

Reporte Instalación de Docker y del SMBD, PostgreSQL

Sistema operativo, distribución y versión

- S.O: Linux, Distribución: Zorin OS, Versión: 16.1.

```

vaniatown@vania-v-MHz:~$ neofetch
             vaniatown@vania-v-MHz
  OS: Zorin OS 16.1 x86_64
  Host: P2020M1 5.8
  Kernel: 5.15.0-46-generic
  Uptime: 1 hour, 30 mins
  Packages: 2845 (dpkg), 12 (flatpak), 10 (snap)
  Shell: Bash 5.0.17
  Resolution: 1366x768
  DE: GNOME
  WM: Mutter
  Theme: ZorinBlue-Light
  Icons: ZorinBlue-Light [GTK2/3]
  Terminal: gnome-terminal
  CPU: Intel Pentium N4200 (4) @ 2.500GHz
  GPU: Intel Celeron N520/Pentium N4200/Atom E3900 Series
  Memory: 2560MiB / 3757MiB
  
```

Instalación de Docker

- Se instaló la versión 27.5.1, utilizando la guía de instalación del documento de apoyo en la práctica. Ya que mi sistema operativo es derivado de ubuntu.
- Actualizamos paquetes.
- Instalamos dependencias y configuramos llaves de licencia por medio de curl HTTPS.

```

vaniatown@vania-v-MHz:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
[... output ...]
vaniatown@vania-v-MHz:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add
[... output ...]

```

■ Instalamos Docker:

```

vaniatown@Vania-v-MTz: ~$ sudo apt install docker-ce
[sudo] contraseña para vaniatown:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes empaquetados a continuación se instalarán de forma automática y ya no son necesarios.
gir1.2-gss-1.0 libtrafopugin1 libges-1.0-0 pttivi
utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin git git-man librerror-perl pigz slurpnetns
Paquetes sugeridos:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-symlink git-doc git-el git-email git-gui gitea git-cvs git-mediawiki git-svn
Se instalarán los siguientes paquetes NUEVOS:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin git git-man librerror-perl pigz slurpnetns
0 actualizados, 11 nuevos se instalarán, 0 para eliminar y 513 no actualizados.
Se necesita descargar 532 MB de archivos
Se utilizarán 497 MB de espacio en disco adicional después de esta operación.
¿Desea continuar? [S/n] S
Des1: https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io amd64 1.7.25-1 [29.6 MB]
Des2: https://mirror.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Des3: https://mirror.archive.ubuntu.com/ubuntu focal/main amd64 librerror-perl all 0.10029-1 [66.5 kB]
Des4: https://mirror.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all 1:2.25-1ubuntu0.13 [887 kB]
Des5: https://mirror.archive.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubuntu5.13 [4.612 kB]
Des6: https://mirror.archive.ubuntu.com/ubuntu focal/universe amd64 slurpnetns amd64 0.6.4-1 [76.3 kB]
Des7: https://download.docker.com/linux/ubuntu focal/stable amd64 docker-buildx-plugin amd64 0.20.0-1ubuntu.20.04-focal [33.2 MB]
Des8: https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-cli amd64 5:27.5.1-1-ubuntu.20.04-focal [16.2 MB]
Des9: https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce amd64 5:27.5.1-1-ubuntu.20.04-focal [126.1 MB]
Des10: https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-rootless-extras amd64 5:27.5.1-1-ubuntu.20.04-focal [9.012 kB]
Des11: https://download.docker.com/linux/ubuntu focal/stable amd64 docker-compose-plugin amd64 2.32.4-1-ubuntu.20.04-focal [12.8 MB]
Descargados 112 MB en 5s (2.433 kB/s)
Selecione el paquete pigz previamente no seleccionado.
(Leyendo la base de datos ... 795257 ficheros y directorios instalados actualmente.)
Preparando para desempacar .../00-pigz-2.4-1-amd64.deb ...
Desempacando pigz (2.4-1) ...
Selecione el paquete containerd.io previamente no seleccionado.
Preparando para desempacar .../01-containerd.io-1.7.25-1-amd64.deb ...
Desempacando containerd.io (1.7.25-1) ...

