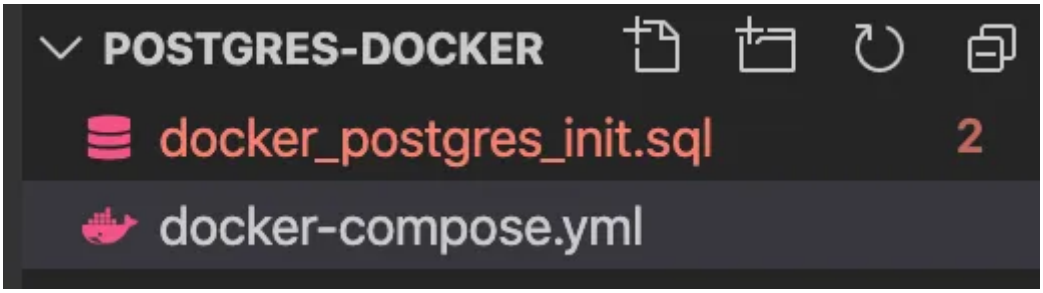
Hope you are familiar with “**Docker-Compose**”

Folder structure

```
Project
├── docker-compose.yml (File)
├── docker_postgres_init.sql (File)
```



Create a new file **docker-compose.yml**

```yaml
version: '3.6'
services:
    postgres:
        image: postgres
        restart: always
        environment:
            - DATABASE_HOST=127.0.0.1
            - POSTGRES_USER=root
            - POSTGRES_PASSWORD=root
            - POSTGRES_DB=root

        ports:
            - "5432:5432"
        volumes:
            - ./docker_postgres_init.sql:/docker-entrypoint-initdb.d/docker_postgres_init.sql

    pgadmin-compose:
        image: dpage/pgadmin4
        environment:
            PGADMIN_DEFAULT_EMAIL: "test@gmail.com"
            PGADMIN_DEFAULT_PASSWORD: "test123!"
        ports:
            - "16543:80"
        depends_on:
            - postgres
```

docker-compose.yml hosted with ❤ by **GitHub** **view raw**

Above file, we have created a **Postgres** Docker container with **Port No:** 5432

As well as set environment variables such as default **username, password** of **Postgres** container.

if you check the entire file we have also created volumes that point to the `docker_postgres_init.sql` file.

```
- ./docker_postgres_init.sql:/docker-entrypoint
initdb.d/docker_postgres_init.sql
```

The script inside **“docker_postgres_init.sql”** will create a new table **student** and insert the **records** as well

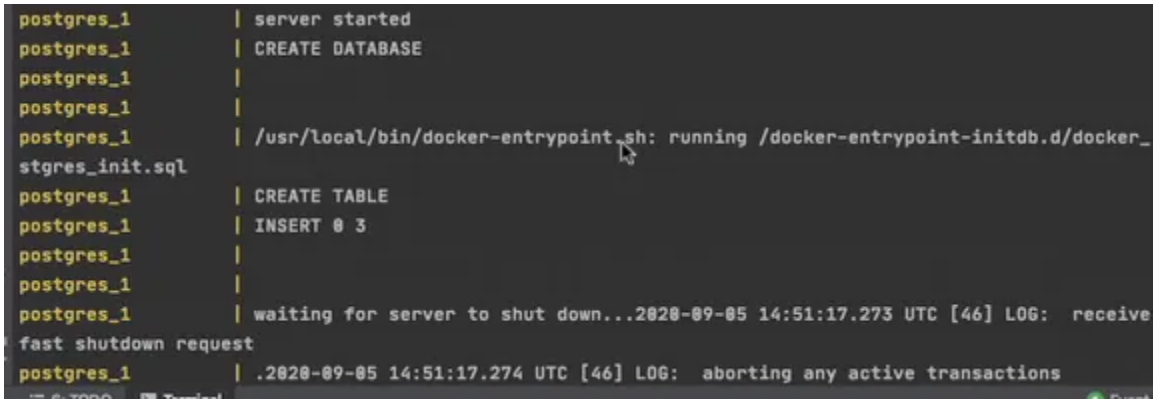
```sql
CREATE TABLE student
(
    id bigint NOT NULL,
    name text COLLATE pg_catalog."default",
    CONSTRAINT student_pkey PRIMARY KEY (id)
);

INSERT INTO student(id, name) VALUES
 (1, 'A'),
 (2, 'B'),
 (3, 'C');
```

