

```
"db" : "test",
"roles" : [
    {
        "role" : "userAdminAnyDatabase",
        "db" : "admin"
    },
    {
        "role" : "readWriteAnyDatabase",
        "db" : "admin"
    },
    {
        "role" : "dbAdminAnyDatabase",
        "db" : "admin"
    }
],
"mechanisms" : [
    "SCRAM-SHA-1",
    "SCRAM-SHA-256"
]
}
{
    "_id" : "test.user1",
    "userId" : UUID("ca0faa72-d31a-47e5-99f3-c35d922b8a22"),
    "user" : "user1",
    "db" : "test",
    "roles" : [
        {
            "role" : "userAdmin",
            "db" : "test"
        },
        {
            "role" : "dbAdmin",
            "db" : "test"
        },
        {
            "role" : "readWrite",
            "db" : "test"
        }
    ],
    "mechanisms" : [
        "SCRAM-SHA-1",
        "SCRAM-SHA-256"
    ]
}
> 
```

Fitxers de configuració

1. Els contenidor de docker porten el mínim instal·lat. Per editar qualsevol arxiu hem d'instal·lar manualment un editor. En aquest, cas, instal·larem vim:

```
# apt-get update  
# apt-get install vim
```

2. Modificar el fitxer /etc/mongod.conf.orig

Per a que els rols que hem assignat funcionin.

```
# vim /etc/mongod.conf orig
```

Hem d'afegir:

```
security:  
    authorization: enabled  
  
#security  
security:  
    authorization: enabled
```

Per defecte mongoDB només accepta connexions de localhost. Hem de permetre acceptar connexions remotes.

En el mateix fitxer d'abans, /etc/mongod.conf, substituïm:

```
# network interfaces  
net:  
    port: 27017  
    bindIp: 127.0.0.1
```

per

```
# network interfaces  
net:  
    port: 27017  
    bindIp: 0.0.0.0
```

```
# network interfaces  
net:  
    port: 27017  
    bindIp: 127.0.0.1
```

Proves

```
root@debian:~# mongo -u admin -p admin 192.168.1.64:2717
MongoDB shell version v4.2.5
connecting to: mongodb://192.168.1.64:2717/test?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("704790c6-0bfe-4483-a893-f8635bbcd357") }
MongoDB server version: 4.2.5
Server has startup warnings:
2020-04-18T10:23:10.446+0000 I STORAGE [initandlisten]
2020-04-18T10:23:10.446+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2020-04-18T10:23:10.446+0000 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/prodnotes/filesystem
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten]
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten] ** Read and write access control configuration is unrestricted.
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten]
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten]
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten] ** We suggest setting it to 'never'.
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

> -
```

```
root@debian:~# mongo -u user1 -p user1 192.168.1.64:2717
MongoDB shell version v4.2.5
connecting to: mongodb://192.168.1.64:2717/test?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("b6649217-758c-4ad0-bbd9-2e20231dff33") }
MongoDB server version: 4.2.5
Server has startup warnings:
2020-04-18T10:23:10.446+0000 I STORAGE [initandlisten]
2020-04-18T10:23:10.446+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2020-04-18T10:23:10.446+0000 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/prodnotes/filesystem
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten]
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten] ** Read and write access control configuration is unrestricted.
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten]
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten]
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten] ** We suggest setting it to 'never'.
2020-04-18T10:23:11.186+0000 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

> -
```

Exportar

1. Mirem quin és l'id del contenidor que volem exportar.

```
# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS		PORTS	NAMES
eed9b8c64f0a	mongo	"docker-entrypoint.s..."	22 hours ago
Up 21 hours		0.0.0.0:2717->27017/tcp	mymongo

2. Exportem el contenidor a un .tar

```
# docker export eed9b8c64f0a > mymongo.tar
```

```
root@seax:~# docker export eed9b8c64f0a > mymongo.tar
root@seax:~# ls
mongo-docker mymongo.tar
```

Servidor DNS amb Docker

Tenir un servidor DNS és un avantatge perquè recordar noms és més fàcil que recordar IPs i podem tenir un lloc amb tots els IPs amb els seus corresponents noms. Abans de començar amb l'implementació hem de dir que el nostre domini s'anomena "VIA.com", a continuació s'explica com implementar un servidor DNS amb docker. La ip del nostre servidor i adreça són el 192.168.1.0 i 192.168.1.4

Passos

Creem el dockerfile i l'omplim de la següent manera:

```
root@seax:/docefila# touch Dockerfile
```

Afegim el següent contingut al Dockerfile:

```
1  FROM alpine:edge
2  LABEL maintainer="dev@jpillora.com"
3  # webproc release settings
4  ENV WEBPROC_VERSION 0.2.2
5  ENV WEBPROC_URL https://github.com/jpillora/webproc/releases/download/$WEBPROC_VERSION/webproc_linux_amd64.gz
6  # fetch dnsmasq and webproc binary
7  RUN apk update \
8      && apk --no-cache add dnsmasq \
9      && apk add --no-cache --virtual .build-deps curl \
10     && curl -sL $WEBPROC_URL | gzip -d - > /usr/local/bin/webproc \
11     && chmod +x /usr/local/bin/webproc \
12     && apk del .build-deps
13 #configure dnsmasq
14 RUN mkdir -p /etc/default/
15 RUN echo -e "ENABLED=1\\IGNORE_RESOLVCONF=yes" > /etc/default/dnsmasq
16 COPY dnsmasq.conf /etc/dnsmasq.conf
17 #run!
18 ENTRYPOINT ["webproc","--config","/etc/dnsmasq.conf","--","dnsmasq","--no-daemon"]
```

Construim la imatge apartir del Dockerfile:

```
[root@seax:/docefila]# docker build --tag dnsmasq .
```

Llavors creem el fitxer /opt/dnsmasq.conf i l'omplim amb el següent contingut:

```
[root@seax:/docefila]# touch /opt/dnsmasq.conf
```

Afegim el següent contingut al fitxer /opt/dnsmasq.conf:

