**Steadylearner**

Posted on Jun 23, 2021 • Updated on Jun 4

How to set up Postgresql and Pgadmin with Docker

#postgres #braziliandevs #docker tutorial

In this post, we will learn how to install Postgresql and Pgadmin with Docker. If you are not familiar with it yet, please read [How to use Docker commands](#) first.

[You can see the Spanish version of this post here.](#)

[You can contact me with Telegram if you need to hire a full stack blockchain dev.](#)

[You can also join the Telegram group where I maintain and you can find other blockchain developers, recruiters, project owners, ask questions and network.](#)

If you use Linux relevant system, type `$docker` in your console and it would show how to install Docker.



If you use Mac or Windows, [it could be better to install the Docker desktop](#). Then, you can set it up to use minimal resource optionally.

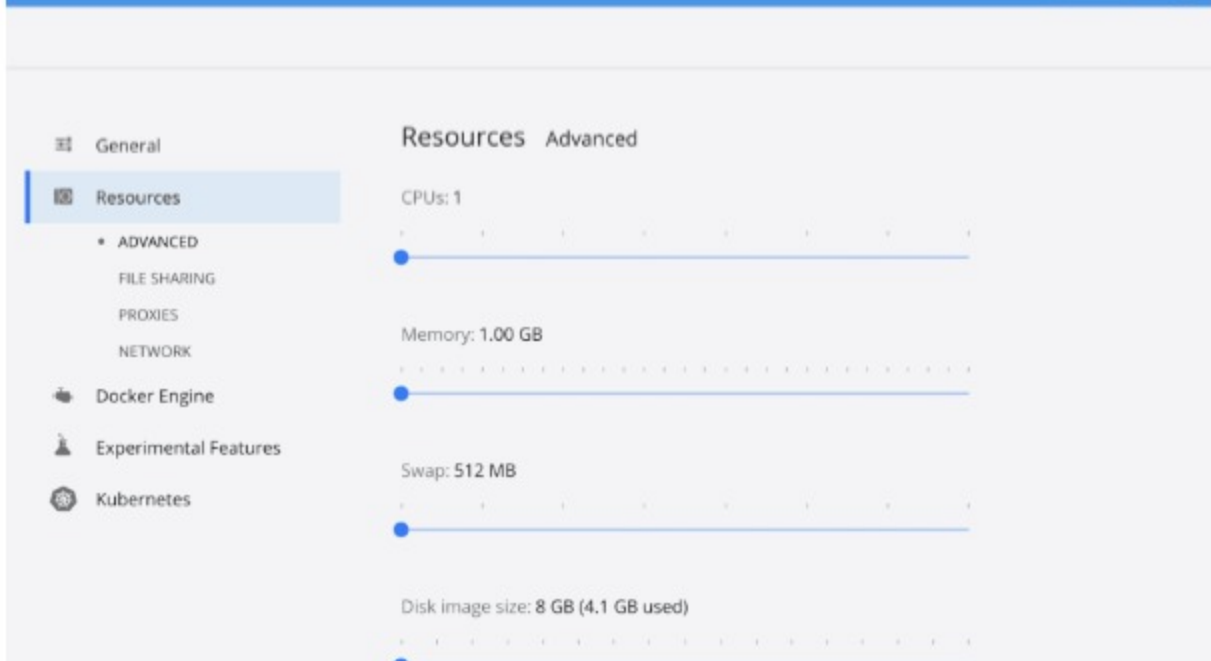


Table of contents

1. Install Postgresql and Pgadmin in Docker
2. Link them with Docker network
3. Set up Pgadmin and confirm connection
4. Conclusion

1. Install Postgresql and Pgadmin with Docker

Before we set up Postgresql with Docker, we will first make its data to persist in your machine with [volume](#).

```
$docker volume create postgresqldata
```

Then, you can see it worked with `$docker volume ls` command.

DRIVER	VOLUME NAME
local	postgresqldata

For volume to save your Postgresql data is ready, we will install it with Docker and link it to the volume we made with the command below.

```
$docker run -d -v postgresqldata:/data/db -e POSTGRES_PASSWORD=postgres
```

You can use your own password if you want after you complete this post.

When install is done, you will see the message similar to this in your console.

```
Status: Downloaded newer image for postgres:latest
```

Then, use `$docker ps -a` to see it is saved ok.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
postgres	"docker-entrypoint.s..."	0.0.0.0:5432->5432/tcp, :::5432->5432/tcp		Up

Use `$docker exec -it postgres /bin/bash` to use the console of the Docker image and then `$psql -h localhost -U postgres` to use psql commands in your console.

You won't need this a lot if you install pgadmin later but you can change your password with `\password` later if you want after you complete this post.

