

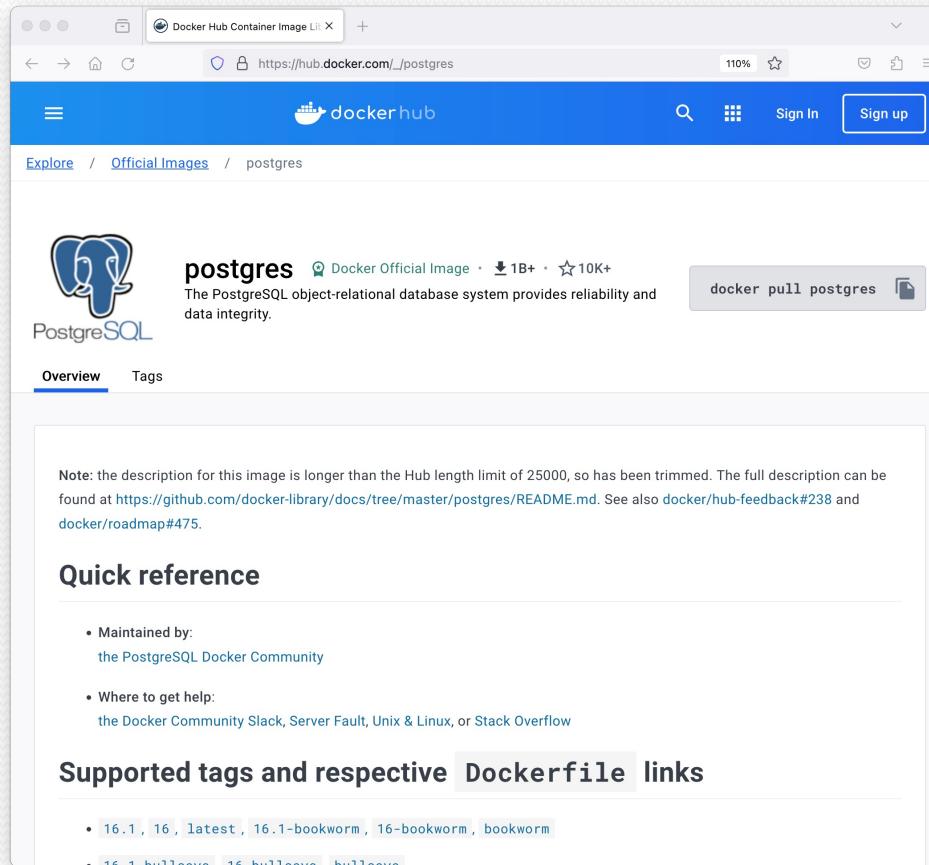
SDSU
Computer Science

Mr. Ken Gamradt

Docker PostgreSQL and pgAdmin

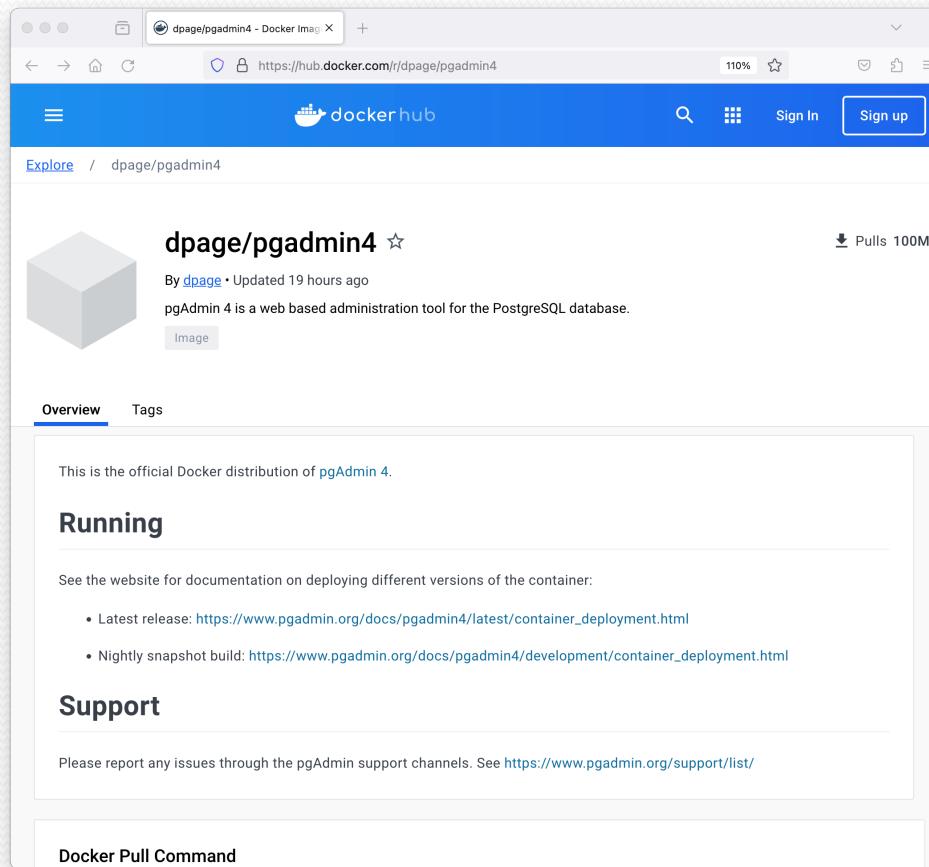
12-07-23

PostgreSQL – dockerhub



A screenshot of a web browser displaying the Docker Hub page for the PostgreSQL image. The URL in the address bar is https://hub.docker.com/_/postgres. The page features the Docker Hub header with a search bar, sign-in links, and a 'Sign up' button. Below the header, the navigation path is 'Explore / Official Images / postgres'. The main content area shows the PostgreSQL logo, the image name 'postgres', and its status as a 'Docker Official Image' with over 1B+ pulls and 10K+ stars. A note states: 'The PostgreSQL object-relational database system provides reliability and data integrity.' To the right is a button labeled 'docker pull postgres' with a clipboard icon. Below this, under 'Quick reference', there are two bullet points: 'Maintained by: the PostgreSQL Docker Community' and 'Where to get help: the Docker Community Slack, Server Fault, Unix & Linux, or Stack Overflow'. At the bottom, a section titled 'Supported tags and respective Dockerfile links' lists several tags: '16.1', '16', 'latest', '16.1-bookworm', '16-bookworm', 'bookworm', '16.1-helium', '16-helium', and 'helium'.

pgAdmin – dockerhub



Database Software

PostgreSQL

- [PostgreSQL](#) is a database management system (DBMS) that is used to store and manage data
- It is open-source, meaning that it is free to use and modify

pgAdmin

- [pgAdmin](#) is a free, open-source graphical tool for managing and administering PostgreSQL databases
- It allows you to easily create and maintain databases, run SQL queries, and view and edit data

Required Software

docker

- [Docker](#) is a tool designed to make it easier to create, deploy, and run applications