```
[root@seax:/docefila]# cat /opt/dnsmasq.conf
log-queries
no-resolv
server=1.0.0.1
server=1.1.1.1
strict-order
server=/VIA/192.168.1.3
address=/VIA.com/192.168.1.4
[root@seax:/docefila]#
```

Després executem el contenidor:

```
[root@seax:/docefila]# docker run --name dnsmasq -d -p 53:53/udp -p 5380:8080 -v /opt/dnsmasq.conf:/etc/dnsmasq.conf --log-opt "max-size=100m" -e "HTTP_USER=foo" -e "HTTP_PASS=bar" --restart always jpilora/dnsmasq
```

Per provar el seu funcionament necessitem instal·lar 'host' amb la comanda 'apt install host'

Ara provem el correcte funcionament del nostre servidor DNS:

```
[root@seax:~# host VIA.com 192.168.1.2
Using domain server:
Name: 192.168.1.2
Address: 192.168.1.2#53
Aliases:

VIA.com has address 192.168.1.4
root@seax:~#
```

Com podem observar el ip de la nostra adreça és el 192.168.1.4 definit anteriorment.

Exportar el contingidor docker

Per exportar un servei:

1- Mirem el id del contingidor a exportar

2- Exportem el contingidor a un .tar

```
root@seax:/docefila# docker ps -a | grep dnsmasq
ebe48f3558a1      jpillora/dnsmasq      "webproc --config /e..."   39 minutes ago      Up 39 m
inutes            0.0.0.0:53->53/udp, 0.0.0.0:5380->8080/tcp  dnsmasq
root@seax:/docefila#
```

```
root@seax:/docefila# docker ps -a | grep dnsmasq
ebe48f3558a1      jpillora/dnsmasq      "webproc --config /e..."   39 minutes ago      Up 39
inutes            0.0.0.0:53->53/udp, 0.0.0.0:5380->8080/tcp  dnsmasq
root@seax:/docefila# docker export ebe48f3558a1 > dns.tar
Error response from daemon: No such container: ebe48f3558a1
root@seax:/docefila# docker export ebe48f3558a1 > dns.tar
root@seax:/docefila# ls
dns.tar  Dockerfile
root@seax:/docefila# _
```

3- Guardem el .tar en un USB al màquina destí. Qualsevol es pot fer servir un altre mètode com ssh o carpeta compartida

```
root@seax:/docefila# mkdir /media/usb-1
root@seax:/docefila# ls /dev/ | grep removable
root@seax:/docefila# mount /dev/sdb1
dns.tar  Dockerfile
root@seax:/docefila# mount /dev/sdb1 /media/usb-1
root@seax:/docefila# ls /media/usb-1/
ADSO          exit
'Alquiler del piso'    FOUND.000
aso-install.tar.gz  Imprimir
aso.sh        INDI
bachillerato.pdf  INDI.tar.gz
baseDeCotizacion issue
'beca 18-19'       'La sal de la Tierra (Subtitulado) (Elitetorrent.net).mp4'
comprovant     lftp_4.3.1-1_i386.deb
'Contrato del piso' lynx_2.8.8dev.9-2_all.deb
Creed.2015.DVDScr.XVID.AC3.SBT lynx-cur_2.8.8dev.9-2_i386.deb
CV             make_3.81-8.1_i386.deb
DELEA2         motd
'Disc d'        P2-Y1422332A-48093956J.zip
ESC2           'Q1(19-20)'
ESTA          'System Volume Information'
ex216.cc       The.Revenant.2015.DVDSCR.XViD.AC3.SBT
examenfinalFOPR.cc
root@seax:/docefila# ls
dns.tar  Dockerfile
root@seax:/docefila# cp dns.tar /media/usb-1/
root@seax:/docefila# _
```

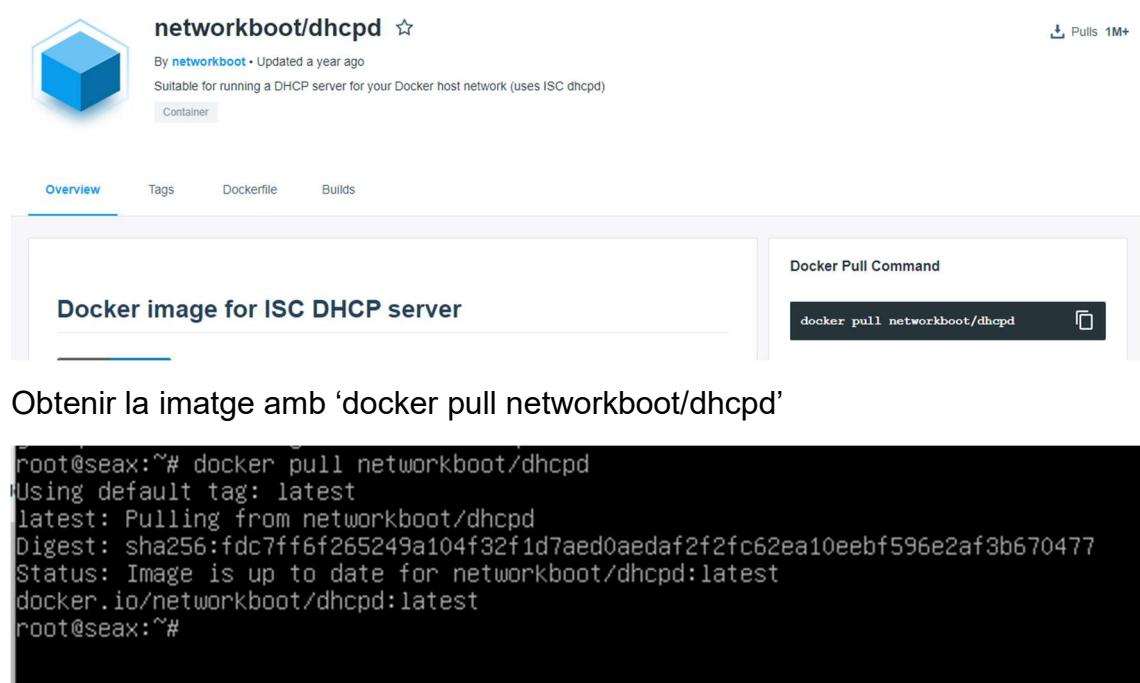
4- Importem la màquina virtual destí

