

PostgreSQL and TimeScaleDB Setup

In my case, I decided to install and run Postgres and TimescaleDB in a Docker container. This is mainly because of the ease of setup, isolation and consistency across environments.

Prerequisites

Make sure Docker is installed and running on your computer.

Step 1: Create Docker Image for TimescaleDB with PostgreSQL 12

Create a directory to store Docker volumes

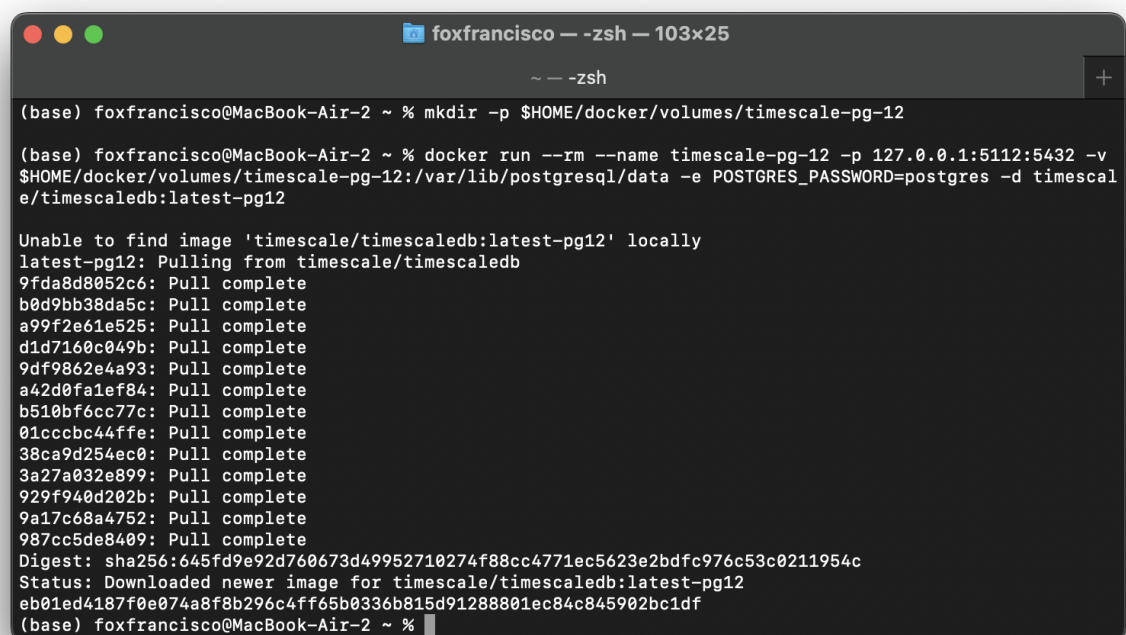
```
mkdir -p $HOME/docker/volumes/timescale-pg-12
```

Run the Docker container for TimescaleDB with PostgreSQL 12

```
docker run --rm --name timescale-pg-12 -p 127.0.0.1:5112:5432 -v  
$HOME/docker/volumes/timescale-pg-12:/var/lib/postgresql/data -e  
POSTGRES_PASSWORD=postgres -d timescale/timescaledb:latest-pg12
```

This command does the following:

- `--rm`: Automatically removes the container when it stops.
- `--name timescale-pg-12`: Assigns the container the name 'timescale-pg-12'.
- `-p 127.0.0.1:5112:5432`: Maps the local port 5112 to the container's PostgreSQL port 5432.
- `-v $HOME/docker/volumes/timescale-pg-12:/var/lib/postgresql/data`: Mounts a local volume for persistent data storage.
- `-e POSTGRES_PASSWORD=postgres`: Sets the PostgreSQL password to 'postgres'.
- `-d timescale/timescaledb:latest-pg12`: Runs the Docker container in detached mode using the latest TimescaleDB with PostgreSQL 12 image.



```
foxfrancisco — zsh — 103x25  
~ — zsh  
(base) foxfrancisco@MacBook-Air-2 ~ % mkdir -p $HOME/docker/volumes/timescale-pg-12  
(base) foxfrancisco@MacBook-Air-2 ~ % docker run --rm --name timescale-pg-12 -p 127.0.0.1:5112:5432 -v  
$HOME/docker/volumes/timescale-pg-12:/var/lib/postgresql/data -e POSTGRES_PASSWORD=postgres -d timescal  
e/timescaledb:latest-pg12  
  
Unable to find image 'timescale/timescaledb:latest-pg12' locally  
latest-pg12: Pulling from timescale/timescaledb  
9fda8d8052c6: Pull complete  
b0d9bb38da5c: Pull complete  
a99f2e61e525: Pull complete  
d1d7160c049b: Pull complete  
9df9862e4a93: Pull complete  
a42d0fa1ef84: Pull complete  
b510bf6cc77c: Pull complete  
01cccbc44ffe: Pull complete  
38ca9d254ec0: Pull complete  
3a27a032e899: Pull complete  
929f940d202b: Pull complete  
9a17c68a4752: Pull complete  
987cc5de8409: Pull complete  
Digest: sha256:645fd9e92d760673d49952710274f88cc4771ec5623e2bdfc976c53c0211954c  
Status: Downloaded newer image for timescale/timescaledb:latest-pg12  
eb01ed4187f0e074a8f8b296c4ff65b0336b815d91288801ec84c845902bc1df  
(base) foxfrancisco@MacBook-Air-2 ~ %
```

Step 2: Verify Container Status

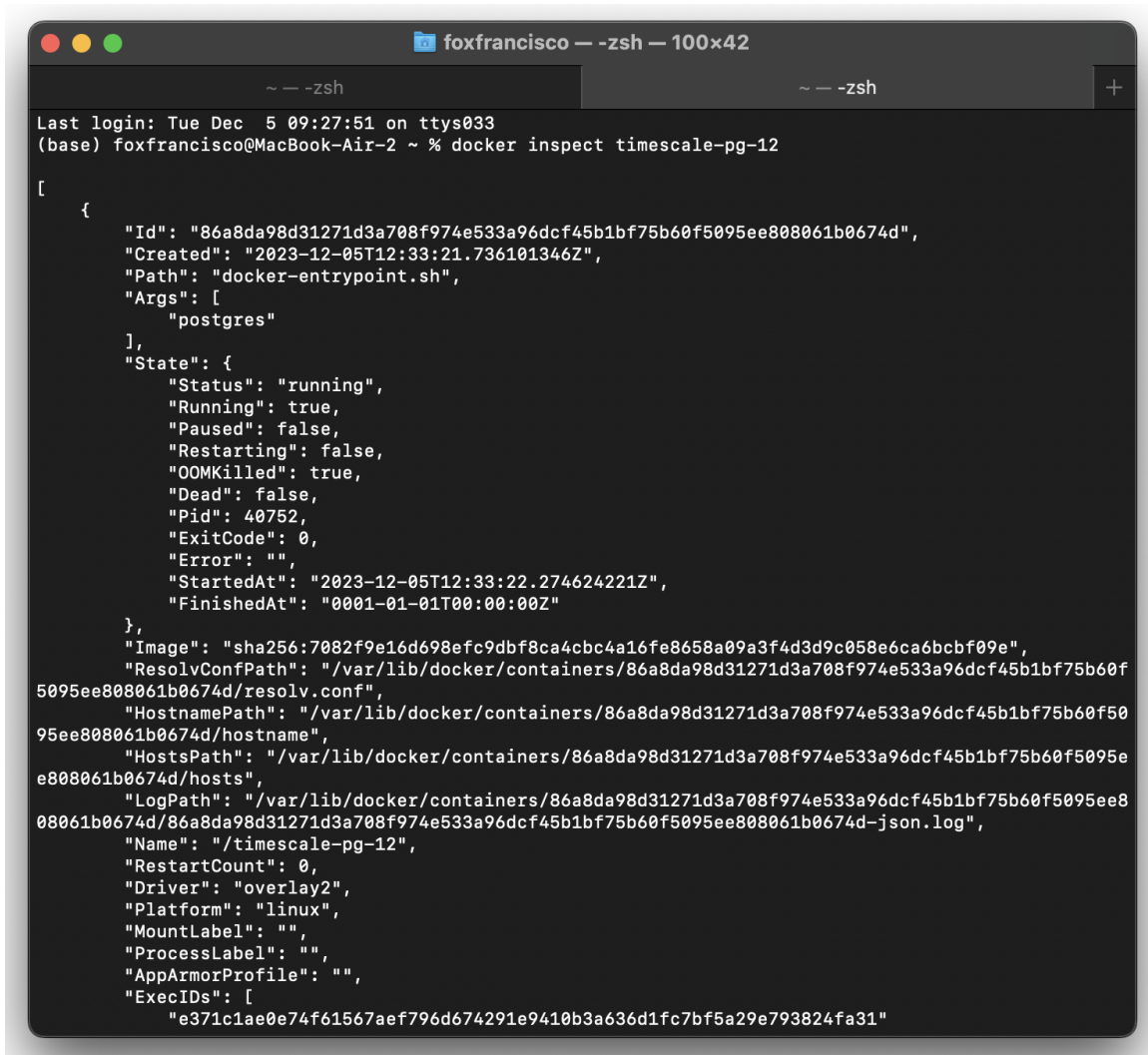
Check if the container is running:

```
docker ps
```