docker_postgres_init.sql hosted with ❤ by GitHub view raw

When you run the following command in the root directory for the project.

```
docker-compose up
```

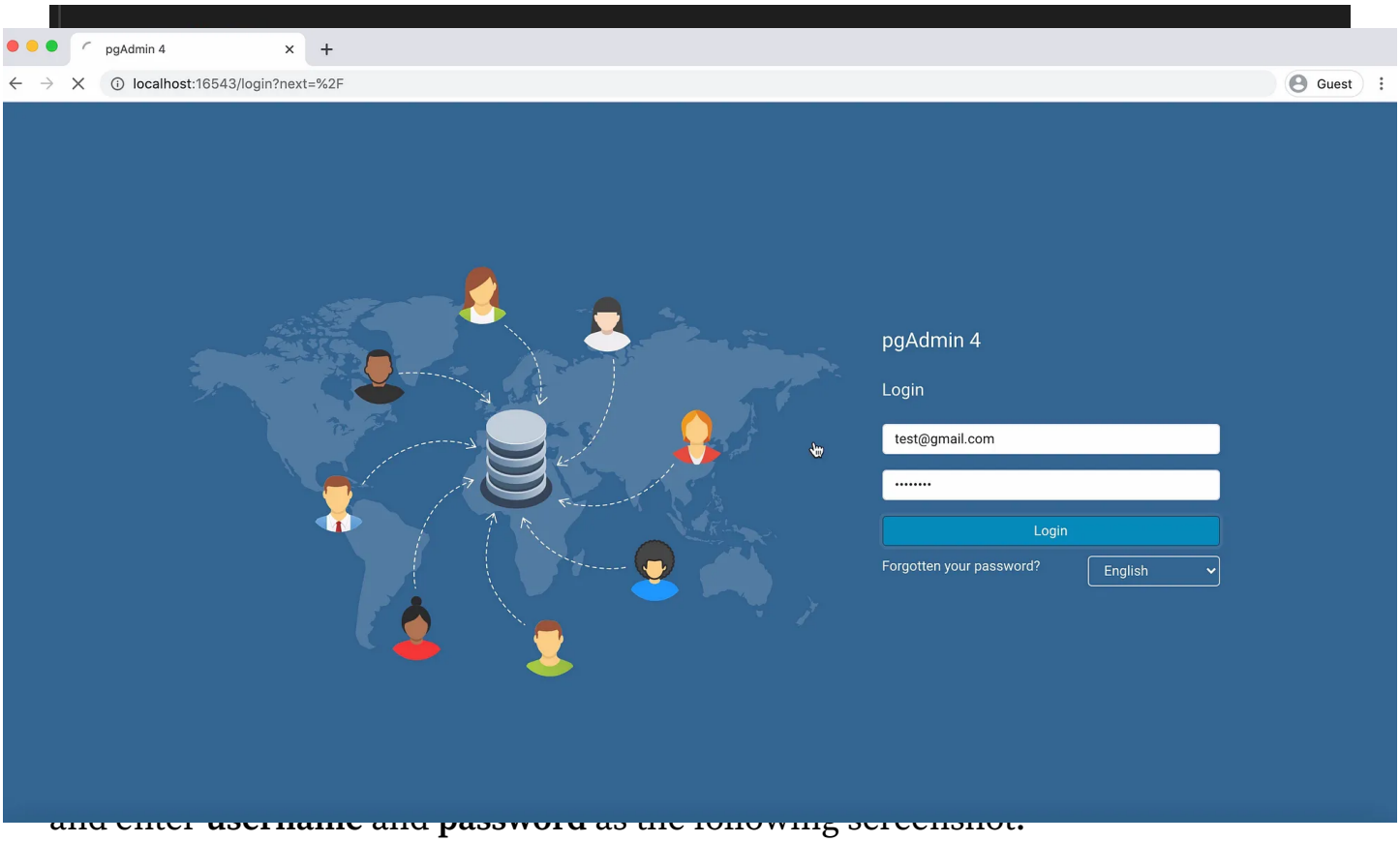
then you will see in the console **docker_postgres_init.sql** script will be executed as the following screenshot



