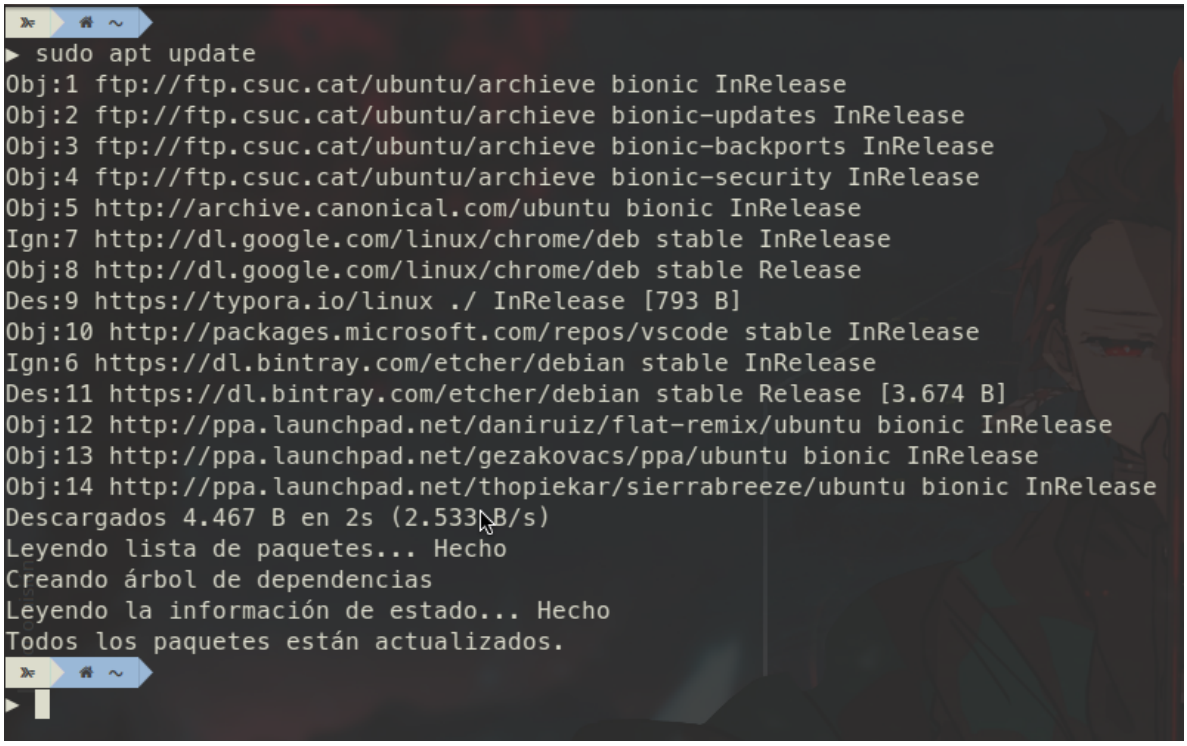
Primers pasos amb Docker

Primers pasos 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
```

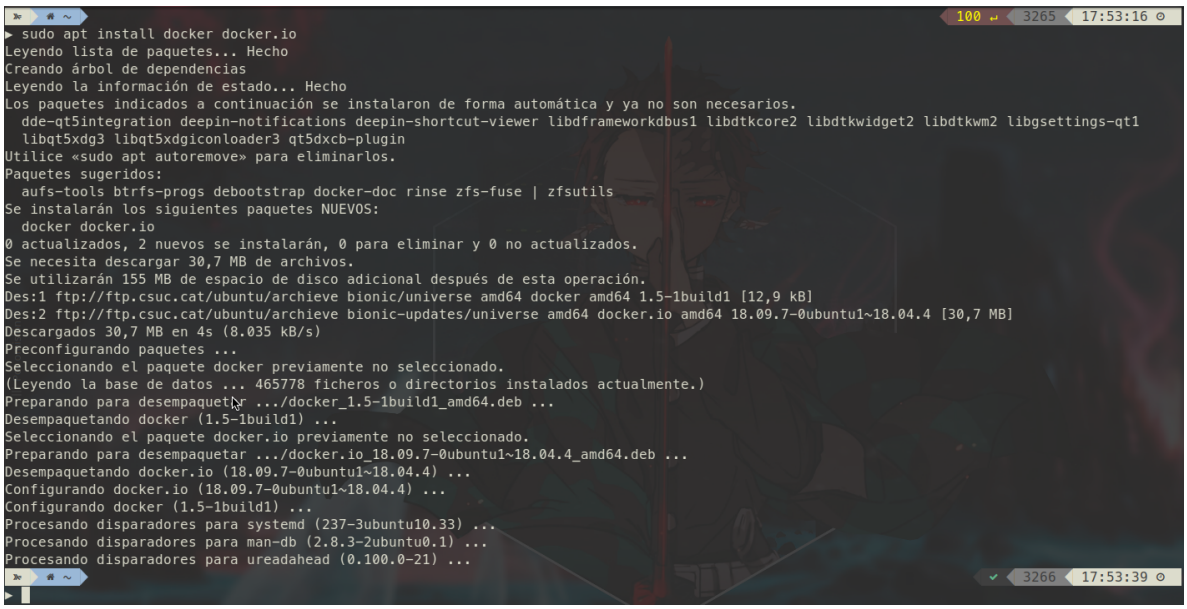


- Seguidament desinstal·larem les versions antigues de **Docker**

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

- Finalment només caldrà instal·lar el software

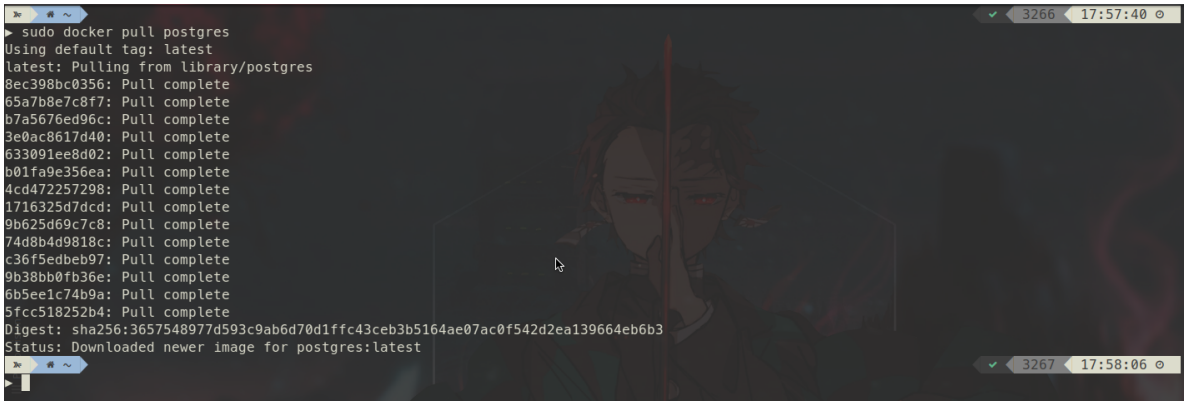
```
sudo apt install docker docker.io
```



2 - Baixar el contenidor

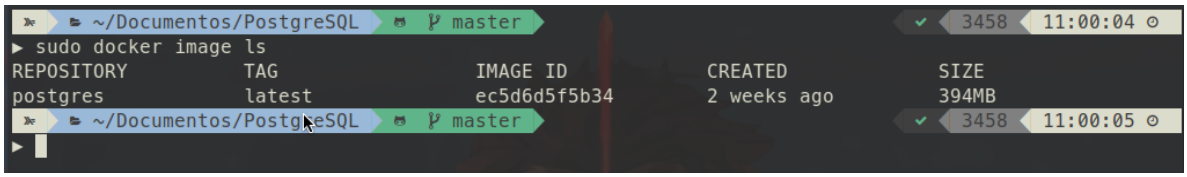
- Seguidament el que farem serà baixar el contenidor a la màquina per fer-lo servir en Docker