```

■ Comprobamos el servicio:

```

vaniatown@Vania-v-MTz: ~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2025-02-15 21:20:59 CST; 1min 15s ago
   TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
   Main PID: 22975 (dockerd)
   Tasks: 18
   Memory: 28.1M
   CGroup: /system.slice/docker.service
           └─22975 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Feb 15 21:20:58 Vania-v-MTz systemd[1]: Starting Docker Application Container Engine...
Feb 15 21:20:58 Vania-v-MTz dockerd[22975]: time="2025-02-15T21:20:58.292224608-08:00" level=info msg="Starting up"
Feb 15 21:20:58 Vania-v-MTz dockerd[22975]: time="2025-02-15T21:20:58.29995610-08:00" level=info msg="OTEL tracing is not configured, using no-op tracer provider"
Feb 15 21:20:58 Vania-v-MTz dockerd[22975]: time="2025-02-15T21:20:58.29641012-08:00" level=info msg="detected 127.0.0.53 nameserver, assuming system-resolved, so using it"
Feb 15 21:20:58 Vania-v-MTz dockerd[22975]: time="2025-02-15T21:20:58.31133204-08:00" level=info msg="Loading containers: start."
Feb 15 21:20:59 Vania-v-MTz dockerd[22975]: time="2025-02-15T21:20:59.31181402-08:00" level=info msg="Loading containers: done."
Feb 15 21:20:59 Vania-v-MTz dockerd[22975]: time="2025-02-15T21:20:59.3604404-08:00" level=info msg="Docker daemon" commit=69b30 containerd-shim-shutdown=false storage=overlayfs2
Feb 15 21:20:59 Vania-v-MTz dockerd[22975]: time="2025-02-15T21:20:59.36911985-08:00" level=info msg="Daemon has completed initialization"
Feb 15 21:20:59 Vania-v-MTz dockerd[22975]: time="2025-02-15T21:20:59.48671972-08:00" level=info msg="API listen on /run/docker.sock"
Feb 15 21:20:59 Vania-v-MTz systemd[1]: Started Docker Application Container Engine.
...skipping...
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2025-02-15 21:20:59 CST; 1min 15s ago
   TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
   Main PID: 22975 (dockerd)
   Tasks: 18
   Memory: 21.1M
   CGroup: /system.slice/docker.service
           └─22975 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Feb 15 21:20:58 Vania-v-MTz systemd[1]: Starting Docker Application Container Engine...
Feb 15 21:20:58 Vania-v-MTz dockerd[22975]: time="2025-02-15T21:20:58.292224608-08:00" level=info msg="Starting up"

vaniatown@Vania-v-MTz: ~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
080b3a04155: Pull complete
Digest: sha256:e0b56945135e00b4ce10a2087e7047e8f787a0e90798163f4a40404a108
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.
   (amd64)
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

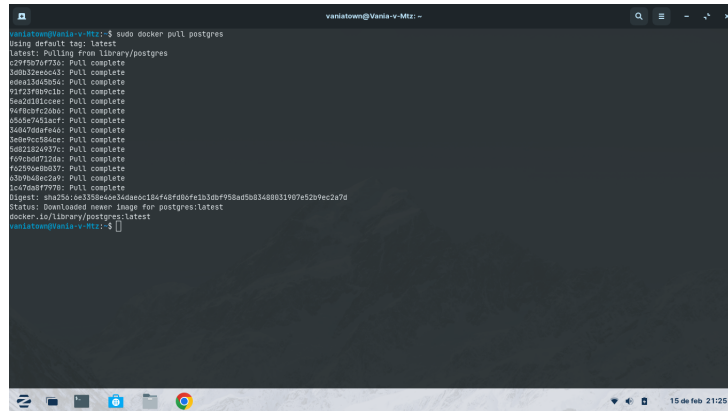
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