that means your script executed successfully.

How to test

For testing, we used **Pgadmin4** configured already in the **docker-compose** file.

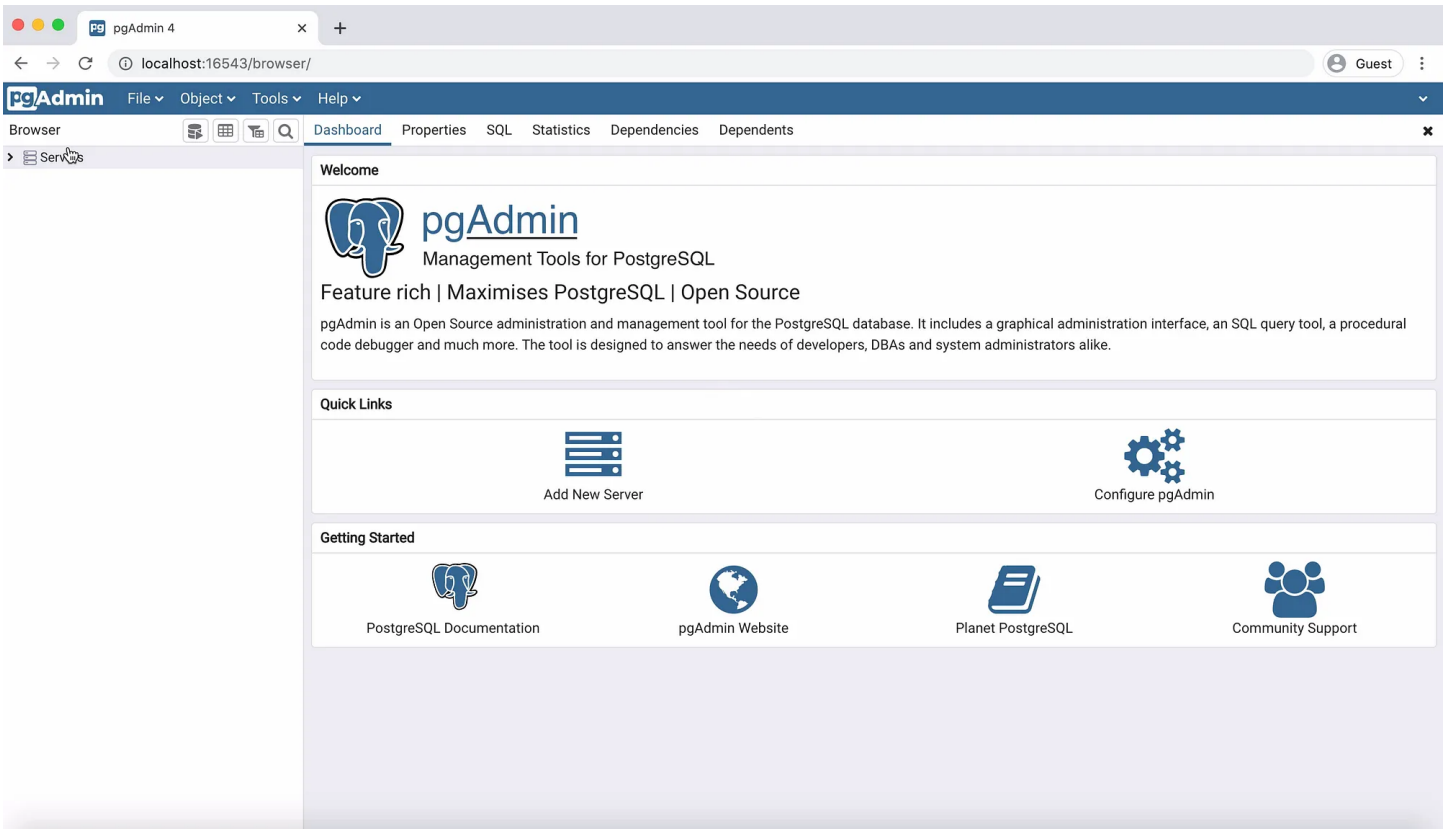