```
sudo docker pull postgres
```



- Ara comprovarem que realment tenim el contenedor baixat

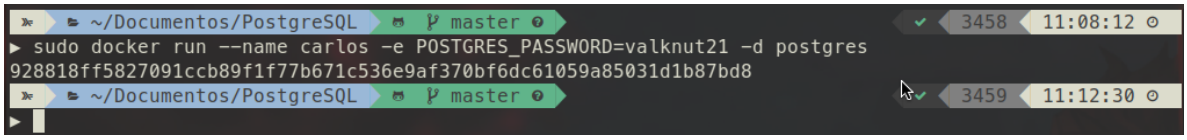
```
sudo docker image ls
```



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 database
```



- I per ultim només caldra 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
```

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;
```


```
CREATE DATABASE albums;
```

4.2 - Donar password al usuari

- Dóna-li una contrasenya a l'usuari.

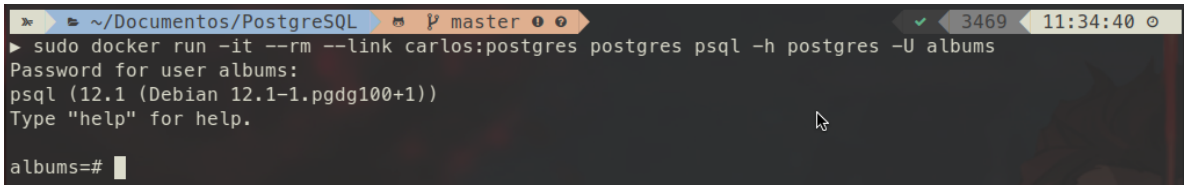
```
CREATE USER albums WITH PASSWORD '123';
```

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, titol 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
```



> 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

```sql
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

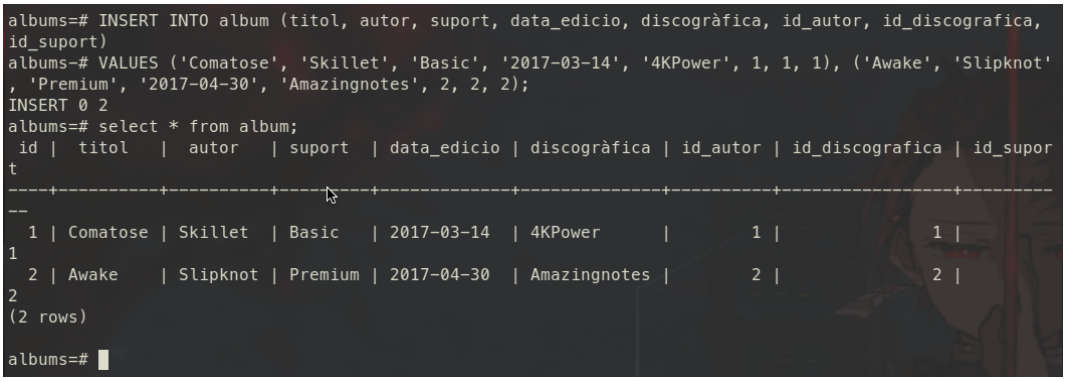
```sql
INSERT INTO autor (nom, cognom)
VALUES ('Skillet', 'Hard Rock'), ('Slipknot', 'Nu Metal');
```

```
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

```sql
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
t
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=#