|  |  |
| --- | --- |
| POSTGRESQL DOCKER  Manual de Postgresql con Docker |  |
|  | **BASES DE DATOS DISTRIBUIDAS (OPTATIVA 1)** **Juan Daniel Ávila Caicedo**  **Nicolas Sanchez Barahona**  **Nicolas Caro Ñustez** |

Contenido

[**BASES DE DATOS DISTRIBUIDAS (OPTATIVA 1)** 2](#_Toc71656979)

[Descarga e instalación de Docker 1](#_Toc71656980)

[Puesta a punto 2](#_Toc71656981)

[Visualización de contenedores 2](#_Toc71656982)

[Detener contenedores 2](#_Toc71656983)

[Eliminar contenedores 2](#_Toc71656984)

[Creación de volumen (--run) 4](#_Toc71656985)

[Conexión cliente 5](#_Toc71656986)

[Docker compose 6](#_Toc71656987)

[Services: 6](#_Toc71656988)

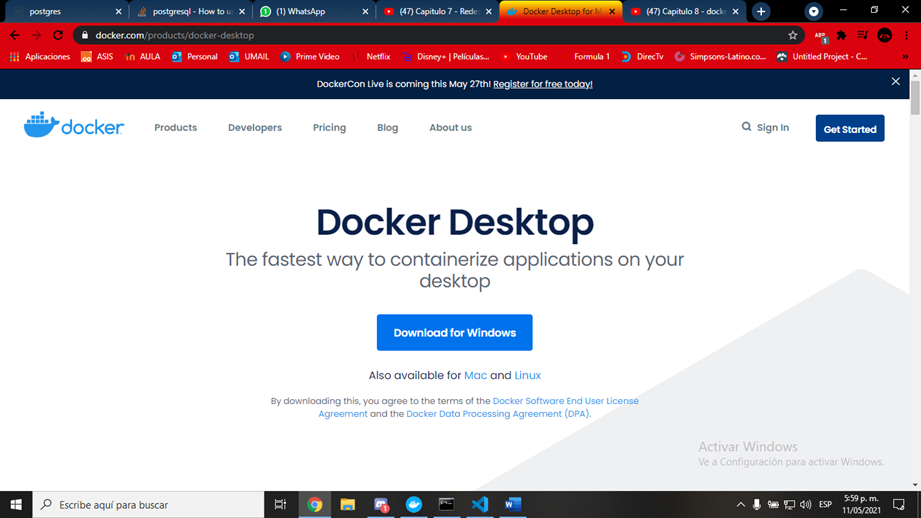
[Adminer: 6](#_Toc71656989)

[Build up: 6](#_Toc71656990)

[Confirmación de funcionamiento de Adminer 8](#_Toc71656991)

# Descarga e instalación de Docker

Descargamos e instalamos Docker de la página oficial <https://www.docker.com/get-started>



Una vez instalado el Docker debe lucir de esta manera y adicional debe estar activo el servicio.

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

# Puesta a punto

## Visualización de contenedores

Texto

Descripción generada automáticamente

# docker ps

Se hace uso de este comando para ver cuales y cuantos contenedores se están ejecutando en el momento.

## Detener contenedores

# docker stop 18

# docker stop 6c

Con estos comandos se detienen los contenedores identificados por las iniciales del CONTAINER ID “18 y 6c”.

Texto

Descripción generada automáticamente

# docker ps -a

Se hace uso de este comando para identificar los contenedores que están detenidos.

## Eliminar contenedores

Texto

Descripción generada automáticamente

# docker system prune

Se hace uso de este comando para eliminar todos los contenedores que están detenidos y así limpiar el buffer de contenedores.

Texto

Descripción generada automáticamente

Para verificar que todos los contenedores se hayan eliminado se ejecutan los comandos de visualización de contenedores y el de visualización de contenedores en segundo plano.

Texto

Descripción generada automáticamente

# docker run --name some-postgres -p 5432:5432 -e POSTGRES\_PASSWORD=mysecretpassword -d postgres

Este comando se utiliza para inicializar el contenedor, el cual se compone –name “some-postgres”, este nombre es asignado al contenedor, -p 5432:5432, este fragmento se encarga de la asignación de puertos, -e POSTGRES\_PASSWORD se encarga de la asignación de la contraseña del postgres, -d postgres se encarga de dejar el contenedor ejecutado en segundo plano.

# docker exec -it some-postgres bash

Este comando permite inicializar el contenedor y a la vez entrar a la terminal para la manipulación del postgres.