```

Instalación del SMBD PostgreSQL en contenedor de docker

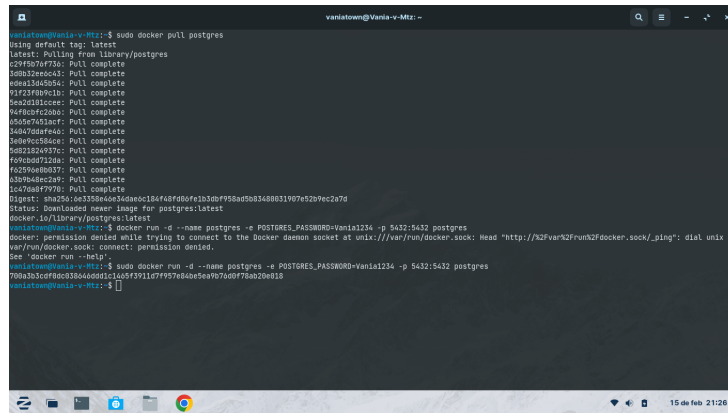
- Instalamos la imagen de posrgreSQL en Docker:



```

vaniatown@Vania-v-Mtz: ~$ sudo docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
c29f8b7a7f736: Pull complete
508b32eeec43: Pull complete
e0ea13a65054: Pull complete
91f23f8b9c1b: Pull complete
6e0c2818cc9e: Pull complete
9f88c9fc2086: Pull complete
a5d5e7d51ac7: Pull complete
346470d8ef40: Pull complete
3d8e9cc584ce: Pull complete
60d213a4937c: Pull complete
f09c8d0712d4: Pull complete
f0239e8b8037: Pull complete
a3b9b48ec2a9: Pull complete
1c476a8f7770: Pull complete
Digest: sha256:1c330e40d3d6c0c8b6468f8b0fe1b3dbf958a5b03488031907e520ec2a7d
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
vaniatown@Vania-v-Mtz: ~$
  
```

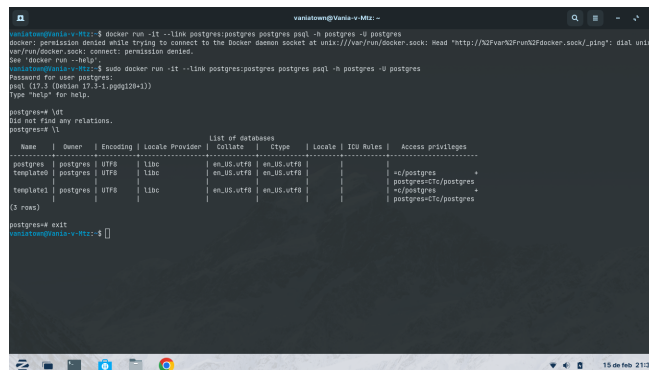
- Configuramos contraseña y puerto para el servicio. Y prendemos el contenedor de postgresQL.



```

vaniatown@Vania-v-Mtz: ~$ sudo docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
c29f8b7a7f736: Pull complete
508b32eeec43: Pull complete
e0ea13a65054: Pull complete
91f23f8b9c1b: Pull complete
6e0c2818cc9e: Pull complete
9f88c9fc2086: Pull complete
a5d5e7d51ac7: Pull complete
346470d8ef40: Pull complete
3d8e9cc584ce: Pull complete
60d213a4937c: Pull complete
f09c8d0712d4: Pull complete
f0239e8b8037: Pull complete
a3b9b48ec2a9: Pull complete
1c476a8f7770: Pull complete
Digest: sha256:1c330e40d3d6c0c8b6468f8b0fe1b3dbf958a5b03488031907e520ec2a7d
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
vaniatown@Vania-v-Mtz: ~$ docker run -d --name postgres -e POSTGRES_PASSWORD=vanial234 -p 5432:5432 postgres
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://127.0.0.1:2375/v1.42/run/docker.sock/_ping": dial unix /var/run/docker.sock: connect: permission denied.
See "docker run --help".
vaniatown@Vania-v-Mtz: ~$ sudo docker run -d --name postgres -e POSTGRES_PASSWORD=vanial234 -p 5432:5432 postgres
700a3b3c8d8c3b6a6d0d1c1a05f91107f957b04a5e9b7600778ab20e018
vaniatown@Vania-v-Mtz: ~$
  