- It does this by using containers, which are essentially self-contained environments that contain all of the necessary parts for an application to run, such as libraries, dependencies, and runtime
- Docker helps to simplify the process of building and deploying applications

Install docker

- First, make sure you have Docker installed on your computer
- If you don't have it, you can download it from the Docker [website](#)

Installation

Run the docker compose file

- A docker compose file is a way to define and run multi-container Docker applications
- It's like a blueprint for your application, telling Docker what images to use and how to connect them together
- This will save time and hassle because you can set everything up in one place and start everything with a single command
- It's written in yaml format and has sections for defining the version and the different services that make up the application
- Each service can be customized with options like images, environment variables, volumes, ports, and links to other services

docker-compose.yaml (.yml) file

```
version: '3.9'
services:
  postgres:
    container_name: postgresql
    image: postgres
    hostname: localhost
    ports:
      - '5432:5432'
    environment:
      POSTGRES_USER: admin
      POSTGRES_PASSWORD: root
      POSTGRES_DB: test_db
    volumes:
      - postgres-data:/var/lib/postgresql/data
    restart: unless-stopped
```

```
pgadmin:
  container_name: pgadmin
  image: dpage/pgadmin4
  depends_on:
    - postgres
  ports:
    - '5050:80'
  environment:
    PGADMIN_DEFAULT_EMAIL: admin@admin.com
    PGADMIN_DEFAULT_PASSWORD: root
  volumes:
    - pgadmin-data:/var/lib/pgadmin
  restart: unless-stopped

volumes:
  postgres-data:
  pgadmin-data:
```