```
root@seax:/docefila# docker ps -a | grep dnsmasq
ebe48f3558a1      jpillora/dnsmasq      "webproc --config /e..."   39 minutes ago      Up 39
inutes            0.0.0.0:53->53/udp, 0.0.0.0:5380->8080/tcp  dnsmasq
root@seax:/docefila# docker export ebe48f3558a1 > dns.tar
Error response from daemon: No such container: ebe48f3558a1
root@seax:/docefila# docker export ebe48f3558a1 > dns.tar
root@seax:/docefila# ls
dns.tar  Dockerfile
root@seax:/docefila# _
```

Servidor DHCP a Docker

Instal·lació de isc-dhcp-server

Instal·larem el servei isc-dhcp-server, per això podem aprofitar la següent imatge que ja que aporta el servei i té força popularitat.



Obtenir la imatge amb 'docker pull networkboot/dhcpd'

```
root@seax:~# docker pull networkboot/dhcpd
Using default tag: latest
latest: Pulling from networkboot/dhcpd
Digest: sha256:fdc7ff6f265249a104f32f1d7aed0aedaf2f2fc62ea10eebf596e2af3b670477
Status: Image is up to date for networkboot/dhcpd:latest
docker.io/networkboot/dhcpd:latest
root@seax:~#
```

Configuració

Per poder utilitzar la imatge, haurem de proporcionar l'arxiu dhcpd.conf, que correspon al fitxer on es configuren les xarxes del dhcp.

Per fer el següent, crearem al directori actual una carpeta anomenada /data i dins li col·locarem el fitxer amb un touch data/dhcpd.conf. Ara caldrà configurar les nostres xarxes, per tant obrirem el fitxer amb un nano data/dhcpd.conf i escriurem el següent:

```
root@seax:~# mkdir data_
root@seax:~# touch data/dhcpd.conf
```

Aquest serà el contingut de dhcpd.conf:

```
#default-lease-time 600;
#max-lease-time 7200;
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.8 192.168.1.250;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;
    option domain-name-servers 192.168.1.3;
}
```

Ara caldrà canviar la configuració de la xarxa que usrà el contenidor. Per això anirem obrirem amb nano el fitxer /etc/network/interfaces i modificarem l'arxiu així:

```
GNU nano 3.2                               /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

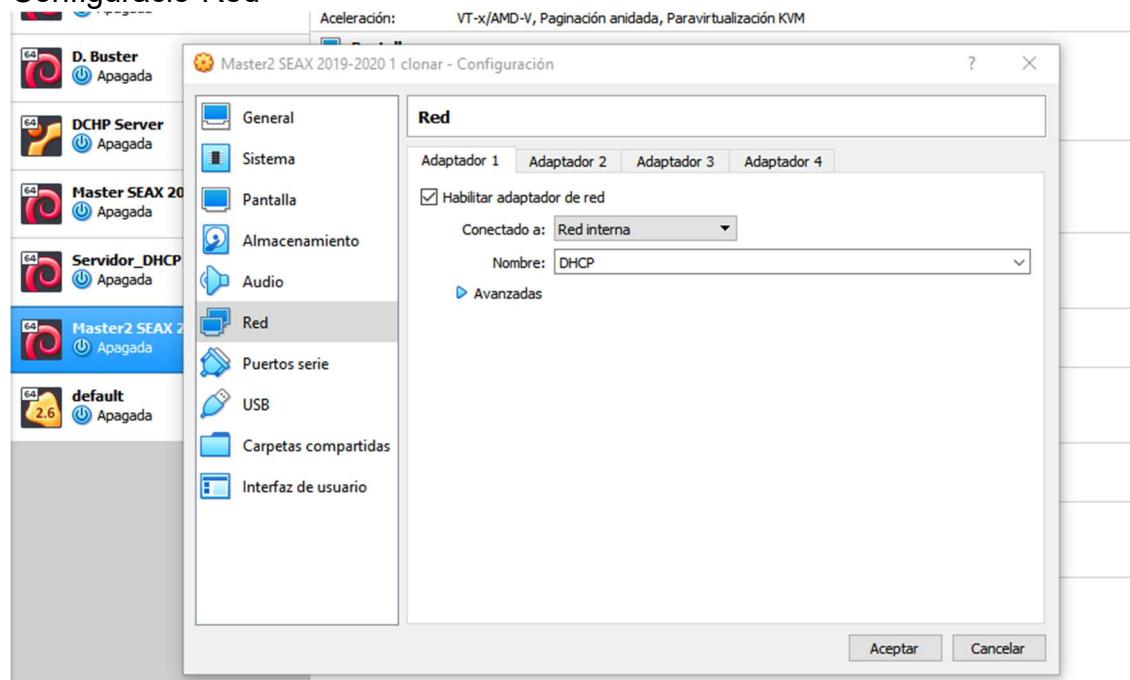
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
iface enp0s3 inet static
    address 192.168.1.2
    netmask 255.255.255.0
    network 192.168.1.0
    gateway 192.168.1.1
    broadcast 192.168.1.255
    dns-nameservers 192.168.1.3
```

A continuació ens caldrà apagar la màquina, posar-la en mode 'red interna', anomenar la nova xarxa 'DHCP' i tornar a obrir-la. Això es trobarà a l'apartat

Configuració-Red



Ara ja tenim tots els requisits i ja podem executar i provar el servei. L'executarem amb:

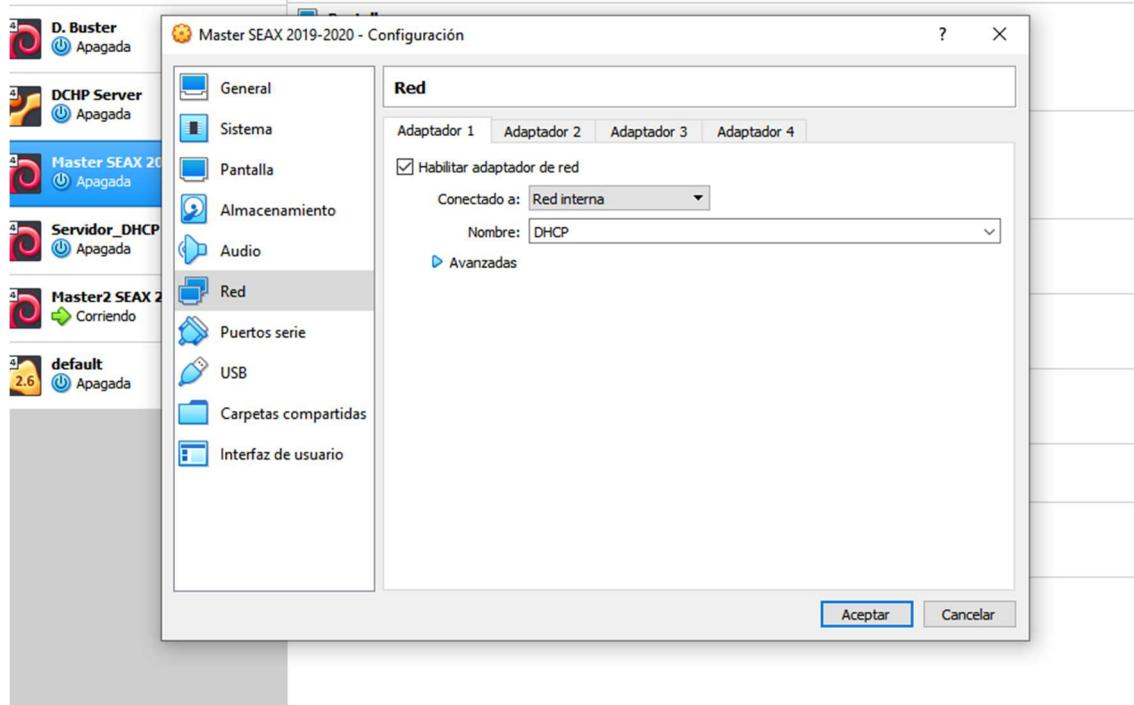
```
'docker run -it --rm --init --net host -v "$(pwd)/data":/data networkboot/dhcpd enp0s3'
```

Amb les tecles Ctrl +C podem acabar el servei.

