How to install PostgreSQL and PGadmin4 in docker

1. Download PGadmin4 Image from docker

docker run -p 8089:80 -e 'PGADMIN\_DEFAULT\_EMAIL=kawalkamboj91@gmail.com' -e 'PGADMIN\_DEFAULT\_PASSWORD=123' -d dpage/pgadmin4

1. Download docker image for Postgresql

docker run --name pg -d -e POSTGRES\_USER=lunar -e POSTGRES\_PASSWORD=123 postgres

1. run docker inspect to see ip of container

Most cases ip will be 172.17.0.2

1. Then we will enter inside postgresql container

docker exec -it pg bash

1. After this we will run below command to assign port and ip and enter into database

psql -U lunar -h 172.17.0.2 -p 5432

user will be lunar

password will 123

1. \l is to list database q
2. To create database Command is CREATE DATBASE “databasename”; without quote
3. \c is to connect th databse
4. To delete a database use command DROP DATBASE “name of database”; without quotes
5. Syntax for creating table #NOT NULL means it can’t be empty must have a value

test=# CREATE TABLE person (

test(# id BIGSERIAL NOT NULL PRIMARY KEY,

test(# first\_name VARCHAR(50) NOT NULL,

test(# last\_name VARCHAR(50) NOT NULL,

test(# gender VARCHAR(7) NOT NULL,

test(# date\_of\_birth DATE NOT NULL

test(# email VARCHAR (100) );

11 to see relation of database use command \d

12 to see the details of table use command \d followed by table name

For example here table name is person so command would be \d person

Syntax to insert values in table called person  
INSERT INTO person (first\_name, last\_name, gender, date\_of\_birth, email)

VALUES ('John', 'Jones', 'MALE', date '1990-01-09', 'john@gmail.com');

To view all the enteries in particular table called person  
SELECT \* FROM person;

To sort data, we can use Order By keyword. Suppose you want to sort data by ascending order of email use below command   
SELECT \* FROM person ORDER BY email

How to backup & restore PostgreSQL in docker container

docker exec <postgres\_container\_name> pg\_dump -U postgres <database\_name> > backup.sql

To restore data   
docker exec -i <postgres\_container\_name> psql -U postgres -d <database\_name> < backup.sql

Explanation of above scenario

I use the standard official postgres docker image from [Docker Hub](https://hub.docker.com/_/postgres).

This image comes with two handy tools (pg\_dump and psql) which let us take easy backups and restore them with equal ease.

To backup, we use the pg\_dump tool:

docker exec <postgres\_container\_name> pg\_dump -U postgres <database\_name> > backup.sql

comment – in my case my container name is pg and username is lunar

This would create a text file named backup.sql containing all the data and schema of your database. You can then import this data back into postgres using the psql tool:

docker exec -i <postgres\_container\_name> psql -U postgres -d <database\_name> < backup.sql

The -i flag is of particular importance here because the psql tool needs to be run interactively for it to be able to read from the backup.sql file.

docker run --name pg5 -l com.datadoghq.ad.check\_names='["postgres"]' -l com.datadoghq.ad.init\_configs='[{}]' -l com.datadoghq.ad.instances='["host":"%%pg5%%", "port":5432,"username":"datadog","password":"password"]' -d -e POSTGRES\_USER=lunar -e POSTGRES\_PASSWORD=123 postgres

LABEL "com.datadoghq.ad.check\_names"='[“postgres”]'

LABEL "com.datadoghq.ad.init\_configs"='[{}]'

LABEL "com.datadoghq.ad.instances"='["host":"%%172.17.0.5%%", "port":5432,"username":"datadog","password":"123"]'

docker run -l com.datadoghq.ad.check\_names='[“postgres”]' -l com.datadoghq.ad.init\_configs='[{}]' -l com.datadoghq.ad.instances='[["host":"%%pg3%%", "port":5432,"username":"datadog","password":"123"]'