# su postgres

Se utiliza este comando para entrar desde el usuario de postgres.

# psql

Se usa para la manipulación de los schemas de postgres.

Texto

Descripción generada automáticamente

# ipconfig

Se usa este comando para conocer la ip de la maquina física, para la futura conexión con un cliente de administración (PgAdmin4).

# Creación de volumen (--run)

Texto

Descripción generada automáticamente

# docker volume create postgresV

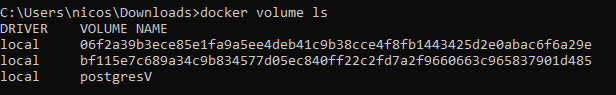
Se usa este comando para crear el volumen que almacenara los datos del contenedor.

# docker volume inspect postgresV

Se usa este comando para obtener mas detalles del volumen que se creó previamente.

# docker volume ls

Se usa este comando para listar los volúmenes de los contenedores que se encuentran en la maquina física.



# --network

En este apartado se va a configurar la red de los contenedores que se disponen, como primer paso se debe crear la red, para ello se usa este comando:



Ejecutando este comando se crea una red de nombre redPrueba con su respectivo Id.

Para verificar su creación, se puede hacer uso del siguiente comando, el cual proporciona una lista con la Id de las redes y su respectivo nombre.

Texto

Descripción generada automáticamente

Incluso si se precisa de más información acerca de la red se puede utilizar el siguiente comando

Texto

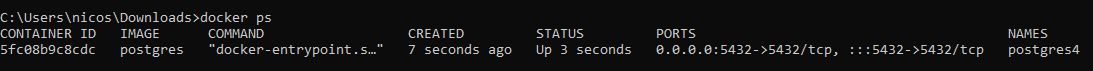
Descripción generada automáticamente

# Creación de contenedor asociado con red creada

Para la creación de un contenedor que este asociado con una red creada se debe ejecutar el comando de creación estándar, previamente expuesto, pero esta vez se debe agregar a la línea de comando el siguiente parámetro --network redPrueba, esto para asociar la red al contenedor.



Para verificar que este proceso se haya llevado a cabo correctamente, se usa el siguiente comando



Además de esta información, se puede validar que la red que está siendo asociada, este verdaderamente vinculada al contenedor, esto con el siguiente comando

Texto

Descripción generada automáticamente

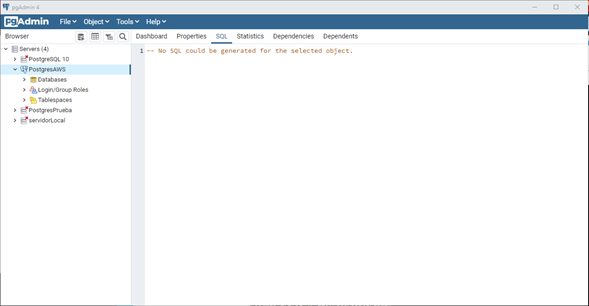
# Conexión cliente

Se hace conexión al servidor por medio de PgAdmin4 ingresando host, puerto, usuario y contraseña de la maquina física y del contenedor.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Se puede evidenciar la conexión con el servidor.



# Docker compose

Se crea un archivo llamado obligatoriamente “Docker-compose.yml” en cualquier editor de texto.

Texto

Descripción generada automáticamente

## Services:

En esta parte del archivo .yml, se dispondrá de datos tale como la imagen, en este caso llamada postgres; el restart con parámetros always que ayuda en caso de fallo al ejecutar el archivo .yml se reinicie siempre.

En la parte enviroment se dispondrá de la contraseña del postgres, en este caso la contraseña corresponde a example.

En la parte ports se dispondrá de los puertos exclusivos del postgres 5432:5432.

Y en la parte de volumes se asignará la ruta del volumen del contenedor que se creará a partir del Docker compose.

## Adminer:

En esta parte del archivo .yml se dispondrá de los parámetros de la creación de la herramienta de administración “Adminer” tales como el nombre de la herramienta y los puertos que corresponde a 8080.

# Build up:

Este comando se usa par inicializar el contenedor por medio del archivo de Docker compose creado previamente.