`docker-compose.yaml (.yml)` file

Version

- Specifies the version of the docker compose file format

Services

- Defines the services (i.e., containers) that make up the application
- In this case, we have two services: `postgres` and `pgadmin`

Images

- Specifies the docker image to be used for the container
- For the `postgres` service, we are using the `postgres` image, which is the official PostgreSQL image
- For the `pgadmin` service, we are using the `dpage/pgadmin4` image

docker-compose.yaml (.yml) file

Environments

- Sets environment variables for the container
- In the postgres service, we are setting the POSTGRES_USER, POSTGRES_PASSWORD, and POSTGRES_DB variables, which are used to configure the PostgreSQL database
- In the pgadmin service, we are setting the PGADMIN_DEFAULT_EMAIL and PGADMIN_DEFAULT_PASSWORD variables, which are used to set the default email and password for the pgAdmin login

Volumes

- Mounts a volume (i.e., persistent storage) for the container
- In the postgres service, we are mounting a volume named postgres_data at the /var/lib/postgresql/data directory, which is where the PostgreSQL data is stored
- This allows the data to persist even if the container is stopped or removed

docker-compose.yaml (.yml) file

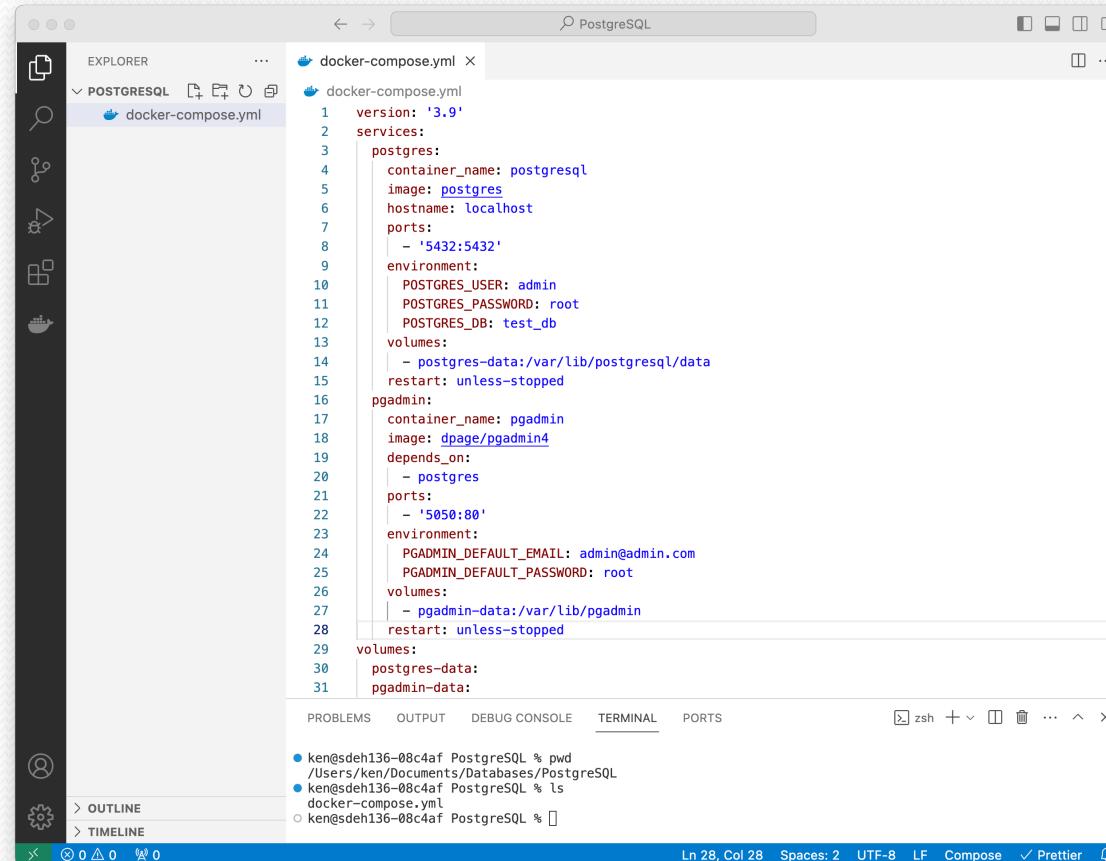
Ports