You should see the 'timescale-pg-12' container listed.

View container details:

```
docker inspect timescale-pg-12
```

A terminal window titled 'foxfrancisco --zsh-- 100x42' showing the output of the command 'docker inspect timescale-pg-12'. The output is a JSON object representing the container's configuration and state. The 'Status' is 'running', and the 'Pid' is 40752. The 'Image' is 'sha256:7082f9e16d698efc9dbf8ca4cbc4a16fe8658a09a3f4d3d9c058e6ca6bcbf09e'. The 'Name' is '/timescale-pg-12'. The 'ExecIDs' array contains one ID: 'e371c1ae0e74f61567aef796d674291e9410b3a636d1fc7bf5a29e793824fa31'.

```
foxfrancisco --zsh-- 100x42
~ --zsh
Last login: Tue Dec  5 09:27:51 on ttys033
(base) foxfrancisco@MacBook-Air-2 ~ % docker inspect timescale-pg-12

[
  {
    "Id": "86a8da98d31271d3a708f974e533a96dcf45b1bf75b60f5095ee808061b0674d",
    "Created": "2023-12-05T12:33:21.736101346Z",
    "Path": "docker-entrypoint.sh",
    "Args": [
      "postgres"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": true,
      "Dead": false,
      "Pid": 40752,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2023-12-05T12:33:22.274624221Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:7082f9e16d698efc9dbf8ca4cbc4a16fe8658a09a3f4d3d9c058e6ca6bcbf09e",
    "ResolvConfPath": "/var/lib/docker/containers/86a8da98d31271d3a708f974e533a96dcf45b1bf75b60f5095ee808061b0674d/resolv.conf",
    "HostnamePath": "/var/lib/docker/containers/86a8da98d31271d3a708f974e533a96dcf45b1bf75b60f5095ee808061b0674d/hostname",
    "HostsPath": "/var/lib/docker/containers/86a8da98d31271d3a708f974e533a96dcf45b1bf75b60f5095ee808061b0674d/hosts",
    "LogPath": "/var/lib/docker/containers/86a8da98d31271d3a708f974e533a96dcf45b1bf75b60f5095ee808061b0674d/86a8da98d31271d3a708f974e533a96dcf45b1bf75b60f5095ee808061b0674d-86a8da98d31271d3a708f974e533a96dcf45b1bf75b60f5095ee808061b0674d-json.log",
    "Name": "/timescale-pg-12",
    "RestartCount": 0,
    "Driver": "overlay2",
    "Platform": "linux",
    "MountLabel": "",
    "ProcessLabel": "",
    "AppArmorProfile": "",
    "ExecIDs": [
      "e371c1ae0e74f61567aef796d674291e9410b3a636d1fc7bf5a29e793824fa31"
    ]
  }
]
```

Step 3: Access PostgreSQL and TimescaleDB

Connect to the PostgreSQL instance inside the container:

```
docker exec -it timescale-pg-12 psql -U postgres
```

This opens a PostgreSQL shell. You can now run SQL queries or set up your database.

Step 4: Stop and Remove the Container

To stop the container:

```
docker stop timescale-pg-12
```