```
root@seax:~# docker run -it --rm --init --net host -v "$(pwd)/data":/data networkboot/dhcpd enp0s3
Internet Systems Consortium DHCP Server 4.3.5
Copyright 2004-2016 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /data/dhcpd.conf
Database file: /data/dhcpd.leases
PID file: /var/run/dhcpd.pid
Wrote 1 leases to leases file.
Listening on LPF/enp0s3/08:00:27:91:ba:00/192.168.1.0/24
Sending on  LPF/enp0s3/08:00:27:91:ba:00/192.168.1.0/24
Sending on  Socket/fallback/fallback-net
Server starting service.
^Croot@seax:~#
```

Comprovació

Ara comprovarem el funcionament del servidor. Per això necessitarem una nova màquina i que el servidor estigui funcionant. Aquesta màquina només s'haurà de modificar el tipus de xarxa a Xarxa interna



Per verificar la connexió, només caldrà iniciar la màquina, si aquesta apareix una entrada al servidor, voldrà dir que està funcionant correctament. També es pot observar que la IP de la màquina sigui una vàlida dins el rang

Podem veure la IP del host amb 'ip a' i com es pot apreciar, està conectat amb la IP 192.168.1.9

```
entel@seax:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 000
    link/ether 08:00:27:1a:8b:bd brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.9/24 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 576sec preferred_lft 576sec
    inet6 fe80::1a00:27ff:fe1a:8bbd/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:74:6d:b9 brd ff:ff:ff:ff:ff:ff
entel@seax:~$
```

En el servidor podem observar el següent com li ha donat una direcció

```
root@seax:~# docker run -it --rm --init --net host -v "$(pwd)/data":/data networkboot/dhcpd enp0s3
Internet Systems Consortium DHCP Server 4.3.5
Copyright 2004-2016 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /data/dhcpd.conf
Database file: /data/dhcpd.leases
PID file: /var/run/dhcpd.pid
Wrote 1 leases to leases file.
Listening on LPF/enp0s3/08:00:27:91:ba:00/192.168.1.0/24
Sending on LPF/enp0s3/08:00:27:91:ba:00/192.168.1.0/24
Sending on Socket/fallback/fallback-net
Server starting service.
DHCPREQUEST for 10.0.2.15 from 08:00:27:1a:8b:bd via enp0s3: ignored (not authoritative).
DHCPREQUEST for 10.0.2.15 from 08:00:27:1a:8b:bd via enp0s3: ignored (not authoritative).
DHCPDISCOVER from 08:00:27:1a:8b:bd via enp0s3
DHCPoffer on 192.168.1.9 to 08:00:27:1a:8b:bd (seax) via enp0s3
DHCPREQUEST for 192.168.1.9 (192.168.1.2) from 08:00:27:1a:8b:bd (seax) via enp0s3
DHCPACK on 192.168.1.9 to 08:00:27:1a:8b:bd (seax) via enp0s3
^C
```

Amb això es pot donar per assegurat les proves de validació i ja podem passar a l'exportació.

Exportar/Importar

Per exportar el nostre dhcp caldrà usar la comanda: ‘docker save networkboot/dhcpd’ > dhcp-server.tar’. Ara tindrem la imatge preparada per posar-la a diferents màquines.

