

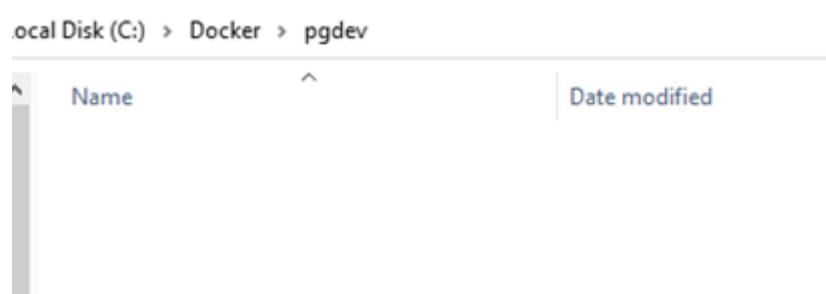
Creating a PostgreSQL Docker Container with a Volume on Windows



voiceofthedba.com/2023/01/09/creating-a-postgresql-docker-container-with-a-volume-on-windows

View all posts by way0utwest →

January 9, 2023



This post looks at how to set up a [PostgreSQL](#) container on Windows using Docker for Windows. I've seen a few posts, but I had to cobble together some instructions from places, so I decided to make my own post to help me remember and keep things simple.

tl;dr: do this:

- Create a folder `c:\docker\pgdev`
- get the Docker image: `docker pull postgresql:latest`
- Run the container, command below:

```
docker run --name pgdev -e POSTGRES_PASSWORD=Str0ngP@ssword -d -p 5432:5432 -v C:\Docker\pgdev:/var/lib/postgresql/data postgres
```

That's it. Then you have a PostgreSQL instance running on port 5432 (default) with a user, postgresql, and a password, [Str0ngP@ssword](#).

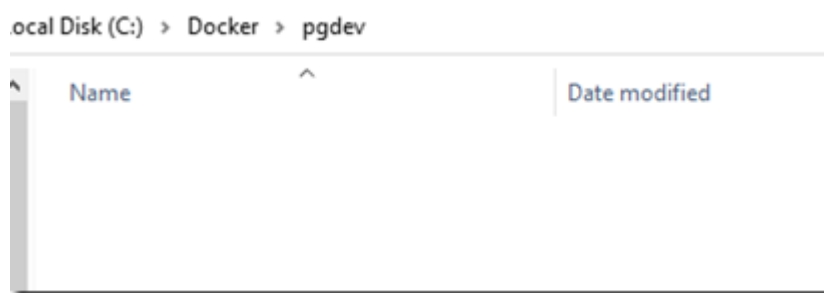
More detailed instructions below

Create a place for data

Containers are ephemeral, which isn't what we want for a database. We want to keep data around, so let's make a place for this. This will be a volume for our container, which we will map to a particular location inside the container. Then if the container dies, we can map this to another PostgreSQL container and have our data appear.

Create a `c:\Docker` folder on your machine. This is a good spot for any Docker related volumes.

Now create a `pgdev` folder under `c:\docker`. This is the place we'll keep data for this PostgreSQL container. It should be empty.

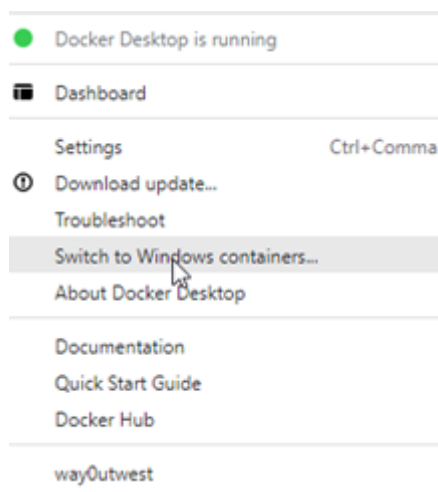


Get the Image

Container images are available from Docker. I won't cover installing Docker or setting up Linux containers, but you do need to do this. I used Windows 10, and I have WSL v2, as you can see:

```
steve@ARISTOTLE C:\Users\Steve
wsl --list --verbose
NAME                STATE      VERSION
Ubuntu-18.04        Stopped    1
docker-desktop       Running    2
docker-desktop-data  Running    2
```

Docker Desktop is running Linux containers. You can see that since it say "Switch to Windows containers".