- Exposes a container's port to the host
- In the postgres service, we are exposing the default PostgreSQL port (5432) on the host
- In the pgadmin service, we are exposing port 80 on the host

Depends on

- Specifies that a service depends on another service
- In the pgadmin service, we are specifying that it depends on the postgres service, which means that the postgres service must be started before the pgadmin service

Build/Compose your database system containers



A screenshot of a code editor window titled "PostgreSQL". The editor shows a file named "docker-compose.yml" with the following content:

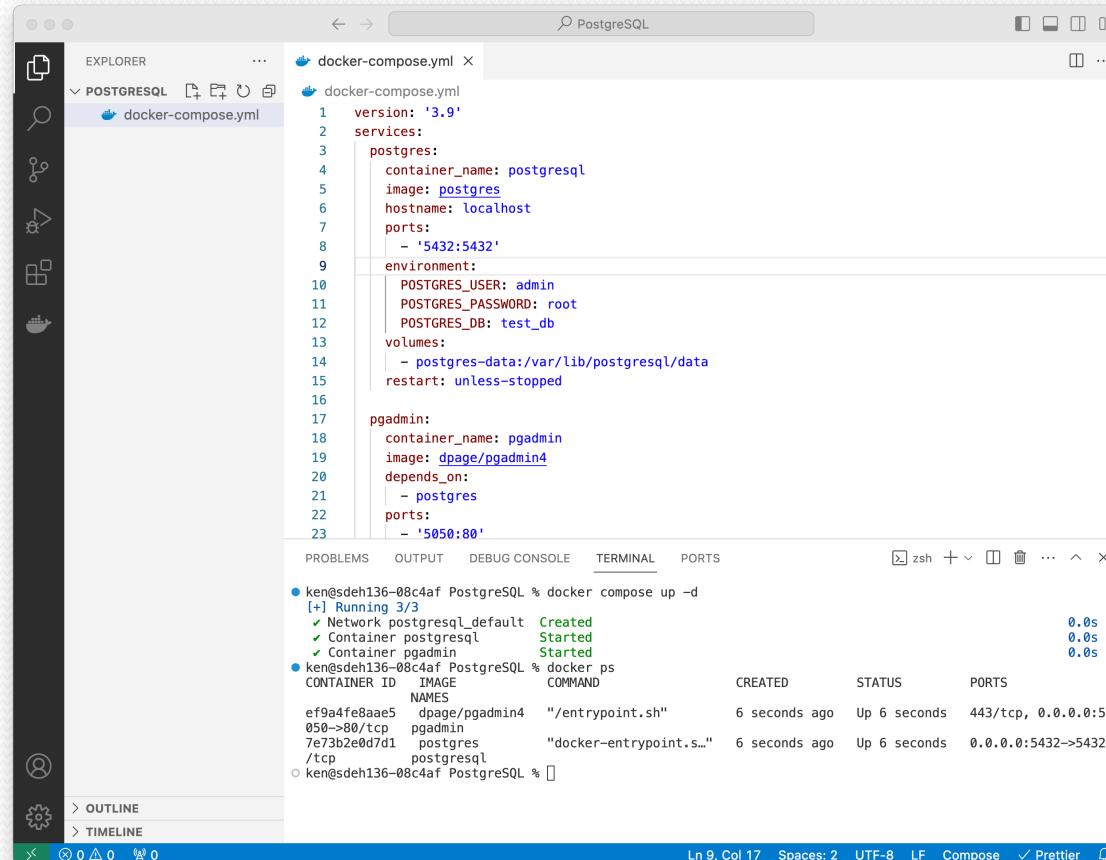
```
version: '3.9'
services:
  postgres:
    container_name: postgresql
    image: postgres
    hostname: localhost
    ports:
      - '5432:5432'
    environment:
      POSTGRES_USER: admin
      POSTGRES_PASSWORD: root
      POSTGRES_DB: test_db
    volumes:
      - postgres-data:/var/lib/postgresql/data
    restart: unless-stopped
  pgadmin:
    container_name: pgadmin
    image: dpage/pgadmin4
    depends_on:
      - postgres
    ports:
      - '5050:80'
    environment:
      PGADMIN_DEFAULT_EMAIL: admin@admin.com
      PGADMIN_DEFAULT_PASSWORD: root
    volumes:
      - pgadmin-data:/var/lib/pgadmin
    restart: unless-stopped
volumes:
  postgres-data:
  pgadmin-data:
```