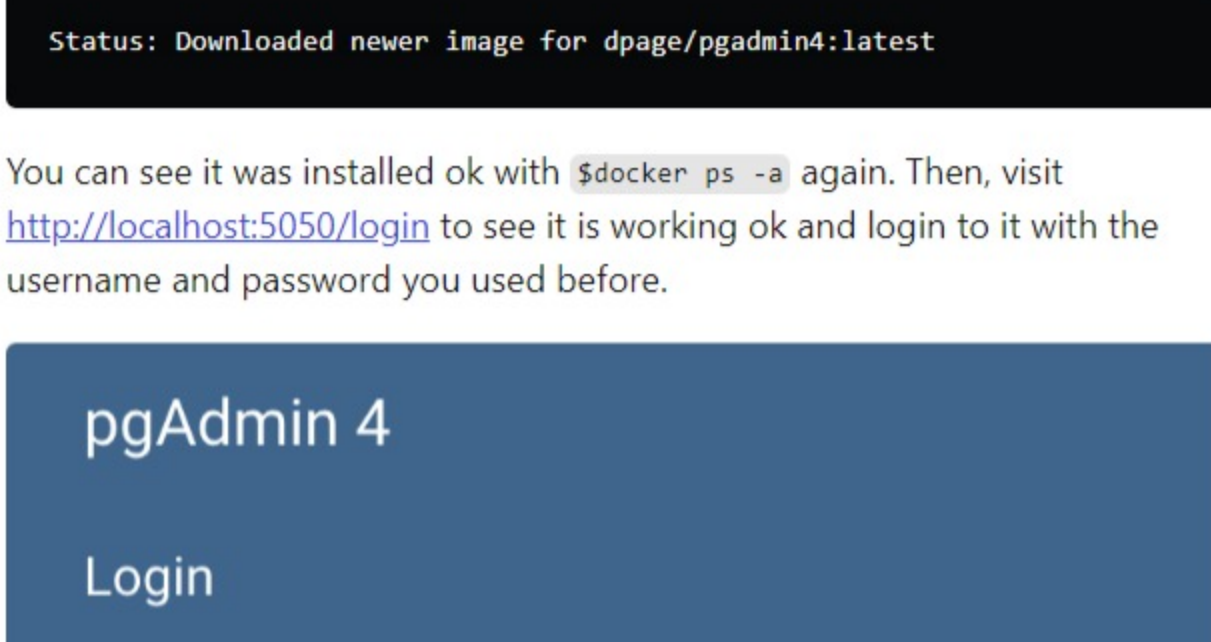
We verified that Postgresql was installed ok with Docker. We will use another console to install Pgadmin to manage it easily with the command below.

```
$docker run --name pgadmin -e "PGADMIN_DEFAULT_EMAIL=name@example.com"
```

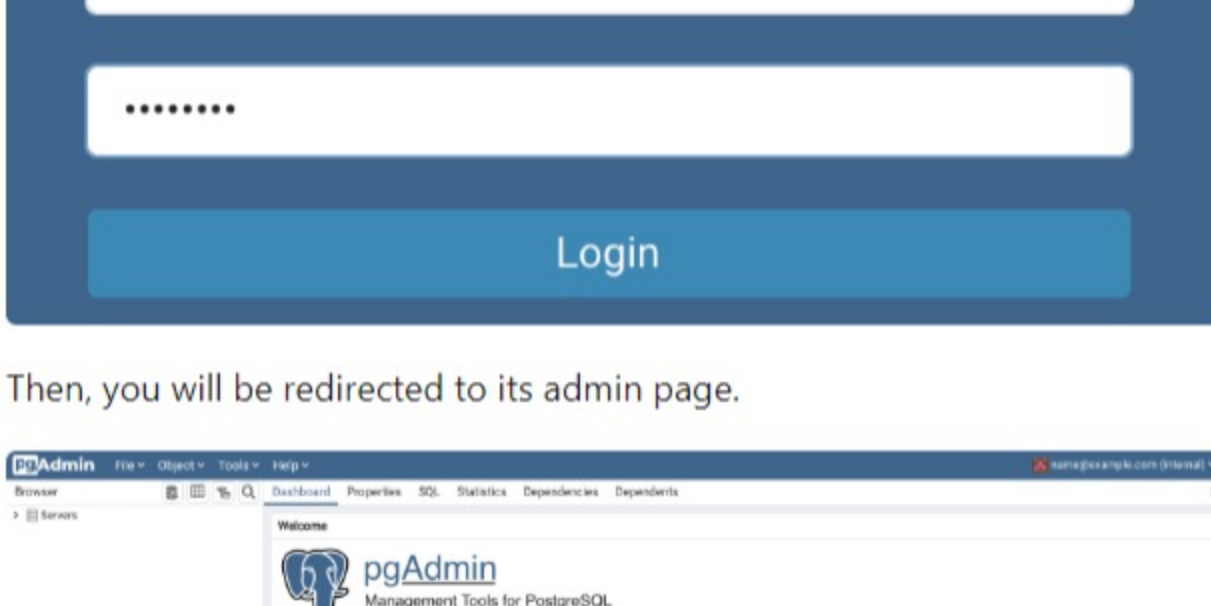
Then, you will see the message similar to this.

```
Status: Downloaded newer image for dpage/pgadmin4:latest
```

You can see it was installed ok with `$docker ps -a` again. Then, visit <http://localhost:5050/login> to see it is working ok and login to it with the username and password you used before.