```
root@seax:~# docker save networkboot/dhcpd > dhcp-server.tar
root@seax:~# ls
data  dhcp-server.tar  docket-net.txt
root@seax:~# _
```

Per importar la imatge caldrà fer el proces a la inversa amb la comanda docker load -i dhcp-server.tar

```
root@seax:~# docker load -i dhcp-server.tar
Loaded image: networkboot/dhcpd:latest
root@seax:~# S_
```

Cloud Computing (computació en el núvol)

El "cloud computing" consisteix a utilitzar recursos informàtics a través d'Internet que no estan situats en un servidor concret d'Internet, sinó que hi ha diversos disponibles en centres de processaments de dades que als usuaris, els resulta indiferent, ja que és una cosa transparent i no percepren si usen el servidor A del servei Z o el servidor B del mateix servei Z.



Els servidors que proporcionen els serveis cloud, no se sap on estan però poden estar molt lluny físicament, per poder utilitzar aquests serveis s'ha d'enviar la informació a través d'internet i això portarà un temps dependent de la distància (entre client-servidor), la congestió de les xarxes...

Tipus de serveis del cloud

Podem agrupar els tipus de serveis que proporciona el "cloud computing" en tres grans blocs:

- IaaS (Infrastructure as a Service) o Infraestructura com a Servei: en ell, el client adquireix del proveïdor extern tots els recursos, com són servidors, espai d'emmagatzematge en un centre de dades o l'equipament per a xarxes. Normalment es duu a terme mitjançant una plataforma de virtualització. Dels tres tipus de serveis, és el que major control atorga a l'usuari i, per tant, el que té major complexitat d'ús.
- IaaS és ideal per a desenvolupadors, que poden triar Sistema Operatiu, quantitat d'emmagatzematge i altres possibilitats gràcies a la total transparència del servei. Exemples de IaaS són Microsoft Azure i Amazon Web Services.

- PaaS (Platform as a Service) o Plataforma com a Servei. Amb aquest model el proveïdor proporciona la plataforma per al desenvolupament d'aplicacions, així com el seu manteniment i gestió. En altres paraules, ofereix tot el que es necessita per a completar el cicle de vida i desplegament d'aquestes aplicacions o serveis web.

Aquí l'usuari manca de control sobre la infraestructura d'emmagatzematge o xarxes. PaaS manté l'escalabilitat automàtica, en funció del que exigeixi la situació. Exemple de Paas és Google App Engine, on els desenvolupadors poden crear les seves aplicacions a Java o Python.

- SaaS (Software as a Service) o Software com a Servei:
en aquest cas el client solament pot fer ús d'un programari que es troba allotjat en el núvol. Per tant, tot el relacionat amb el manteniment, desenvolupament, suport i operacions és responsabilitat de l'empresa que ofereix el servei.

El client usrà el sistema allotjat per aquesta empresa, la qual mantindrà la informació del client en els seus sistemes i proveirà els recursos necessaris per a explotar aquesta informació. En aquest grup podem posar com a exemple qualsevol servei web de correu electrònic, com Gmail.

Exemple de servei del cloud

Un exemple de servei de cloud pot ser l'ús de Siri, Amazon Alexa o Google Now, quan li parlem al dispositiu preguntant-li l'hora llavors aquesta grava la veu i l'envia (a través d'internet) a servidors remots que formen el cloud d'Apple, Amazon o Google respectivament. En aquests centres de processament de dades s'analitza la informació de l'enregistrament (el que diu la persona), es processa i es retorna una resposta, que, en aquest cas, fa que el dispositiu en qüestió sàpiga que ha de dir en veu alta l'hora actual. Tot això es fa en qüestió de fraccions de segon.

Serveis del cloud a l'aeroport (VIA)

Els servidors que tenim a la capa cloud del VIA són el WEB, BD, DNS i DHCP. A continuació expliquem els serveis amb el seu servidor:

WEB:

- Informació dels vols: proporciona informació com el codi identificador de cada vol, el destí, l'hora de sortida, l'hora d'arribada, la terminal, la porta d'embarcament i informació addicional com si el vol ha patit un retard o que s'hi ha cancel·lat.

- Reserva de vols: dóna l'opció de reservar un o varis dels vols que hi ha disponibles.
- Localització i informació dels cotxes autònoms: proporciona informació com el codi identificador de cada cotxe, l'estat que diu si està ocupat (porta passatgers) o no, la terminal on es troba, el destí (porta d'embarcament) cap a on va i el nombre de passatgers que porta.
- Reserva dels cotxes autònoms: dóna l'opció de cridar a un cotxe autònom al lloc on es vol per anar a un destí (porta d'embarcament) en concret.
- Informació dels passatgers: s'obté les dades personals (DNI, nom, cognoms) de cada passatger, el seu vol, la porta d'embarcament del vol i l'hora de sortida del vol.
Informació dels locals (restaurants i botigues) de l'aeroport: proporciona els productes, les ofertes i la localització de cada local.

BD:

- Obtenir els passatgers: obtenim tots els passatgers amb les seves dades com el codi identificador, el nom, els cognoms, la nacionalitat, el gènere, si és "VIP" o no, el número de telèfon, el perfil, el nom d'usuari, la contrasenya i un històric de tots els bitllets.
