

## Reporte Instalación Docker y PostgreSQL

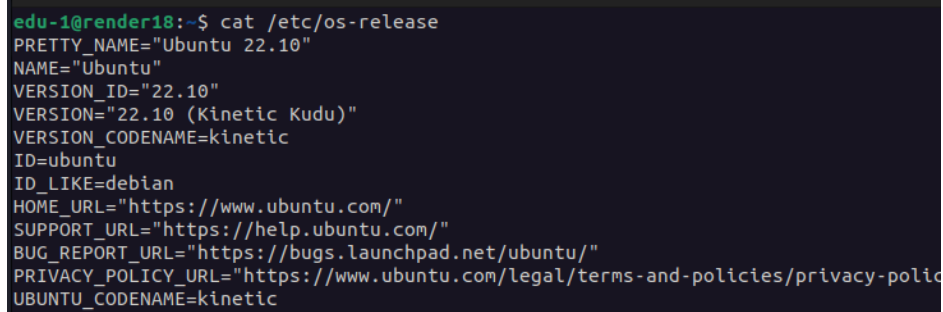
Integrante: Eduardo Montaña Gómez

### 1. Sistema Operativo y versión

Realizamos el siguiente comando para saber nuestro sistema operativo y versión:

```
1 $ cat /etc/os-release
```

Nos aparece lo siguiente:



```
edu-1@render18:~$ cat /etc/os-release
PRETTY_NAME="Ubuntu 22.10"
NAME="Ubuntu"
VERSION_ID="22.10"
VERSION="22.10 (Kinetic Kudu)"
VERSION_CODENAME=kinetic
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=kinetic
```

Por lo que el sistema operativo es *Ubuntu* y la version es 22.10.

Si nos referimos al kernel entonces hacemos

```
1 $ uname -a
```

Y obtengo que mi kernel es el 5.19.

### 2. Distribución

Podemos ver en el paso anterior que mi distribución es *Ubuntu*.

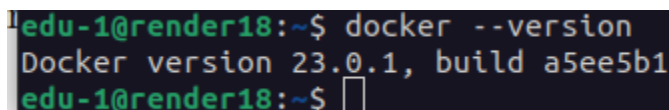
### 3. Versión

La versión de dbeaver instalada es la versión 22.3.5.

Ahora ejecuto el siguiente comando:

```
1 $ docker --version
```

Y mi versión de docker es 23.0.1



```
edu-1@render18:~$ docker --version
Docker version 23.0.1, build a5ee5b1
edu-1@render18:~$
```

## 4. Tiempo requerido

El tiempo total requerido fue de 1 hora. Empecé a las 10:30 de la mañana y terminé a las 11:30 am.

## 5. Explicación paso a paso

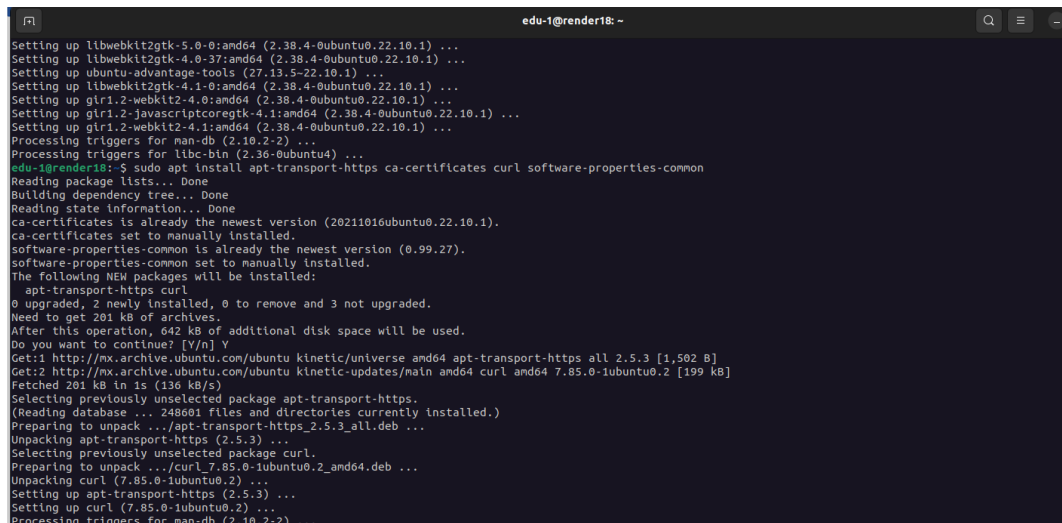
### 5.1. Instalación Docker

Realizamos el siguiente comando para actualizar el repositorio:

```
1 $ sudo apt update
```