Texto

Descripción generada automáticamente

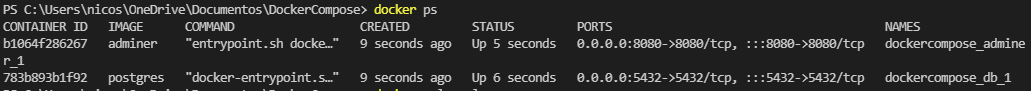
Aquí se evidencia la compilación del archivo Docker compose y adicionalmente se evidencia la creación del contenedor de postgres y del adminer.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Usamos el comando Docker ps para verificar que se están ejecutando los contenedores de postgres y adminer.

# docker ps



Hacemos uso del Docker volume ls para listar los volúmenes de los contenedores que se encuentran en la maquina física.

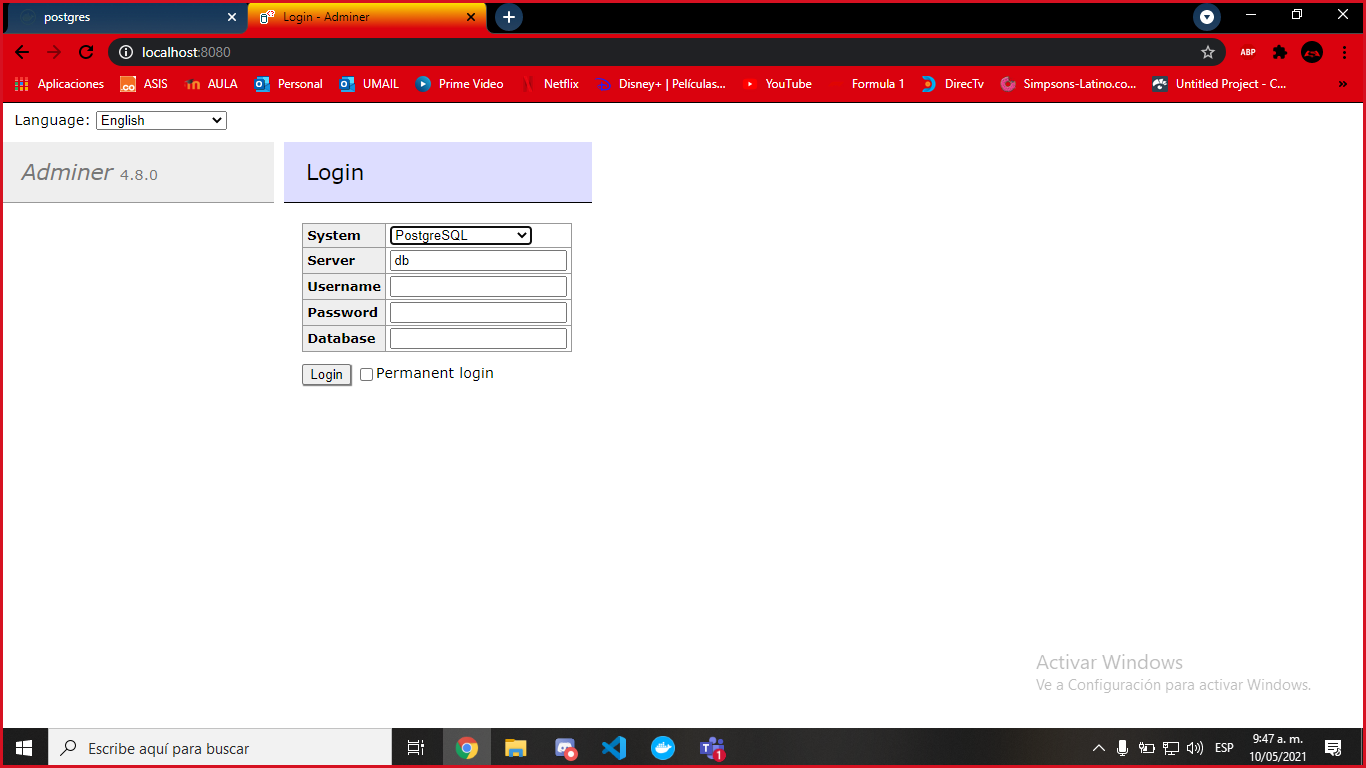
# docker volume ls

Texto

Descripción generada automáticamente

# Confirmación de funcionamiento de Adminer

Para verificar el funcionamiento del contenedor, se debe ir al navegador y escribir la siguiente dirección <http://localhost:8080> , si no hubo problemas en la ejecución del archivo .yml deberá aparecer la página.



Se debe ingresar los parámetros correspondientes al usuario y contraseña del archivo .yml y así poder entrar al home de la herramienta de administración

