

Primers passos amb Docker

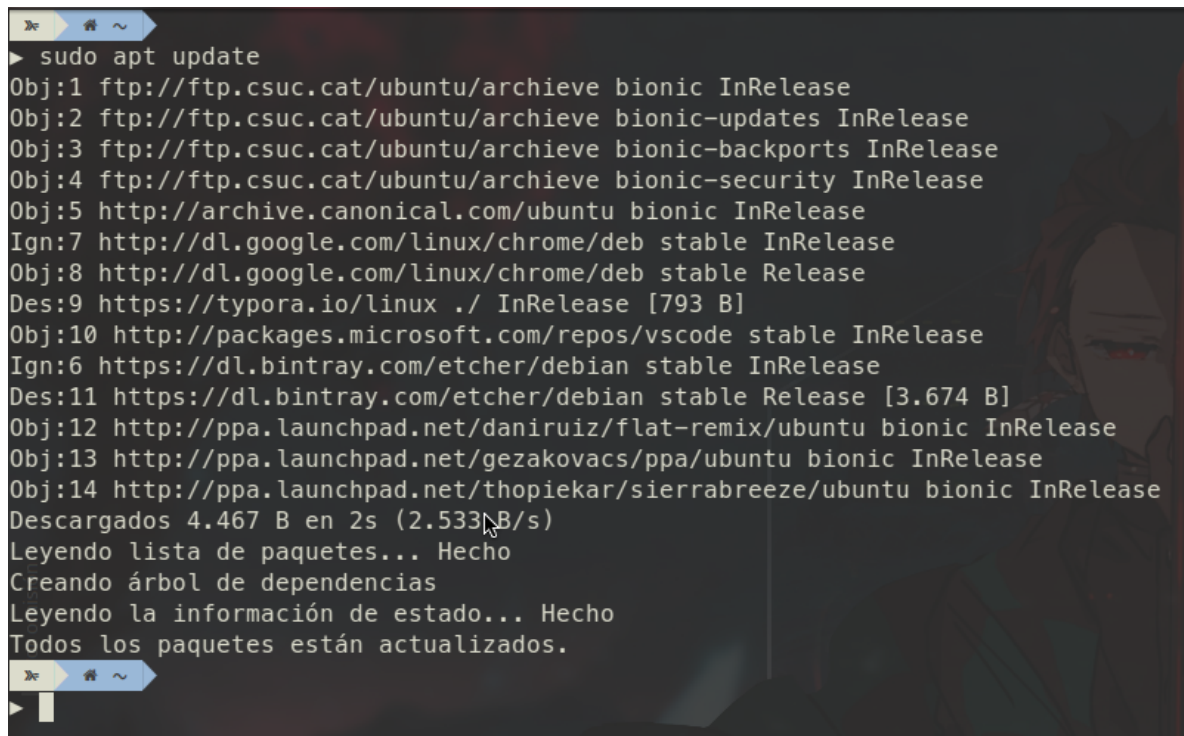
Primers passos amb Docker

- 1 - Instal·lar docker
- 2 - Baixar el contenidor
- 3 - Iniciem el contenidor
- 4 - Repetir pràctica 1.2
 - 4.1 - Crear user albums i BD
 - 4.2 - Donar password al usuari
 - 4.3 - Crear taula principal
 - 4.4 - Crear taules complementàries
 - 4.5 - Crear les claus foranes
 - 4.6 - Introdueix dades