Instalamos paquetes de requisitos previos:

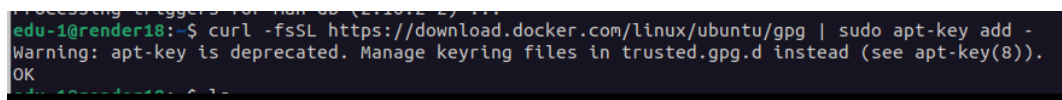
```
1 $ sudo apt install apt-transport-https ca-certificates curl  
2 software-properties-common
```



```
Setting up libwebkit2gtk-5.0-0:amd64 (2.38.4-0ubuntu0.22.10.1) ...  
Setting up libwebkit2gtk-4.0-37:amd64 (2.38.4-0ubuntu0.22.10.1) ...  
Setting up ubuntu-advantage-tools (27.13.5-22.10.1) ...  
Setting up libwebkit2gtk-4.1-0:amd64 (2.38.4-0ubuntu0.22.10.1) ...  
Setting up gir1.2-webkit2-4.0:amd64 (2.38.4-0ubuntu0.22.10.1) ...  
Setting up gir1.2-javascriptcoregtk-4.1:amd64 (2.38.4-0ubuntu0.22.10.1) ...  
Setting up gir1.2-webkit2-4.1:amd64 (2.38.4-0ubuntu0.22.10.1) ...  
Processing triggers for man-db (2.10.2-2) ...  
Processing triggers for libc-bin (2.36-0ubuntu4) ...  
edu-1@render18:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
ca-certificates is already the newest version (20211016ubuntu0.22.10.1).  
ca-certificates set to manually installed.  
software-properties-common is already the newest version (0.99.27).  
software-properties-common set to manually installed.  
The following NEW packages will be installed:  
  apt-transport-https curl  
0 upgraded, 2 newly installed, 0 to remove and 3 not upgraded.  
Need to get 201 kB of archives.  
After this operation, 642 kB of additional disk space will be used.  
Do you want to continue? [Y/n] Y  
Get:1 http://mx.archive.ubuntu.com/ubuntu kinetic/universe amd64 apt-transport-https all 2.5.3 [1,502 B]  
Get:2 http://mx.archive.ubuntu.com/ubuntu kinetic-updates/main amd64 curl amd64 7.85.0-1ubuntu0.2 [199 kB]  
Fetched 201 kB in 1s (136 kB/s)  
Selecting previously unselected package apt-transport-https.  
(Reading database ... 248601 files and directories currently installed.)  
Preparing to unpack .../apt-transport-https_2.5.3_all.deb ...  
Unpacking apt-transport-https (2.5.3) ...  
Selecting previously unselected package curl.  
Preparing to unpack .../curl_7.85.0-1ubuntu0.2_amd64.deb ...  
Unpacking curl (7.85.0-1ubuntu0.2) ...  
Setting up apt-transport-https (2.5.3) ...  
Setting up curl (7.85.0-1ubuntu0.2) ...  
Processing triggers for man-db (2.10.2-2) ...
```

Añadimos la clave GPG para el repositorio de Docker

```
1 $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo  
2 apt-key add -
```



```
edu-1@render18:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).  
OK
```

Agregamos el repositorio a las fuentes APT

```
1 $ sudo add-apt-repository  
2 "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"
```

Actualizamos lista de paquetes fuente con el paquete de Docker

```

Processing triggers for man-db (2.10.2-2) ...
edu-1@render18:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK

```

```
1 $ sudo apt update
```

```

edu-1@render18:~$ sudo apt update
Get:1 http://security.ubuntu.com/ubuntu kinetic-security InRelease [109 kB]
Hit:2 http://mx.archive.ubuntu.com/ubuntu kinetic InRelease
Get:3 http://mx.archive.ubuntu.com/ubuntu kinetic-updates InRelease [118 kB]
Hit:4 https://download.docker.com/linux/ubuntu focal InRelease
Get:5 http://mx.archive.ubuntu.com/ubuntu kinetic-backports InRelease [99.9 kB]
Fetched 326 kB in 1s (218 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: https://download.docker.com/linux/ubuntu/dists/focal/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
edu-1@render18:~$