and enter **username** and **password** as the following screenshot.

Pgadmin4

```
username: test@gmail.com
password: test123!
```

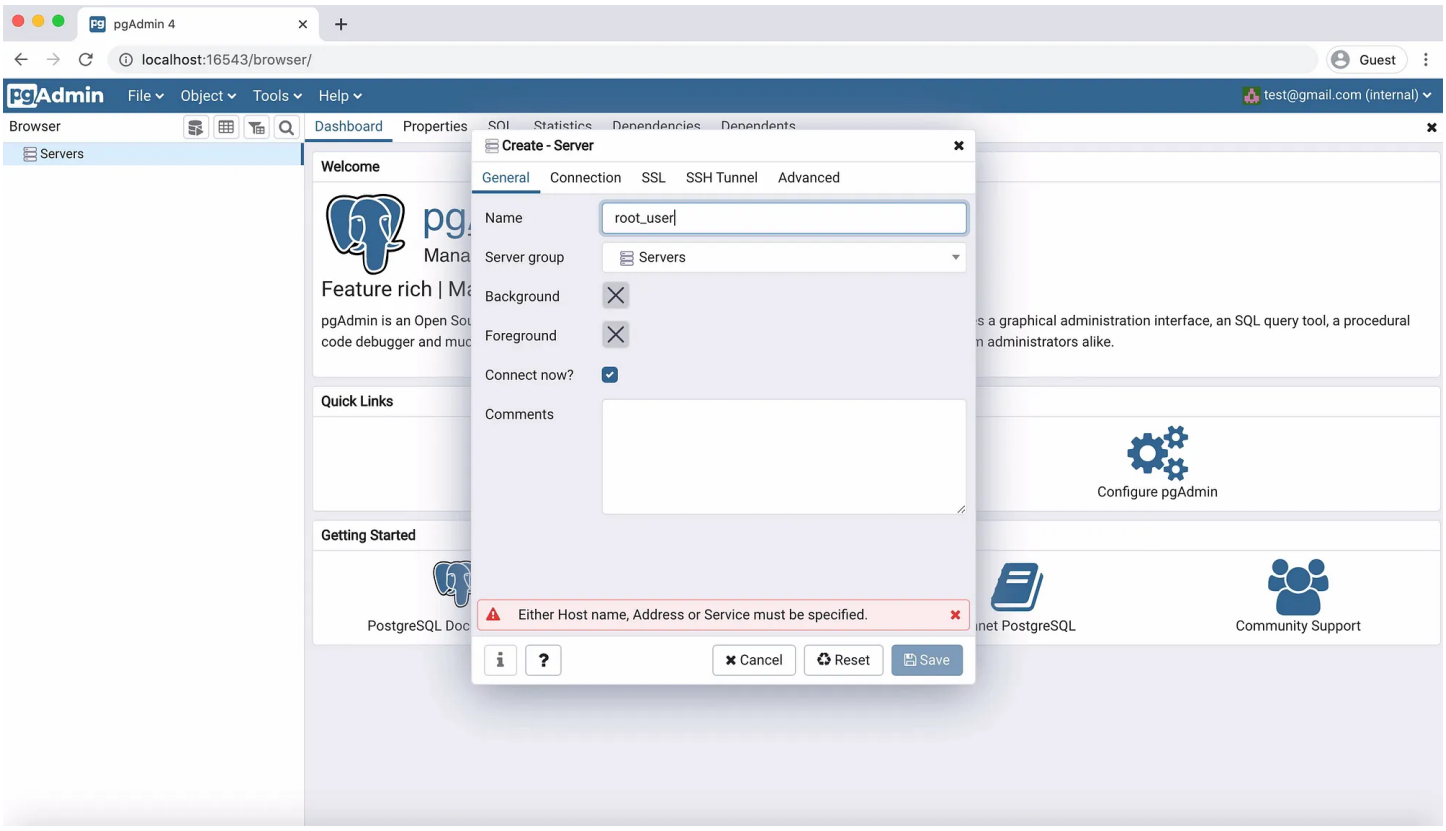
after successful login, you will see the following screen.