- Obtenir els bitllets: s'obté tots els bitllets amb la informació com el codi identificador, el passatger a qui pertany, l'identificador del vol, l'aerolínia, la data del vol, el seient, si és "VIP" o no i el nombre de maletes.
- Obtenir les ofertes: obtenim totes les ofertes amb l'identificador del local que ho ofereix, la descripció de l'oferta, la data de caducitat i el públic a qui va dirigit.
- Obtenir els vols: s'obté tots els vols amb el seu origen, el destí, la data del vol, l'hora del vol, l'aerolínia, la porta d'embarcament i l'estat del vol.
- Obtenir els negocis: obtenim tots els negocis el nom, el tipus, el local, una descripció, l'estat, el logo i una foto.

- Còpies de seguretat: es fa una còpia de seguretat de totes les dades per tal d'evitar les seves pèrdues.
- Emmagatzemar els passatgers: emmagatzemem els passatgers amb les seves dades com el codi identificador, el nom, els cognoms, la nacionalitat, el gènere, si és "VIP" o no, el número de telèfon, el perfil, el nom d'usuari, la contrasenya i un històric de tots els bitllets a la base de dades.
- Emmagatzemar els bitllets: emmagatzemem els bitllets amb la informació com el codi identificador, el passatger a qui pertany, l'identificador del vol, l'aerolínia, la data del vol, el seient, si és "VIP" o no i el nombre de maletes a la base de dades.
- Emmagatzemar les ofertes: emmagatzemem les ofertes amb l'identificador del local que ho ofereix, la descripció de l'oferta, la data de caducitat i el públic a qui va dirigit a la base de dades.
- Emmagatzemar els vols: emmagatzemem els vols amb el seu origen, el destí, la data del vol, l'hora del vol, l'aerolínia, la porta d'embarcament i l'estat del vol a la base de dades.
- Emmagatzemar els negocis: emmagatzemem tots els negocis el nom, el tipus, el local, una descripció, l'estat, el logo i una foto a la base de dades.

DNS:

- Resol nom de dominis: tradueix una direcció (VIA.com) a una adreça ip (192.168.1.3).

DHCP:

- Assigna direccions: assigna dinàmicament una adreça ip als diferents dispositius de la mateixa xarxa.

Subtasques dels serveis del cloud a l'aeroport (VIA)

A continuació dividim cada servei definit anteriorment en subtasques indicant els elements que hi participen i les responsabilitats que té cada subtasca, cal tenir en compte que les fletxes ← i → indiquen la direcció en la qual viatja la informació:

WEB:

- Informació dels vols:
 - Fer la petició a la base de dades per obtenir les dades (elements que participen: servidor WEB→ servidor BD): la seva responsabilitat és aconseguir de la base de dades l'informació dels vols.
 - Mostrar les dades (elements que participen: usuari←dispositiu←servidor WEB←servidor BD): la seva responsabilitat és oferir l'informació dels vols agafada de la base de dades.
- Reserva de vols:
 - Fer la petició a la base de dades per obtenir les dades (elements que participen: servidor WEB→ servidor BD): la seva responsabilitat és aconseguir de la base de dades l'informació dels vols.
 - Mostrar les dades (elements que participen: usuari←dispositiu←servidor WEB←servidor BD): la seva responsabilitat és oferir l'informació dels vols agafada de la base de dades.
 - Recollir l'opció escollida (elements que participen: usuari→dispositiu→servidor WEB): la seva responsabilitat és guardar el vol reservat.
 - Fer la petició de modificació a la base de dades (elements que participen: servidor WEB→ servidor BD): la seva responsabilitat és modificar el vol reservat a la base de dades.
 - Confirmar la reserva (elements que participen: usuari←dispositiu←servidor WEB←servidor BD): la seva responsabilitat és assegurar-se de que l'operació s'ha fet correctament.
- Localització i informació dels cotxes autònoms:
 - Fer la petició a la base de dades per obtenir les dades (elements que participen: servidor WEB→ servidor BD): la seva responsabilitat és aconseguir de la base de dades l'informació dels cotxes autònoms.
 - Mostrar les dades (elements que participen: usuari←dispositiu←servidor WEB←servidor BD): la seva responsabilitat és oferir l'informació dels cotxes autònoms de la base de dades.

- Reserva dels cotxes autònoms:
 - Fer la petició a la base de dades per obtenir les dades (elements que participen: servidor WEB→ servidor BD): la seva responsabilitat és aconseguir de la base de dades l'informació dels cotxes autònoms.
 - Mostrar les dades (elements que participen: usuari←dispositiu←servidor WEB←servidor BD): la seva responsabilitat és oferir l'informació dels vols agafada de la base de cotxes autònoms.
 - Recollir la reserva (elements que participen: usuari→dispositiu→servidor WEB): la seva responsabilitat és guardar el cotxe autònom reservat, és a dir, tenir-ho identificat.
 - Enviar l'avís al cotxe amb l'informació de la reserva (elements que participen: servidor WEB→cotxe): la seva responsabilitat és enviar les dades rellevants respecte la reserva com la ruta que ha de fer i els passatgers que ha de portar.