The editor interface includes a sidebar with icons for file operations like copy, paste, and search. Below the editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing a terminal session with the following history:

- ken@sdeh136-08c4af PostgreSQL % pwd
/Users/ken/Documents/Database/PostgreSQL
- ken@sdeh136-08c4af PostgreSQL % ls
docker-compose.yml
- ken@sdeh136-08c4af PostgreSQL %

At the bottom, status indicators show 0 errors, 0 warnings, and 0 info messages. The status bar at the bottom right shows "Ln 28, Col 28" and other file metadata.

Build/Compose your database system containers



The screenshot shows a code editor window with a dark theme. On the left is a sidebar with icons for file operations like Open, Save, Find, and Refresh. The main area displays a file named `docker-compose.yml` under a folder named `POSTGRES`. The code in the file is:

```
version: '3.9'
services:
  postgres:
    container_name: postgresql
    image: postgres
    hostname: localhost
    ports:
      - '5432:5432'
    environment:
      POSTGRES_USER: admin
      POSTGRES_PASSWORD: root
      POSTGRES_DB: test_db
    volumes:
      - postgres-data:/var/lib/postgresql/data
    restart: unless-stopped
  pgadmin:
    container_name: pgadmin
    image: dpage/pgadmin4
    depends_on:
      - postgres
    ports:
      - '5050:80'
```

Below the code editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing the command `docker compose up -d` being run and its output:

```
[+] Running 3/3
  ✓ Network postgresql_default Created
  ✓ Container postgresql Started
  ✓ Container pgadmin Started
  ● ken@sdeh136-08c4af PostgreSQL % docker ps
  CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
  NAMES
  ef9a4fe8aae5 dpage/pgadmin4 "/entrypoint.sh" 6 seconds ago Up 6 seconds 443/tcp, 0.0.0.0:5
  050->80/tcp pgadmin
  7e73b2e0d7d1 postgres "docker-entrypoint.s..." 6 seconds ago Up 6 seconds 0.0.0.0:5432->5432
  /tcp postgresql
  ● ken@sdeh136-08c4af PostgreSQL %
```

At the bottom of the terminal output, there are status indicators: Ln 9, Col 17, Spaces: 2, UTF-8, LF, Compose, Prettier.

Start your database system containers

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS [zsh] zsh + ↻ □ 🗑 … ⌂ ⌃ ⌄

- ken@sdeh136-08c4af PostgreSQL % docker compose up -d
[+] Building 0.0s (0/0)
[+] Running 3/3
✓ Network postgresql_default Created 0.0s
✓ Container postgresql Started 0.0s
✓ Container pgadmin Started 0.0s docker:desktop-linux
- ken@sdeh136-08c4af PostgreSQL % docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
	NAMES				
803fcda77539	dpage/pgadmin4	"/entrypoint.sh"	22 seconds ago	Up 22 seconds	443/tcp, 0.0.0.0
:5050->80/tcp	pgadmin				
6157bc2cc3bd	postgres	"docker-entrypoint.s..."	22 seconds ago	Up 22 seconds	0.0.0.0:5432->54
32/tcp	postgresql				
- ken@sdeh136-08c4af PostgreSQL %