1 - Instal·lar docker

- El primer serà realitzar un update del repositoris del sistema

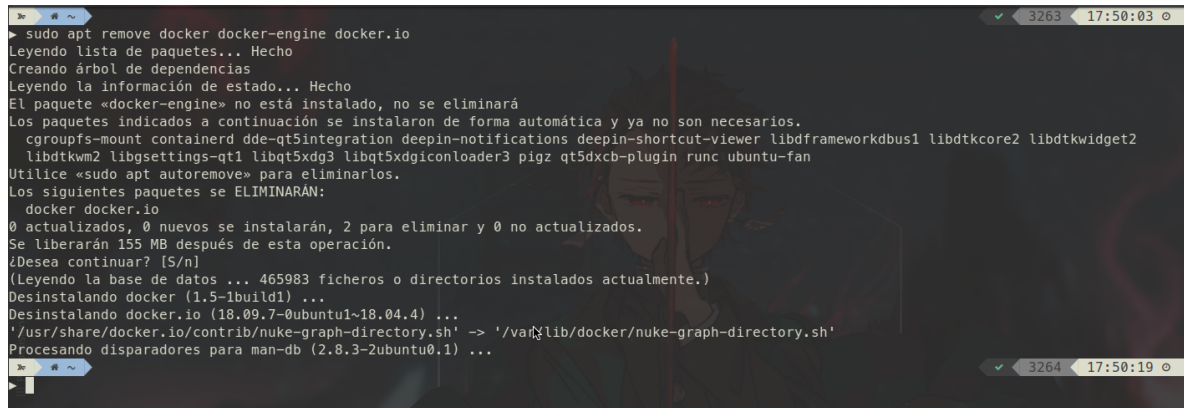
```
sudo apt update
```



```
> sudo apt update
Obj:1 ftp://ftp.csuc.cat/ubuntu/archive bionic InRelease
Obj:2 ftp://ftp.csuc.cat/ubuntu/archive bionic-updates InRelease
Obj:3 ftp://ftp.csuc.cat/ubuntu/archive bionic-backports InRelease
Obj:4 ftp://ftp.csuc.cat/ubuntu/archive bionic-security InRelease
Obj:5 http://archive.canonical.com/ubuntu bionic InRelease
Ign:7 http://dl.google.com/linux/chrome/deb stable InRelease
Obj:8 http://dl.google.com/linux/chrome/deb stable Release
Des:9 https://typora.io/linux./ InRelease [793 B]
Obj:10 http://packages.microsoft.com/repos/vscode stable InRelease
Ign:6 https://dl.bintray.com/etcher/debian stable InRelease
Des:11 https://dl.bintray.com/etcher/debian stable Release [3.674 B]
Obj:12 http://ppa.launchpad.net/danirui/flat-remix/ubuntu bionic InRelease
Obj:13 http://ppa.launchpad.net/gezakovacs/ppa/ubuntu bionic InRelease
Obj:14 http://ppa.launchpad.net/thopiekar/sierrabreeze/ubuntu bionic InRelease
Descargados 4.467 B en 2s (2.533B/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
```

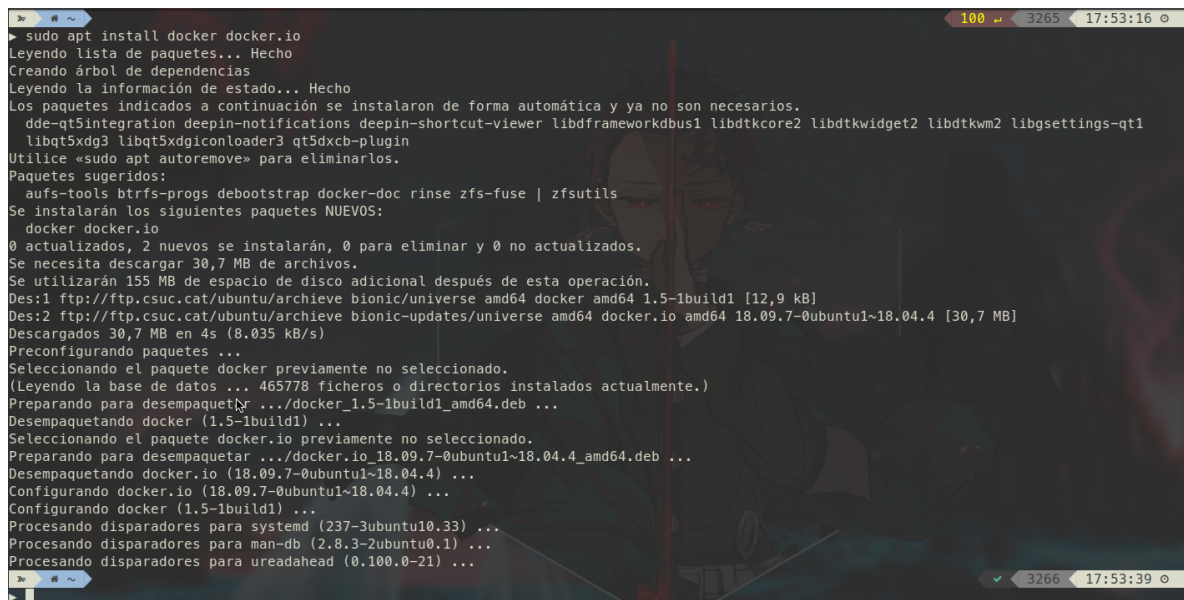
- Seguidament desinstal·larem les versions antigues de **Docker**

```
sudo apt remove docker docker-engine docker.io
```