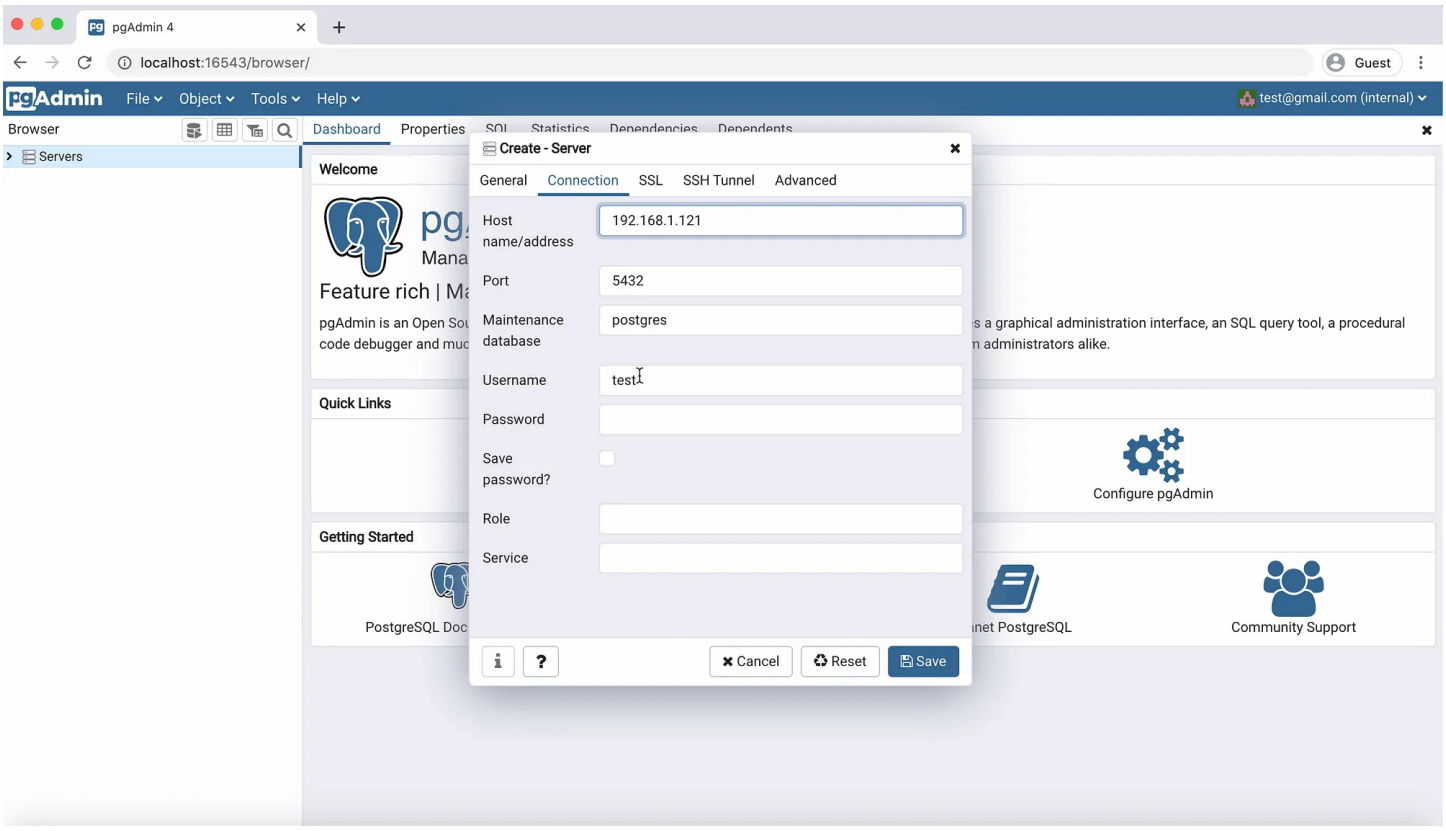
Then click on to **server -> create -> server**