Shutdown your database system containers

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS □ zsh + ⌂ ⚡ ... ^ ×
```

- ken@sdeh136-08c4af PostgreSQL % docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
803fcda77539	dpage/pgadmin4	"/entrypoint.sh"	2 minutes ago	Up 2 minutes	443/tcp, 0.0.0.0:5
050->80/tcp	pgadmin				
6157bc2cc3bd	postgres	"docker-entrypoint.s..."	2 minutes ago	Up 2 minutes	0.0.0.0:5432->5432
/tcp	postgresql				

- ken@sdeh136-08c4af PostgreSQL % docker compose down
- [+] Running 3/3
 - ✓ Container pgadmin Removed 1.4s
 - ✓ Container postgresql Removed 0.1s
 - ✓ Network postgresql_default Removed 0.1s
- ken@sdeh136-08c4af PostgreSQL % docker ps

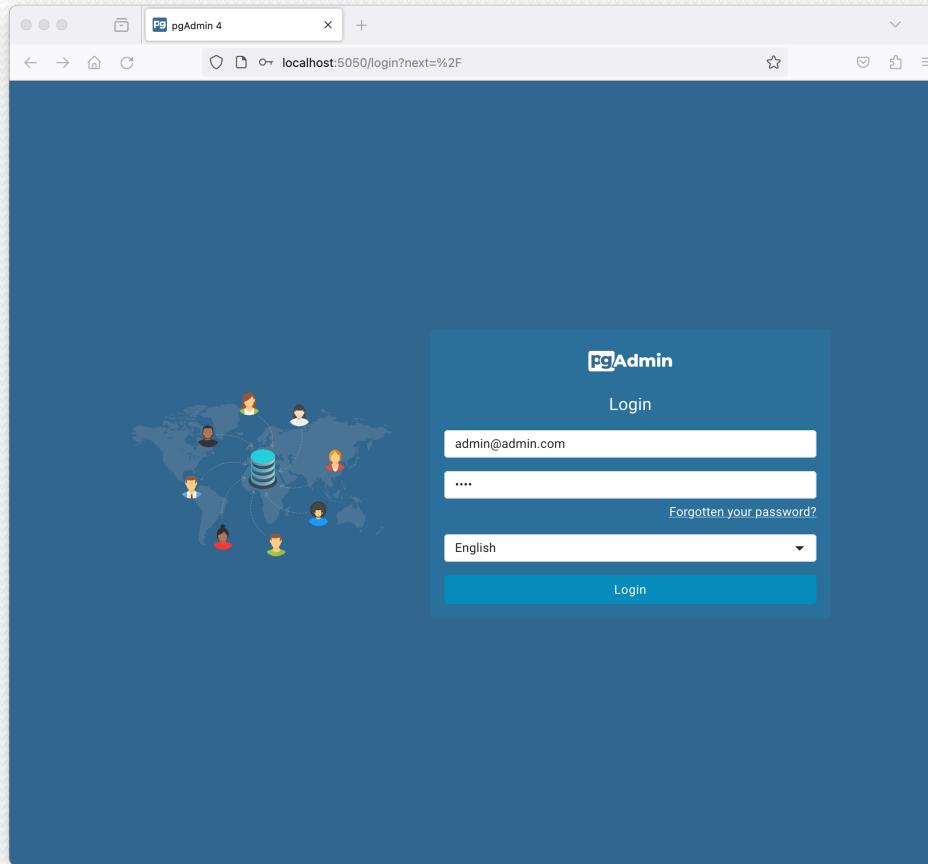
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ken@sdeh136-08c4af	PostgreSQL	%				

- ken@sdeh136-08c4af PostgreSQL %

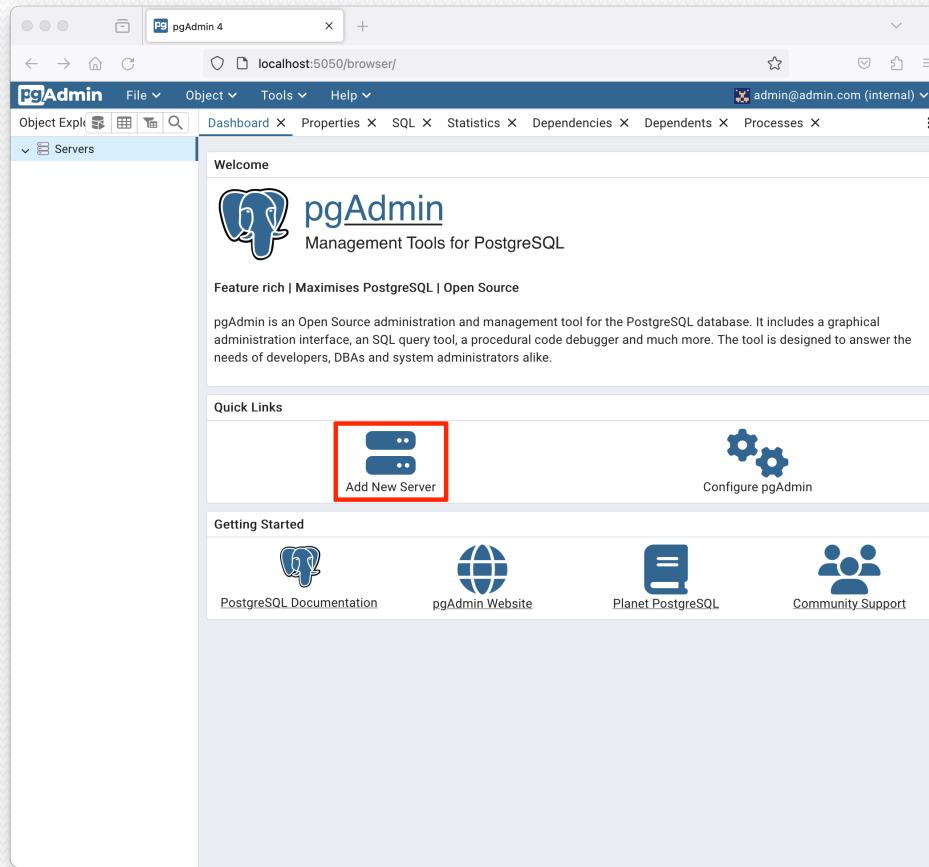
Access your database system containers – pgAdmin