Then, you will be redirected to its admin page.



You can set it while you [refer to this](#) but it will not work yet.

2. Link them with Docker network and set up Pgadmin

To make Pgadmin installed with Docker work along with Postgesql, we should link them with the Docker network command.

We will first create pgnetwork for that purpose with the command below.

```
$docker network create --driver bridge pgnetwork
```

Then, you can see it was created with `$docker network ls` command.

NETWORK ID	NAME	DRIVER	SCOPE
	pgnetwork	bridge	local

You can link Pgadmin and Postgresql to it with these.

```
$docker network connect pgnetwork pgadmin  
$docker network connect pgnetwork postgres
```

You can verify they were connected with pgnetwork with `docker network inspect pgnetwork` command.

```
[  
  {  
    "Name": "pgnetwork",  
    {  
      "Name": "pgadmin",  
      "Name": "postgres",  
    },  
  },  
]  
]
```

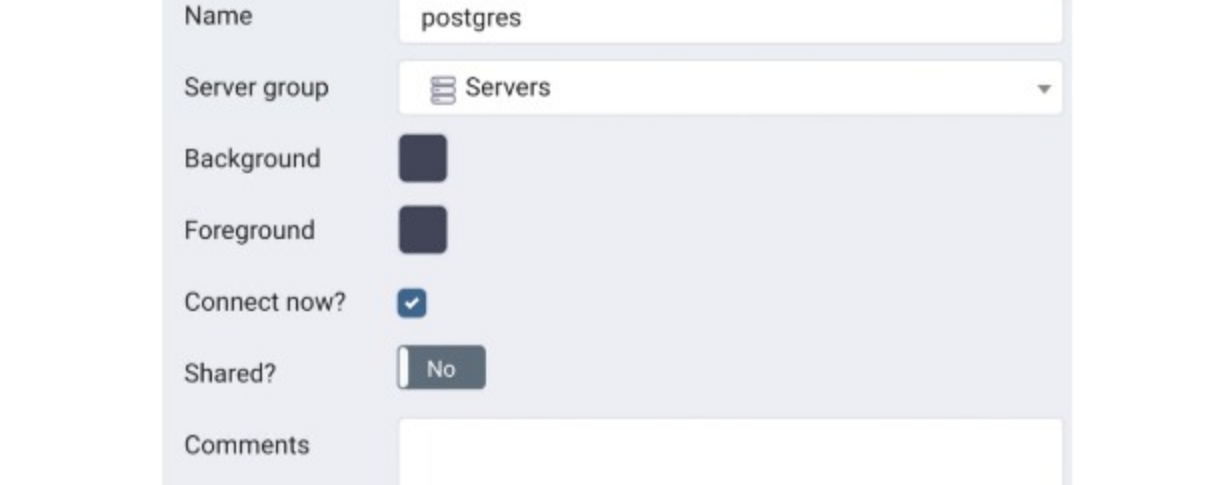
If you could make it here, everything is ready to make your Pgadmin with Postgresql with Docker.

3. Set up Pgadmin and confirm connection

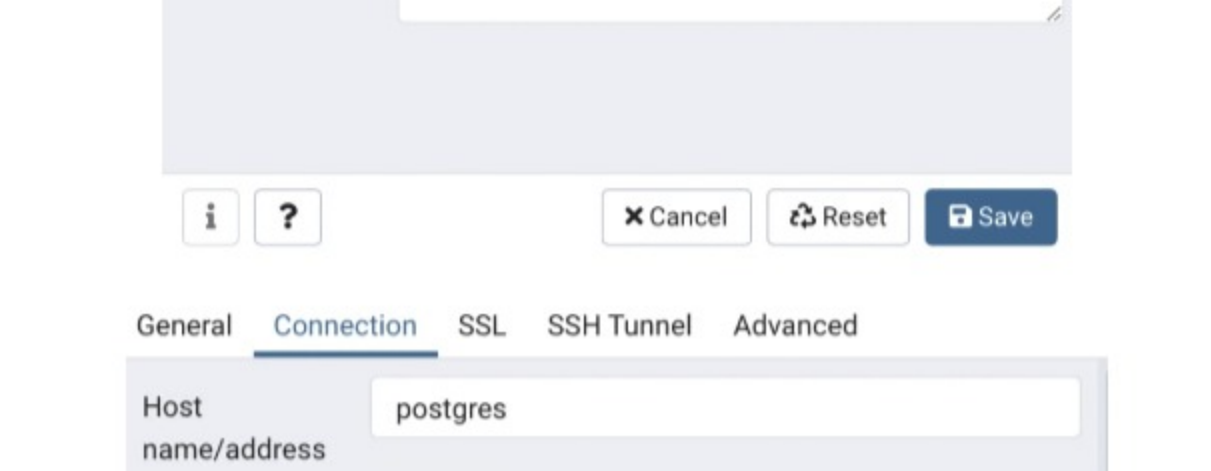
Because we linked the Pgadmin and Postgresql installed with Docker in the same network, you can finally configure Pgadmin to connect with Postgresql.