Next, use Docker Pull. I assume you are working with the default Docker registry, so this command should work:

```
Docker image pull postgres:latest
```

This will start downloading the image.

```
Steve@ARISTOTLE C:\Users\Steve
$ docker image pull postgres:latest
latest: Pulling from library/postgres
3f4ca61aafcd: Downloading [====>
048d3078d446: Downloading [=====>
c6d23b4fe6c1: Download complete
d846f6946dd5: Download complete
76f7157f330d: Downloading [==>
4eacfb0464b2: Waiting
5c197e2b597b: Waiting
2c4576649951: Waiting
1ee267d33d50: Waiting
```

When this completes, move on.

Starting the container

Once we have the image, we can start the container. You could do this without the folder above, but your data would be in the container and if the container were ever deleted, then the data is lost.

The basic command for starting the container requires a few parameters. Here is a list of what I provided in the command at the beginning:

- `--name` – This is a name you can use in docker commands to refer to the container. You can put anything. I chose “pgdev”.
- The password in the database system for the postgres user. This is a default user and you send this in as an environment variable with `-e`. The password I used here is: [Str0ngP@ssword](#)
- `-d` runs this detached, rather than interactively. This means your command shell can return command to you. Otherwise, all output from the container appears in the shell and you can’t type anything.
- `-p` is the port mapping. This is host:container. In this case, we map 5432 on the host (where we use some postgresSQL driver to connect) to 5432 inside the container. You can choose any unused port for the first number, but 5432 is needed for the second number as the postgresql service is listening on 5432.
- `-v` is the volume mapping. Here we map our host folder to a container folder (host:container). We enter the folder we created above and then map this to the place where postgresql stores data. That’s in `/var/lib/postgresql/data`
- `postgres` is the image name.

Here is the command again:

```
docker run --name pgdev -e POSTGRES_PASSWORD=Str0ngP@ssword -d -p 5432:5432 -v
C:\Docker\pgdev:/var/lib/postgresql/data postgres
```

Once this executes, we should see a long hex code returned, which is the container identifier.

```
Steve@ARISTOTLE C:\Users\Steve
└─ docker run --name pgdev1 -e POSTGRES_PASSWORD=Str0ngP@ssw0rd -d -p 5432:5432 -v C:\Docker\pgdev1\var\lib\postgresql\data postgres
a7bd3442763c3cfc4e5d1ff4b58702cc858a25974aefd6af6a2de2585109fc79
```

We also see our folder is now filled with postgresql specific data files and folders:

The screenshot shows a Windows File Explorer window titled "cd C:\> - Docker - pgsql". It displays a list of files and folders in the "pgsql" directory. The columns are "Name", "Date modified", "Type", and "Size". The file "pg_makedir" is selected.

Name	Date modified	Type	Size
base	12/27/2022 10:22 AM	File folder	
global	12/27/2022 10:22 AM	File folder	
pg_commit_ts	12/27/2022 10:22 AM	File folder	
pg_dynshmem	12/27/2022 10:22 AM	File folder	
pg_inetutils	12/27/2022 10:22 AM	File folder	
pg_makedir	12/27/2022 10:22 AM	File folder	
pg_notify	12/27/2022 10:22 AM	File folder	
pg_replinit	12/27/2022 10:22 AM	File folder	
pg_serial	12/27/2022 10:22 AM	File folder	
pg_snapshots	12/27/2022 10:22 AM	File folder	
pg_stat	12/27/2022 10:22 AM	File folder	
pg_stat_tmp	12/27/2022 10:22 AM	File folder	
pg_subtrans	12/27/2022 10:22 AM	File folder	
pg_tuptojson	12/27/2022 10:22 AM	File folder	
pg_wal	12/27/2022 10:22 AM	File folder	
pg_waldir	12/27/2022 10:22 AM	File folder	
pg_xact	12/27/2022 10:22 AM	File folder	
pg_hba.conf	12/27/2022 10:22 AM	CONF File	5 KB
pg_ident.conf	12/27/2022 10:22 AM	CONF File	2 KB
PG_VERSION	12/27/2022 10:22 AM	File	1 KB
postgresql.auto.conf	12/27/2022 10:22 AM	CONF File	1 KB
postgresql.conf	12/27/2022 10:22 AM	CONF File	26 KB
postgresql.opts	12/27/2022 10:22 AM	OPTS File	1 KB
postgresql.sysconf	12/27/2022 10:22 AM	PID File	1 KB

That's it, our container is running.