```
ken@sdeh136-08c4af PostgreSQL % pwd
/Users/ken/Documents/Database/PostgreSQL
ken@sdeh136-08c4af PostgreSQL % ls
docker-compose.yml
ken@sdeh136-08c4af PostgreSQL % docker compose up -d
[+] Building 0.0s (0/0)                                            docker:desktop-linux
[+] Running 3/3
  ✓ Network postgresql_default  Created          0.0s
  ✓ Container postgresql      Started          0.0s
  ✓ Container pgadmin         Started          0.0s
ken@sdeh136-08c4af PostgreSQL % docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED        STATUS
              PORTS          NAMES
69b06426b203   dpage/pgadmin4  "/entrypoint.sh"    5 minutes ago  Up 5 minutes
nutes          443/tcp, 0.0.0.0:5050->80/tcp   pgadmin
95cc61d22356   postgres        "docker-entrypoint.s..."  5 minutes ago  Up 5 minutes
nutes          0.0.0.0:5432->5432/tcp     postgresql
ken@sdeh136-08c4af PostgreSQL %
```

Access your database system containers – pgAdmin



Access your database system – pgAdmin -> postgres



Access your database system – pgAdmin -> postgres

Register - Server

General Connection Parameters SSH Tunnel Advanced

Name	PostgreSQL
Server group	Servers
Background	X
Foreground	X
Connect now?	<input checked="" type="checkbox"/>
Shared?	<input type="checkbox"/>
Shared Username	
Comments	

Buttons:

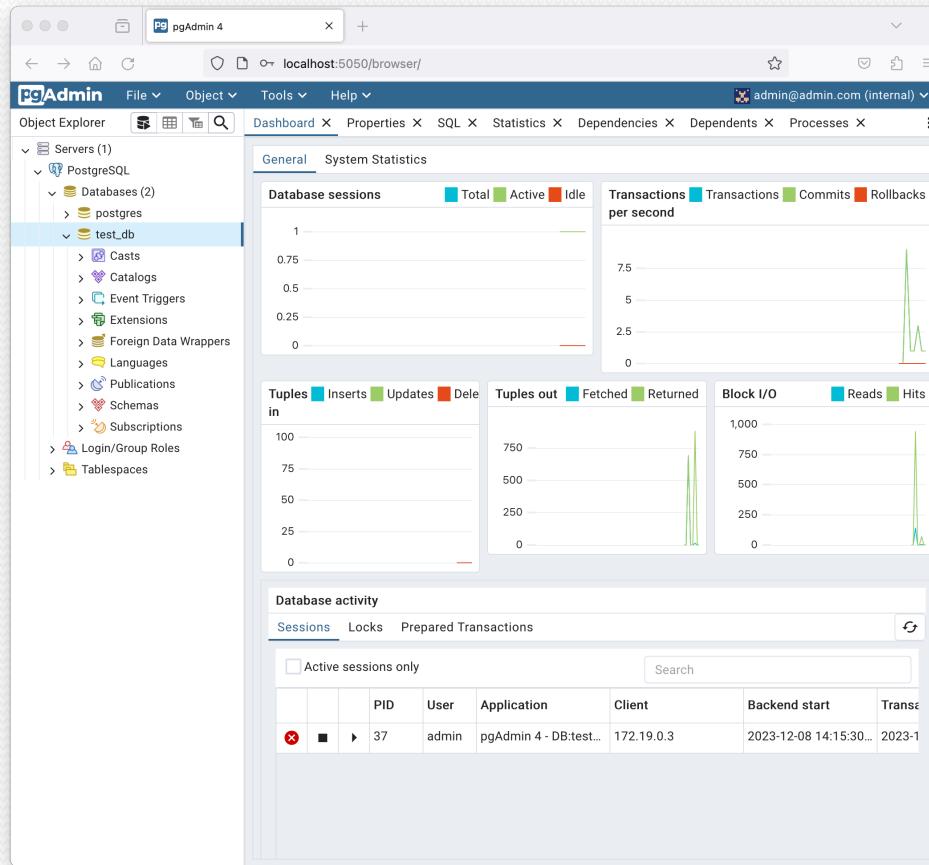
Register - Server

General Connection Parameters SSH Tunnel Advanced

Host name/address	postgres
Port	5432
Maintenance database	postgres
Username	admin
Kerberos authentication?	<input type="checkbox"/>
Password
Save password?	<input checked="" type="checkbox"/>
Role	
Service	

Buttons:

Access your database system – pgAdmin -> postgres



Access your database system – pgAdmin -> postgres

```
ken@sdeh136-08c4af PostgreSQL % pwd
/Users/ken/Documents/Database/PostgreSQL
ken@sdeh136-08c4af PostgreSQL % ls
docker-compose.yml
ken@sdeh136-08c4af PostgreSQL % docker compose down
[+] Running 3/3
✓ Container pgadmin           Removed          1.0s
✓ Container postgresql        Removed          0.1s
✓ Network postgresql_default Removed          0.0s
ken@sdeh136-08c4af PostgreSQL % docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED     STATUS      PORTS      NAMES
ken@sdeh136-08c4af PostgreSQL %
```

Acknowledgements

- Quickly set up a local postgres database using docker – Medium
 - <https://medium.com/@jewelski/quickly-set-up-a-local-postgres-database-using-docker-5098052a4726>
- docker – Docker Desktop
 - <https://www.docker.com/products/docker-desktop/>
- docker docs - Docker Compose
 - <https://docs.docker.com/compose/>
- dockerhub
 - <https://hub.docker.com/>
- PostgreSQL
 - <https://www.postgresql.org/>
- pgAdmin
 - <https://www.pgadmin.org/>