PostgresSQL 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

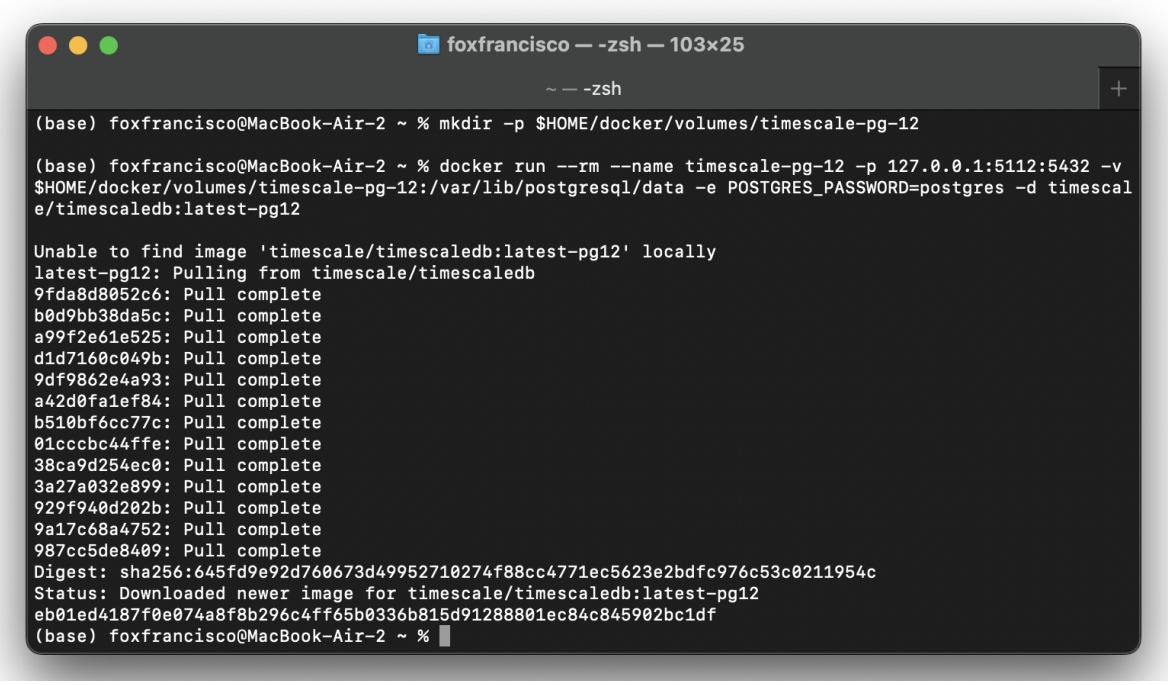
```
mkdir -p $HOME/docker/volumes/timescale-pg-12
```

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:

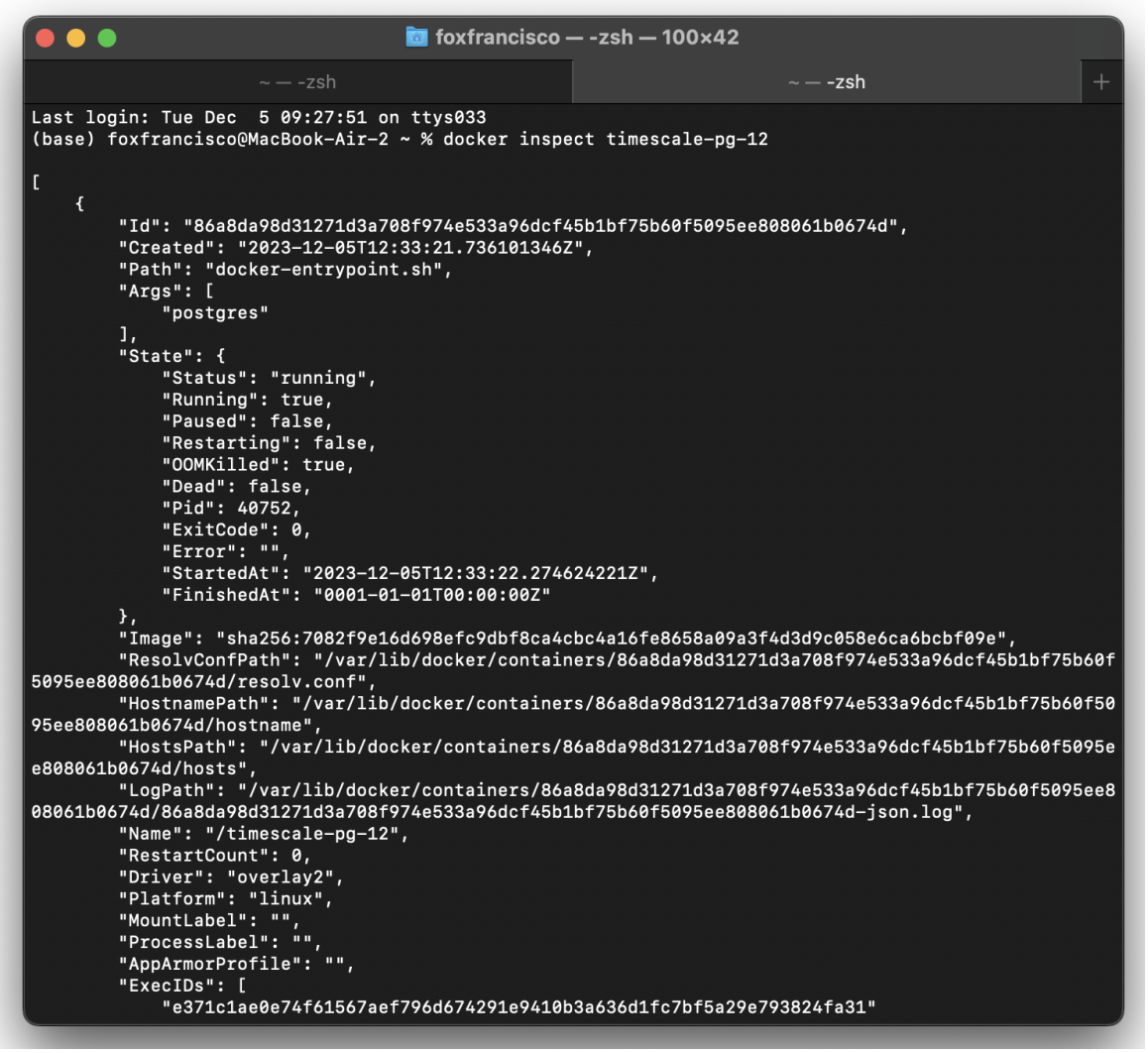
- `--rm`: Automatically removes the container when it stops.
- `--name timescale-pg-12`: Assigns the container the name 'timescale-pg-12'.
- `-p 127.0.0.1:5112:5432`: Maps the local port 5112 to the container's PostgreSQL port 5432.
- `-v $HOME/docker/volumes/timescale-pg-12:/var/lib/postgresql/data`: 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.



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`



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`