Click new server button in [the admin browser page](#) first.

Then, use **postgres** for every necessary fields if you haven't changed anything yet.



Then, click save button and you will see the connection is finally made.



Hope you could make it. [You can search more about how to use it at its website.](#)

You can stop them after you test them not to use your resources with this.

```
$docker stop pgadmin postgres
```

If you want to use it with Python, these commands can be helpful.

```
# Intel  
# $pip install psycopg2 or  
  
# M1  
# $brew install postgresql, $brew link openssl  
# export CPPFLAGS="-I/opt/homebrew/opt/openssl@1.1/include -I/opt/homebr  
# export PATH="/opt/homebrew/opt/postgresql@13/bin:$PATH  
# export LDLAGS="-L/opt/homebrew/opt/openssl@1.1/lib"  
# export CPPFLAGS="-I/opt/homebrew/opt/openssl@1.1/include"  
# export PKG_CONFIG_PATH="/opt/homebrew/opt/openssl@1.1/lib/pkgconfig"  
# pip install psycopg2-binary
```

4. Conclusion

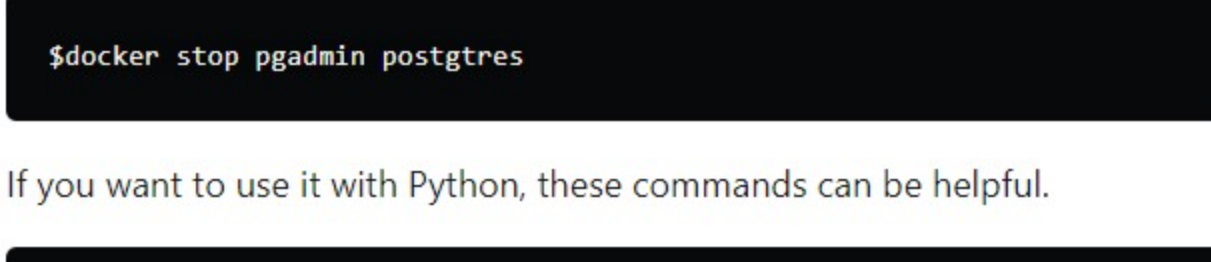
In this post, we learnt how to set up Postgresql and Pgadmin with Docker. You will be able to use it with a remote Postgresql instance also.

If you could make it work, you will not need to install Postgresql and pgadmin locally anymore.

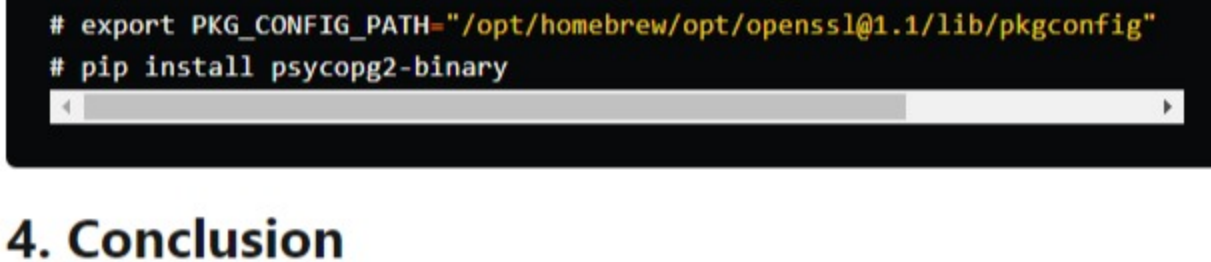
[If you need to hire a developer, you can contact me.](#)

[I can write a full stack dapp.](#)

Thanks.



[Code of Conduct](#) - [Report abuse](#)



DEV Community — A constructive and inclusive social network for software developers. With you every step of your journey.

Built on [Forum](#) — the open source software that powers DEV and other inclusive communities.

Made with love and Ruby on Rails. DEV Community © 2016 - 2022.