```

Instalamos docker:

```
1 $ sudo apt install docker-ce
```

```

edu-1@render18:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin docker-scan-plugin git git-man liberror-perl libslirp0 pigz
  slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin docker-scan-plugin git git-man liberror-perl
  libslirp0 pigz slirp4netns
0 upgraded, 13 newly installed, 0 to remove and 3 not upgraded.
Need to get 116 MB of archives.
After this operation, 423 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://mx.archive.ubuntu.com/ubuntu kinetic/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io amd64 1.6.18-1 [28.2 MB]
Get:3 http://mx.archive.ubuntu.com/ubuntu kinetic/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:4 http://mx.archive.ubuntu.com/ubuntu kinetic-updates/main amd64 git-man all 1:2.37.2-1ubuntu1.4 [975 kB]
Get:5 http://mx.archive.ubuntu.com/ubuntu kinetic-updates/main amd64 git amd64 1:2.37.2-1ubuntu1.4 [3,287 kB]
Get:6 http://mx.archive.ubuntu.com/ubuntu kinetic/main amd64 libslirp0 amd64 4.7.0-1 [62.7 kB]
Get:7 http://mx.archive.ubuntu.com/ubuntu kinetic/universe amd64 slirp4netns amd64 1.2.0-1 [33.5 kB]
Get:8 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-buildx-plugin amd64 0.10.2-1-ubuntu.20.04-focal [25.9 MB]
Get:9 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-cli amd64 5:23.0.1-1-ubuntu.20.04-focal [13.2 MB]
Get:10 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce amd64 5:23.0.1-1-ubuntu.20.04-focal [22.0 MB]
Get:11 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-rootless-extras amd64 5:23.0.1-1-ubuntu.20.04-focal [8,765 kB]
Get:12 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-compose-plugin amd64 2.16.0-1-ubuntu.20.04-focal [10.2 MB]
Get:13 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-scan-plugin amd64 0.23.0-ubuntu-focal [3,622 kB]
Fetched 116 MB in 1min 35s (1,239 kB/s)
Selecting previously unselected package pigz.
(Reading database ... 248612 files and directories currently installed.)
Preparing to unpack .../00-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../01-containerd.io_1.6.18-1_amd64.deb ...
Unpacking containerd.io (1.6.18-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../02-docker-buildx-plugin_0.10.2-1-ubuntu.20.04-focal_amd64.deb ...

```

Comprobar servicio

```
1 $ sudo systemctl status docker
```

```
edu-1@render18:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Mon 2023-02-20 10:40:12 CST; 1min 46s ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
        Main PID: 21149 (dockerd)
          Tasks: 21
         Memory: 26.0M
            CPU: 23ms
       CGroup: /system.slice/docker.service
              └─21149 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Feb 20 10:40:11 render18 dockerd[21149]: time="2023-02-20T10:40:11.751388416-06:00" level=info msg="[core] [Channel #4 Subchannel #5] Subchannel Connectivity change to READY" module=grpc
Feb 20 10:40:11 render18 dockerd[21149]: time="2023-02-20T10:40:11.751326499-06:00" level=info msg="[core] [Channel #4] Channel Connectivity change to READY" module=grpc
Feb 20 10:40:11 render18 dockerd[21149]: time="2023-02-20T10:40:11.846803619-06:00" level=info msg="Loading containers: start."
Feb 20 10:40:11 render18 dockerd[21149]: time="2023-02-20T10:40:11.963632857-06:00" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/24"
Feb 20 10:40:12 render18 dockerd[21149]: time="2023-02-20T10:40:12.055640139-06:00" level=info msg="Loading containers: done."
Feb 20 10:40:12 render18 dockerd[21149]: time="2023-02-20T10:40:12.099062765-06:00" level=info msg="Docker daemon" commit=bc3805a graphdriver=overlay2 version=23.0.1
Feb 20 10:40:12 render18 dockerd[21149]: time="2023-02-20T10:40:12.099214165-06:00" level=info msg="Daemon has completed initialization"
Feb 20 10:40:12 render18 dockerd[21149]: time="2023-02-20T10:40:12.148871720-06:00" level=info msg="[Server #7] Server created" module=grpc
Feb 20 10:40:12 render18 systemd[1]: Started Docker Application Container Engine.
Feb 20 10:40:12 render18 dockerd[21149]: time="2023-02-20T10:40:12.153381777-06:00" level=info msg="API listen on /run/docker.sock"
lines 1-22/22 (END)
```

Comprobar si se puede trabajar con imágenes de Docker

```
1 $ sudo docker run hello-world
```

```
edu-1@render18:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
20b29710123e: Pull complete
Digest: sha256:688b62fe0089c419eaa0fd02d3905dd0952ad1feeaa67543f525c73a0a790febf
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (and64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
edu-1@render18:~$
```

## 5.2. Instalación Postgres

Ejecutamos los siguientes comandos:

Para que Docker use la imagen de postgres

```
1 $ docker pull postgres
```

```
edu-1@render18:~$ sudo docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
bb263680fed1: Pull complete
75a54e59e691: Pull complete
3ce7f8df2b36: Pull complete
f30287ef02b9: Pull complete
dc1f0e9024d8: Pull complete
7f0a68628bce: Pull complete
32b11818cae3: Pull complete
48111fe612c1: Pull complete
fcedb9c04393: Pull complete
8943748d4e1f: Pull complete
204b98eddef7: Pull complete
9e0624990483: Pull complete
01ebe7b28449: Pull complete
Digest: sha256:901df890146ec46a5cab7a33f4ac84e81bac2fe92b2c9a14fd649502c4adf954
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
edu-1@render18:~$
```

Crear el contenedor (En este paso se cambió la contraseña por una personal)

```
1 $ docker run -d --name postgres -e POSTGRES_PASSWORD=mysecretpassword -p
2 5432:5432 postgres
```

Averiguar el container ID

```
1 $ docker ps -a
```

```
edu-1@render18:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
1147b2f794e1   postgres      "docker-entrypoint.s..." 2 minutes ago  Up 2 minutes  0.0.0.0:5432->5432/tcp, :::5432->5432/tcp  postgres
6268add3e8ef   hello-world    "/hello"                 17 minutes ago Exited (0) 17 minutes ago  laughing_yalow
edu-1@render18:~$
```

Iniciamos el contenedor con el siguiente comando:

```
1 $ docker start id_container
```

Nos conectamos a PostgreSQL

```
1 $ docker run -it --link postgres:postgres postgres psql -h
2 postgres -U postgres
```

```
1147b2f794e1
edu-1@render18:~$ sudo docker run -it --link postgres:postgres postgres psql -h postgres -U postgres
Password for user postgres:
psql (15.2 (Debian 15.2-1.pgdg110+1))
Type "help" for help.

postgres=# help
You are using psql, the command-line interface to PostgreSQL.
Type: \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit
postgres=# \q
edu-1@render18:~$
```

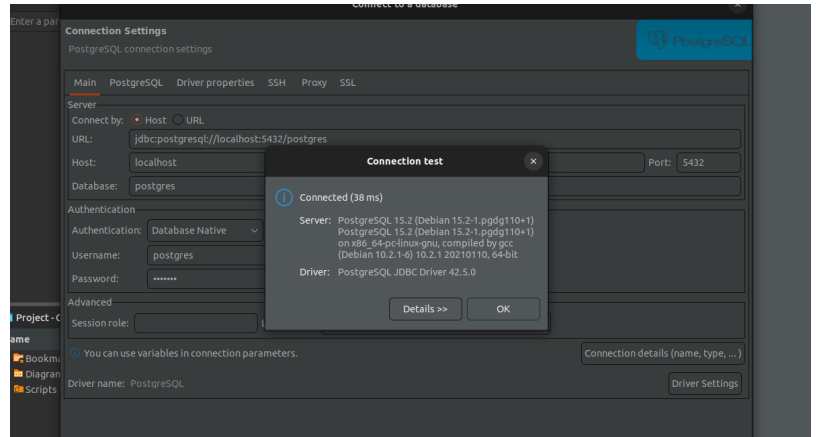
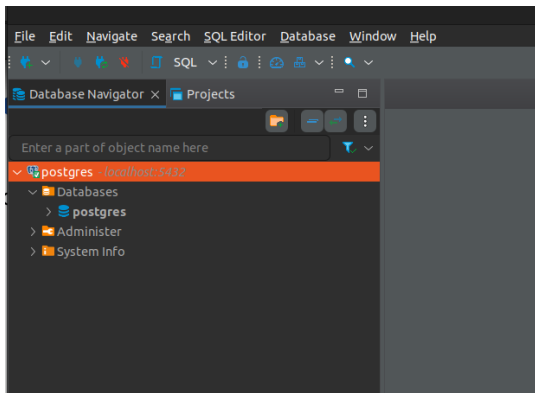
### 5.3. Instalación DBeaver

Instalamos DBeaver desde la página oficial, para ello, instalamos la versión linux (Linux Debian package installer).



- [Linux Debian package \(installer\)](#)
- [Linux RPM package \(installer\)](#)
- [Linux \(zip\)](#)
- [Linux \(zip without Java included\)](#)
- [Linux ARM \(zip without Java included\)](#)

Instalamos gráficamente y al concluir configuramos y este es el resultado:



## 6. Problemas adicionales

Al crear el contenedor de docker con el comando *docker run*, usé la misma contraseña que se ve en el pdf, por lo que tuve que pedir ayuda al ayudante.

Bastó con eliminar dicha imagen a través de su id, con *docker rmi id* y después repitiendo el proceso correctamente para tener mi contraseña personal.