A terminal window showing the command 'sudo apt remove docker docker-engine docker.io'. The output indicates that 'docker-engine' is not installed. It lists several other packages that will be removed along with Docker: 'cgrouperfs-mount', 'containerd', 'dde-qt5integration', 'deepin-notifications', 'deepin-shortcut-viewer', 'libdframeworkdbus1', 'libdtkcore2', 'libdtkwidget2', 'libdtkwm2', 'libgsettings-qt1', 'libqt5xdg3', 'libqt5xdgiconloader3', 'pigz', 'qt5xcb-plugin', 'runc', and 'ubuntu-fan'. It shows that 2 packages will be removed, freeing 155 MB of space. The user is prompted to confirm the removal, and they press 'Y'. The terminal shows the progress of removing the packages, including deleting files and cleaning up. The terminal window has a dark background with a faint image of a person's face in the background. The top bar shows the window title 'Terminal', a green checkmark, the number '3263', and the time '17:50:03'. The bottom bar shows a green checkmark, the number '3264', and the time '17:50:19'.

- Finalment només caldrà instal·lar el software

```
sudo apt install docker docker.io
```

A terminal window showing the command 'sudo apt install docker docker.io'. The output indicates that 'docker' and 'docker.io' will be installed. It shows the progress of downloading the packages, including the size of the files (12.9 kB for 'docker' and 30.7 MB for 'docker.io'). It also shows the progress of unpacking and configuring the packages. The terminal shows that the packages are being installed successfully. The terminal window has a dark background with a faint image of a person's face in the background. The top bar shows the window title 'Terminal', a green checkmark, the number '3265', and the time '17:53:16'. The bottom bar shows a green checkmark, the number '3266', and the time '17:53:39'.

2 - Baixar el contenidor

- Seguidament el que farem serà baixar el contenidor a la màquina per fer-lo servir en Docker

```
sudo docker pull postgres
```

```
➤ sudo docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
8ec398bc0356: Pull complete
65a7b8e7c8f7: Pull complete
b7a5676ed96c: Pull complete
3e0ac8617d40: Pull complete
633091ee8d02: Pull complete
b01fa9e356ea: Pull complete
4cd472257298: Pull complete
1716325d7dcd: Pull complete
9b625d69c7c8: Pull complete
74d8b4d9818c: Pull complete
c36f5edbeb97: Pull complete
9b38bb0fb36e: Pull complete
6b5ee1c74b9a: Pull complete
5fcc518252b4: Pull complete
Digest: sha256:3657548977d593c9ab6d70d1ffc43ceb3b5164ae07ac0f542d2ea139664eb6b3
Status: Downloaded newer image for postgres:latest
```

- Ara comprovarem que realment tenim el contenidor baixat

```
sudo docker image ls
```

```
➤ sudo docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
postgres            latest             ec5d6d5f5b34       2 weeks ago        394MB
```

3 - Iniciem el contenidor

- Ara realitzarem l'acció més important, iniciar el contenidor per al seu ús indicant "username", "password" i "database"

```
sudo docker run --name username -e POSTGRES_PASSWORD=password -d postgres
```

```
➤ sudo docker run --name carlos -e POSTGRES_PASSWORD=valknut21 -d postgres
928818ff5827091ccb89f1f77b671c536e9af370bf6dc61059a85031d1b87bd8
```