 - Confirmar la reserva (elements que participen: usuari←dispositiu←servidor WEB): la seva responsabilitat és assegurar-se de que la reserva s'ha fet correctament agafant el vist i plau del cotxe autònom i el reservant.
 - Assegurar que el cotxe arriba i en temps (elements que participen: servidor WEB→cotxe): la seva responsabilitat és anar guiant al cotxe en tot moment per assegurar que el cotxe arriba i en temps.
 - Mantenir informat de l'estat del cotxe (elements que participen: usuari←dispositiu←servidor WEB←cotxe): la seva responsabilitat és anar enviant la ubicació i l'estat del cotxe en tot moment al reservant.
- Informació dels passatgers:
 - Fer la petició a la base de dades per obtenir les dades (elements que participen: servidor WEB→ servidor BD): la seva responsabilitat és aconseguir de la base de dades l'informació dels passatgers.

- Mostrar les dades (elements que participen: usuari←dispositiu←servidor WEB←servidor BD): la seva responsabilitat és oferir l'informació dels passatgers de la base de dades.

BD:

- Obtenir els passatgers:
 - Accedir a la base de dades per obtenir les dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és obtenir tots els passatgers de la base de dades.
 - Mostrar les dades (elements que participen: usuari←dispositiu←servidor BD): la seva responsabilitat és oferir tots els passatgers de la base de dades obtinguts anteriorment.
- Obtenir els bitllets:
 - Accedir a la base de dades per obtenir les dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és obtenir tots els bitllets de la base de dades.
 - Mostrar les dades (elements que participen: usuari←dispositiu←servidor BD): la seva responsabilitat és oferir tots els bitllets de la base de dades obtinguts anteriorment.
- Obtenir les ofertes:
 - Accedir a la base de dades per obtenir les dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és obtenir totes les ofertes de la base de dades.
 - Mostrar les dades (elements que participen: usuari←dispositiu←servidor BD): la seva responsabilitat és oferir totes les ofertes de la base de dades obtingudes anteriorment.
- Obtenir els vols:
 - Accedir a la base de dades per obtenir les dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és obtenir tots els vols de la base de dades.

- Mostrar les dades (elements que participen: usuari←dispositiu←servidor BD): la seva responsabilitat és oferir tots els vols de la base de dades obtinguts anteriorment.
- Obtenir els negocis:
 - Accedir a la base de dades per obtenir les dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és obtenir tots els negocis de la base de dades.
 - Mostrar les dades (elements que participen: usuari←dispositiu←servidor BD): la seva responsabilitat és oferir tots els negocis de la base de dades obtinguts anteriorment.
- Còpies de seguretat:
 - Accedir a la base de dades per obtenir les dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és obtenir totes les dades de la base de dades i tenir preparat la nova base de dades on es farà la còpia.
 - Fer la còpia (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és fer la còpia de les dades obtingudes anteriorment i en la nova base de dades que s'ha preparat anteriorment.
 - Comprovar que s'ha fet correctament (elements que participen: usuari←dispositiu←servidor BD): la seva responsabilitat és comprovar que les dades de la base de dades i la nova base de dades, on s'ha fet la còpia anteriorment, són la mateixa.
- Emmagatzemar els passatgers:
 - Accedir a la base de dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és tenir preparat la taula dels passatgers a la base de dades per la introducció de noves dades.

- Introduir les dades a la base de dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és introduir les noves dades a la taula dels passatgers preparat anteriorment.
- Comprovar que s'ha fet correctament (elements que participen: usuari←dispositiu←servidor BD): la seva responsabilitat és assegurar que les noves dades hi són correctament a la base de dades.
- Emmagatzemar els bitllets:
 - Accedir a la base de dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és tenir preparat la taula dels bitllets a la base de dades per la introducció de noves dades.
 - Introduir les dades a la base de dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és introduir les noves dades a la taula dels bitllets preparat anteriorment.
 - Comprovar que s'ha fet correctament (elements que participen: usuari←dispositiu←servidor BD): la seva responsabilitat és assegurar que les noves dades hi són correctament a la base de dades.
- Emmagatzemar les ofertes:
 - Accedir a la base de dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és tenir preparat la taula de les ofertes a la base de dades per la introducció de noves dades.
 - Introduir les dades a la base de dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és introduir les noves dades a la taula de les ofertes preparat anteriorment.