```

- Probamos postgresQL en su terminal para ver que esté bien instalado.



```

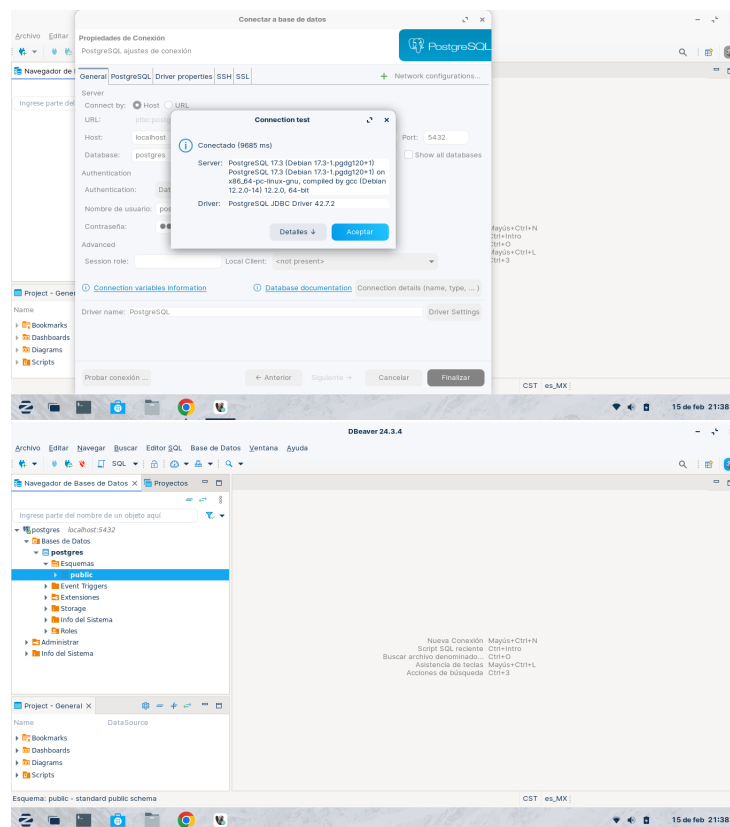
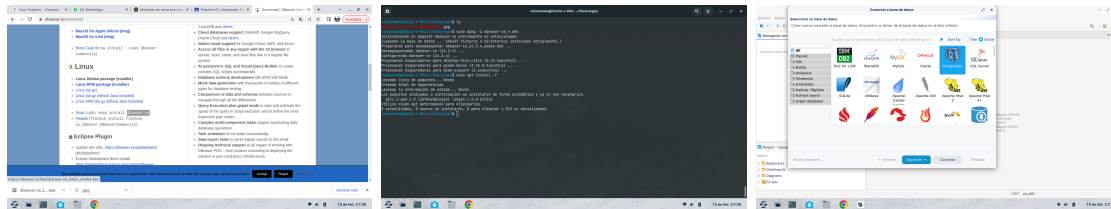
vaniatown@Vania-v-Mtz: ~$ docker run -it --link postgres:postgres postgres psql -h postgres -U postgres
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://127.0.0.1:2375/v1.42/run/docker.sock/_ping": dial unix /var/run/docker.sock: connect: permission denied.
See "docker run --help".
vaniatown@Vania-v-Mtz: ~$ sudo docker run -it --link postgres:postgres postgres psql -h postgres -U postgres
Password for user postgres:
psql (17.5 (Ubuntu 17.5-1.pgd12b1))
Type "help" for help.

postgres=# \ut
\ut not find any relations.
postgres=# \l
          name | owner  | encoding | locale provider | list of databases | collate | ctype | locale | ICU Rules | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
 postgres | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |      |      |      |      | postgres/postgres
 template1 | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |      |      |      |      | postgres/postgres
 template1 | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |      |      |      |      | postgres/postgres
(3 rows)

postgres=# exit
vaniatown@Vania-v-Mtz: ~$
  
```

Conexión a DBeaver

- Instalamos DBeaver.
- Posteriormente seleccionamos el tipo de conexión postgresQL para que nos instale todos los drivers necesarios.
- Y por ultimo probamos conexión:



Tiempo requerido Comentarios y problemas de la instalación

El único problema que se tenía era que las llaves de las licencias de Google no estaban actualizadas por lo que se agregaron de manera manual. Así que, la instalación completa tardó aproximadamente 20 minutos.