- I per ultim només caldrà accedir al servei del nostre contenidor amb les credencials inidicades al pas anterior

```
sudo docker run -it --rm --link username:postgres database psql -h postgres -U postgres
```

```
➤ sudo docker run -it --rm --link carlos:postgres postgres psql -h postgres -U postgres
Password for user postgres:
psql (12.1 (Debian 12.1-1.pgdg100+1))
Type "help" for help.

postgres=#
```

4 - Repetir pràctica 1.2

4.1 - Crear user albums i BD

- Crea un usuari àlbums i una base de dades amb el mateix nom.

```
CREATE USER albums WITH PASSWORD '123';
```

```
ALTER USER albums WITH SUPERUSER;
```

```
postgres=# CREATE USER albums WITH PASSWORD '123';
CREATE ROLE
postgres=# ALTER USER albums WITH SUPERUSER;
ALTER ROLE
postgres=#
```

```
CREATE DATABASE albums;
```

4.2 - Donar password al usuari

- Dóna-li una contrasenya a l'usuari.

```
CREATE USER albums WITH PASSWORD '123';
```

4.3 - Crear taula principal

- Crea la taula principal amb la que poder gestionar la teva col·lecció d'àlbums musicals (ID de tipus serial, títol de tipus varchar, autor de tipus varchar, suport de tipus varchar, data_edicio de tipus date, discogràfica de tipus varchar). Poseu el ID com a clau primària.

Primer que rés entrarem a la BD correcta amb l'usuari desitjat

```
sudo docker run -it --rm --link carlos:postgres postgres psql -h postgres -U
albums
```

```
~/Documentos/PostgreSQL master 3469 11:34:40
▶ sudo docker run -it --rm --link carlos:postgres postgres psql -h postgres -U albums
Password for user albums:
psql (12.1 (Debian 12.1-1.pgdg100+1))
Type "help" for help.

albums=#
```

Seguidament canviarem a la BD desitjada

```
\c albums
```

```
albums=# \c albums
You are now connected to database "albums" as user "albums".
albums=#
```

Ara ja podem començar a crear la taula

```
CREATE TABLE album (
    id            serial CONSTRAINT firstkey PRIMARY KEY,
    titol         varchar(40) NOT NULL,
    autor         varchar(40) NOT NULL,
    suport        varchar(40),
    data_edicio   date,
    discogràfica  varchar(40)
);
```

```
albums=# CREATE TABLE album (
albums(#      id            serial CONSTRAINT firstkey PRIMARY KEY,
albums(#      titol         varchar(40) NOT NULL,
albums(#      autor        varchar(40) NOT NULL,
albums(#      suport        varchar(40),
albums(#      data_edicio   date,
albums(#      discogràfica  varchar(40)
albums(# );
CREATE TABLE
albums=# \dt
          List of relations
 Schema | Name  | Type  | Owner
-----+-----+-----+-----
 public | album | table | albums
(1 row)

albums=#
```

4.4 - Crear taules complementàries

- Crea les següents taules complementàries (cadascuna en la seua clau primària):

- autor

```
CREATE TABLE autor (
    id            serial CONSTRAINT firstkeyautor PRIMARY KEY,
    nom           varchar(40) NOT NULL,
    cognom        varchar(100)
);
```

- discogràfica

```
CREATE TABLE discografica (
    id            serial CONSTRAINT firstkeydiscografica PRIMARY KEY,
    nom           varchar(40) NOT NULL,
    carrer        varchar(200)
);
```

- suport

```
CREATE TABLE suport (
    id          serial CONSTRAINT firstkeysuport PRIMARY KEY,
    nom         varchar(40) NOT NULL
);
```

```
albums=# CREATE TABLE autor (
albums=#     id          serial CONSTRAINT firstkeyautor PRIMARY KEY,
albums=#     nom         varchar(40) NOT NULL,
albums=#     cognom      varchar(100)
albums=# );
CREATE TABLE
albums=# CREATE TABLE discografica (
albums=#     id          serial CONSTRAINT firstkeydiscografica PRIMARY KEY,
albums=#     nom         varchar(40) NOT NULL,
albums=#     carrer      varchar(200)
albums=# );
CREATE TABLE
albums=# CREATE TABLE suport (
albums=#     id          serial CONSTRAINT firstkeysuport PRIMARY KEY,
albums=#     nom         varchar(40) NOT NULL
albums=# );
CREATE TABLE
albums=# \dt
      List of relations
Schema |      Name      | Type  | Owner
-----+-----+-----+-----
public | album          | table | albums
public | autor          | table | albums
public | discografica   | table | albums
public | suport         | table | albums
(4 rows)

albums=#
```