 - Comprovar que s'ha fet correctament (elements que participen: usuari←dispositiu←servidor BD): la seva responsabilitat és assegurar que les noves dades hi són correctament a la base de dades.
- Emmagatzemar els vols:

- Accedir a la base de dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és tenir preparat la taula dels vols a la base de dades per la introducció de noves dades.
- Introduir les dades a la base de dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és introduir les noves dades a la taula dels vols preparat anteriorment.
- Comprovar que s'ha fet correctament (elements que participen: usuari←dispositiu←servidor BD): la seva responsabilitat és assegurar que les noves dades hi són correctament a la base de dades.
- Emmagatzemar els negocis:
 - Accedir a la base de dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és tenir preparat la taula dels negocis a la base de dades per la introducció de noves dades.
 - Introduir les dades a la base de dades (elements que participen: usuari→dispositiu→servidor BD): la seva responsabilitat és introduir les noves dades a la taula dels negocis preparat anteriorment.
 - Comprovar que s'ha fet correctament (elements que participen: usuari←dispositiu←servidor BD): la seva responsabilitat és assegurar que les noves dades hi són correctament a la base de dades.

DNS:

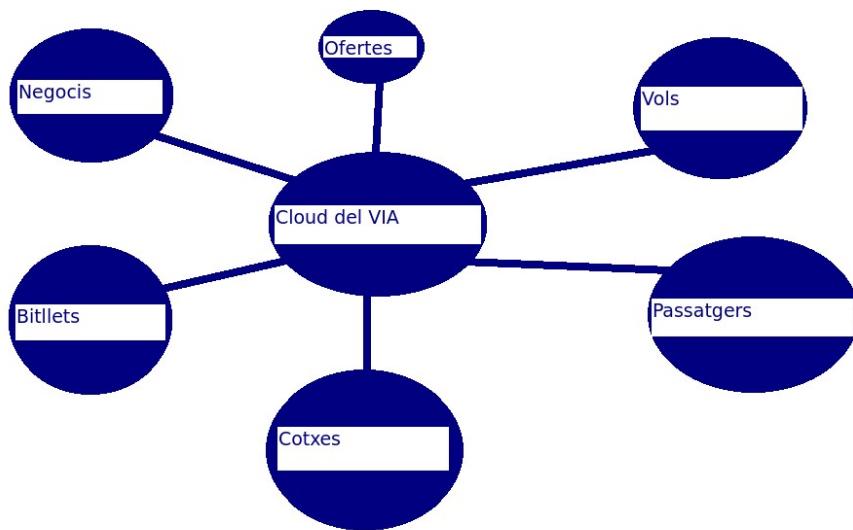
- Resoldre nom de dominis:
 - Traduir una direcció a una adreça ip (elements que participen: usuari→dispositiu→servidor DNS): la seva responsabilitat és assegurar que la traducció de la direcció a l'adreça ip es fa correctament.

DHCP:

- Assignar direccions:

- Detectar els dispositius de la xarxa (elements que participen: usuari←dispositiu←servidor DHCP): la seva responsabilitat és assegurar que tot dispositiu nou que entra a la xarxa està identificat.
- Assignar dinàmicament una adreça ip als diferents dispositius de la mateixa xarxa. (elements que participen: usuari←dispositiu←servidor DHCP): la seva responsabilitat és assignar adreces ips correctament als dispositius identificats anteriorment.

Esquema del cloud computing a l'aeroport (VIA):



Fog

En el fog els serveis que aparta a aquesta arquitectura, són els que requereixen poc temps.

Servidor Líder (1 o més):

Hem de tenir un servidor en el qual sigui capaç de manejar les diferents comunicacions entre les diferents BDD, on tindrem els líders que s'encarregaran de processar la informació i peticions que no requereixen l'ús dels recursos de la capa Cloud o d'altres nodes diferents. En cas de no disposar dels recursos per respondre a l'agent farà d'intermediari entre els agents i el cloud (serà qui respondrà a la petició).

BD:

Han d'estar més a prop de l'usuari, taules molt més petites que les del Cloud. Més fàcil i ràpid d'accedir-hi.

Serveis que proporciona la BDD:

- Guardar la ubicació dels usuaris (dels que són presents actualment en la xarxa de l'aeroport).

En aquesta BDD hauríem de tenir guardada un ID de l'usuari i la seva ubicació. Per detectar a l'usuari dintre de l'aeroport, un cop arribi i amb el mòbil es connecti a la xarxa local, l'app a través de la API enviarà periòdicament la ubicació a aquesta BDD on s'anirà actualitzant. Tenim que tenir en compte que si enviem periòdicament la ubicació de tothom, tenint 1000 passatgers a l'aeroport, la xarxa s'inundarà i es perdran paquets, per tant hauríem de prioritzar els més importants, si no tenim la ubicació en temps real dels passatgers no hi haurà problema sempre i quan podem tenir la ubicació semi-tempo real (periòdicament però cada minut o 2 minuts).

- Guardar la ubicació del la flota de cotxes.

En aquesta BDD hauríem de tenir guardada un ID del cotxe i la seva ubicació. El cotxes mai estaran connectats la xarxa local, a través de la API enviarà periòdicament la ubicació a aquesta BDD on s'anirà actualitzant. Com aquesta dada es més important que la ubicació dels passatgers (ja que si n'hi ha un accident el enrutament ha de ser precís), s'enviarà la posició cada 5 o 10 segons.

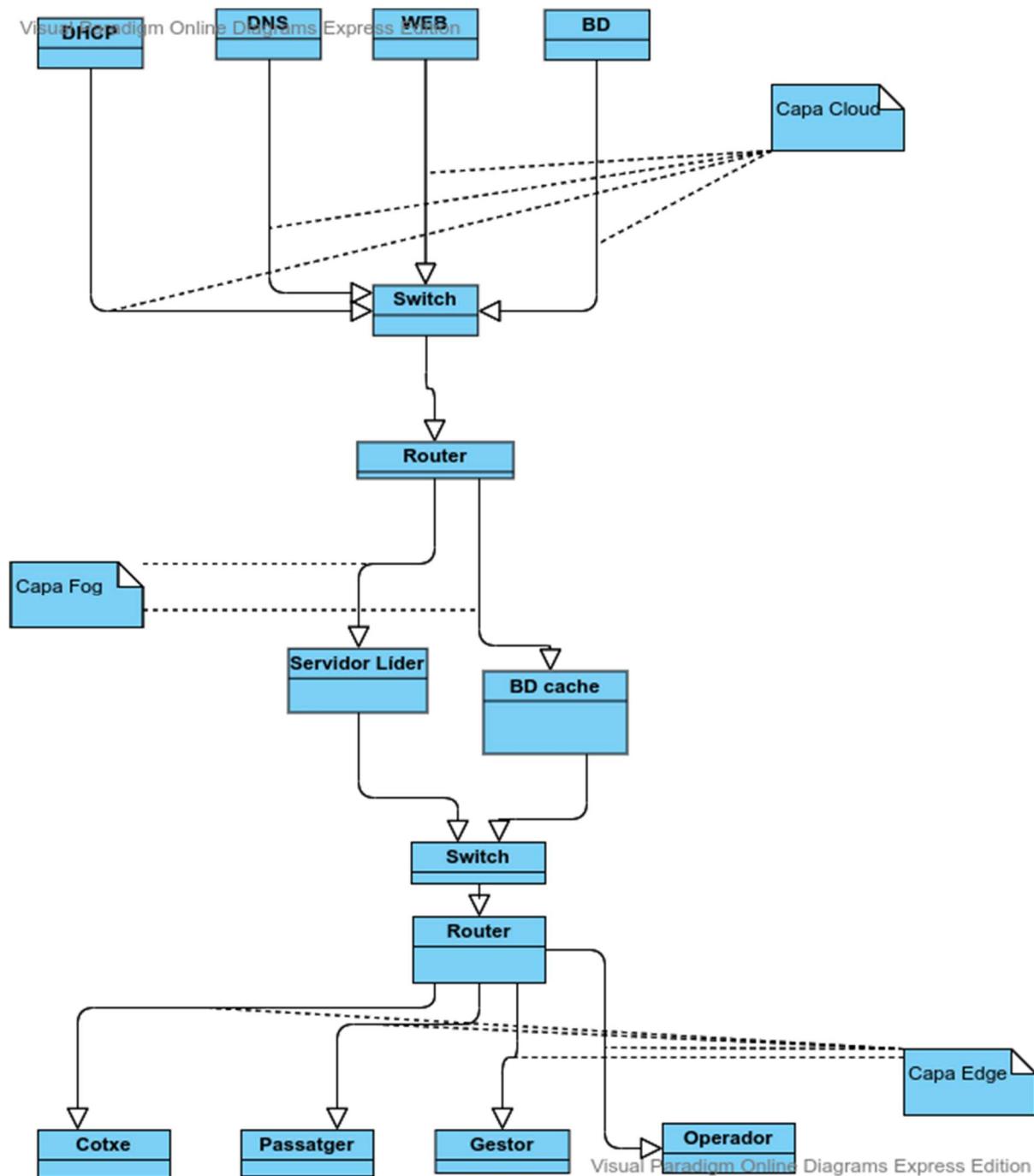
- Guardar la informació rellevant (horaris, porta d'embarcament, etc.) dels vols més recents.

Pel servei del sistema d'avisos és molt important tenir a prop de l'usuari aquesta informació, ja que se l'ha d'avalar en el menor temps possible de l'hora del vol i de la porta d'embarcament, sobretot si l'usuari es troba lluny de la porta y queda poc temps perquè surti el vol.

També hem pensat en tenir un servidor(mínim) de BDD que no tingui tota la informació que té la del cloud, en aquesta no volem un històric. Només volem els X registres més buscats en la BDD del cloud, per apropar a l'usuari la informació que més busca, sense tenir que anar constantment al núvol.

La idea d'aquesta BDD és la mateixa que la dels diferents nivells de caché. On cada BDD és més petita que la del nivell superior, però està més a prop de l'usuari, conté les dades més sol·licitades. D'aquesta forma aconseguiríem satisfer els requisits temporals de cada servei.

Esquema fog-to-cloud:



BURNDOWN