4.5 - Crear les claus foranes

- Crea les claus foranes necessàries a la taula principal.

```
ALTER TABLE album
    ADD id_autor serial CONSTRAINT fk_autor REFERENCES autor (id),
    ADD id_discografica serial CONSTRAINT fk_discografica REFERENCES discografica (id),
    ADD id_suport serial CONSTRAINT fk_suport REFERENCES suport (id);
```

```
albums=# ALTER TABLE album
albums=# ADD id_autor serial CONSTRAINT fk_autor REFERENCES autor (id),
albums=# ADD id_discografica serial CONSTRAINT fk_discografica REFERENCES discografica (id),
albums=# ADD id_suport serial CONSTRAINT fk_suport REFERENCES suport (id);
ALTER TABLE
albums=#
```

4.6 - Introdueix dades

- Introdueix dades a totes les taules (un parell de registres és suficient).
 - Suport

```
INSERT INTO suport (nom)
VALUES ('Basic');

INSERT INTO suport (nom)
VALUES ('Premium');
```

```
albums=# VALUES ('Basic');
INSERT 0 1
albums=#
albums=# INSERT INTO suport (nom)
albums=# VALUES ('Premium');
INSERT 0 1
albums=# select * from autor;
 id | nom | cognom
-----+-----+-----
(0 rows)

albums=# select * from suport;
 id | nom
----+----
 1 | Basic
 2 | Premium
(2 rows)

albums=#
```

◦ Discogràfica

```
INSERT INTO discografica (nom, carrer)
VALUES ('4KPower', 'Num 1 c/Algo'), ('Amazingnotes', 'Num 34 c/Algo2');
```

```
albums=# INSERT INTO discografica (nom, carrer)
albums=# VALUES ('4KPower', 'Num 1 c/Algo'), ('Amazingnotes', 'Num 34 c/Algo2');
INSERT 0 2
albums=# select * from discografica;
 id |      nom      |      carrer
-----+-----+-----
 1 | 4KPower       | Num 1 c/Algo
 2 | Amazingnotes  | Num 34 c/Algo2
(2 rows)

albums=#
```

◦ Autor

```
INSERT INTO autor (nom, cognom)
VALUES ('Skillet', 'Hard Rock'), ('Slipknot', 'Nu Metal');
```

```
albums=# INSERT INTO autor (nom, cognom)
albums=# VALUES ('Skillet', 'Hard Rock'), ('Slipknot', 'Nu Metal');
INSERT 0 2
albums=# select * from autor;
 id | nom      | cognom
-----+-----+-----
 1 | Skillet  | Hard Rock
 2 | Slipknot | Nu Metal
(2 rows)

albums=#
```

◦ Album

```
INSERT INTO album (titol, autor, suport, data_edicio, discogràfica,
id_autor, id_discografica, id_suport)
VALUES ('Comatose', 'Skillet', 'Basic', '2017-03-14', '4KPower', 1, 1,
1), ('Awake', 'Slipknot', 'Premium', '2017-04-30', 'Amazingnotes', 2, 2,
2);
```



```
albums=# INSERT INTO album (titol, autor, suport, data_edicio, discogràfica, id_autor, id_discografica, id_suport)
albums=# VALUES ('Comatose', 'Skillet', 'Basic', '2017-03-14', '4KPower', 1, 1, 1), ('Awake', 'Slipknot', 'Premium', '2017-04-30', 'Amazingnotes', 2, 2, 2);
INSERT 0 2
albums=# select * from album;
 id | titol  | autor  | suport | data_edicio | discogràfica | id_autor | id_discografica | id_supor
----+-----+-----+-----+-----+-----+-----+-----+-----
  1 | Comatose | Skillet | Basic  | 2017-03-14  | 4KPower      | 1        | 1                | 1
  2 | Awake   | Slipknot | Premium | 2017-04-30  | Amazingnotes | 2        | 2                | 2
(2 rows)

albums=#
```