You will see the following **pop up** then in the **General tab** type

name: root_user



Then click on the **Connection tab** as the following screenshot then input **Hostname**



You can get **Hostname** by typing the following command

```
$ ifconfig |grep inet
```

docker-compose.yml — postgres-docker

EXPLORER

OPEN EDITORS
- docker-compose.yml
- docker_postgres_init.sql 2

POSTGRES-DOCKER
- docker_postgres_init.sql 2
- docker-compose.yml

docker-compose.yml > {} services > {} pgadmin-compose > {} environment > PGADMIN_DEFAULT_PASSWORD

```yaml
        ports:
            - "5432:5432"
        volumes:
            - ./docker_postgres_init.sql:/docker-entrypoint-initdb.d/docker_postgres_init.sql

    pgadmin-compose:
        image: dpage/pgadmin4
        environment:
            PGADMIN_DEFAULT_EMAIL: "test@gmail.com"
            PGADMIN_DEFAULT_PASSWORD: "test123!"
        ports:
            - "16543:80"
        depends_on:
            - postgres
```

PROBLEMS 2 OUTPUT TERMINAL DEBUG CONSOLE 2: bash

```
postgres-docker $ifconfig |grep inet
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        inet6 fe80::aede:48ff:fe00:1122%en5 prefixlen 64 scopeid 0x4
        inet6 fe80::c2b:cee0:f50d:d785%en0 prefixlen 64 secured scopeid 0x6
        inet 192.168.1.121 netmask 0xffffff00 broadcast 192.168.1.255
        inet6 fe80::7c66:48ff:fe6f:77a6%awdl0 prefixlen 64 scopeid 0xe
        inet6 fe80::7c66:48ff:fe6f:77a6%llw0 prefixlen 64 scopeid 0xf
        inet6 fe80::8e62:323d:407:fba0%utun0 prefixlen 64 scopeid 0x10
        inet6 fe80::e4e8:69b8:666a:a36a%utun1 prefixlen 64 scopeid 0x11
        inet6 fe80::aff2:347:49bf:754b%utun2 prefixlen 64 scopeid 0x12
        inet6 fe80::31c5:5b12:f494:911b%utun3 prefixlen 64 scopeid 0x13
        inet6 fe80::7fc1:d3cf:83c7:86ce%utun5 prefixlen 64 scopeid 0x15
        inet6 fe80::1c4b:31de:4553:911f%en8 prefixlen 64 secured scopeid 0x17
        inet 169.254.38.217 netmask 0xffff0000 broadcast 169.254.255.255
postgres-docker $
```

OUTLINE

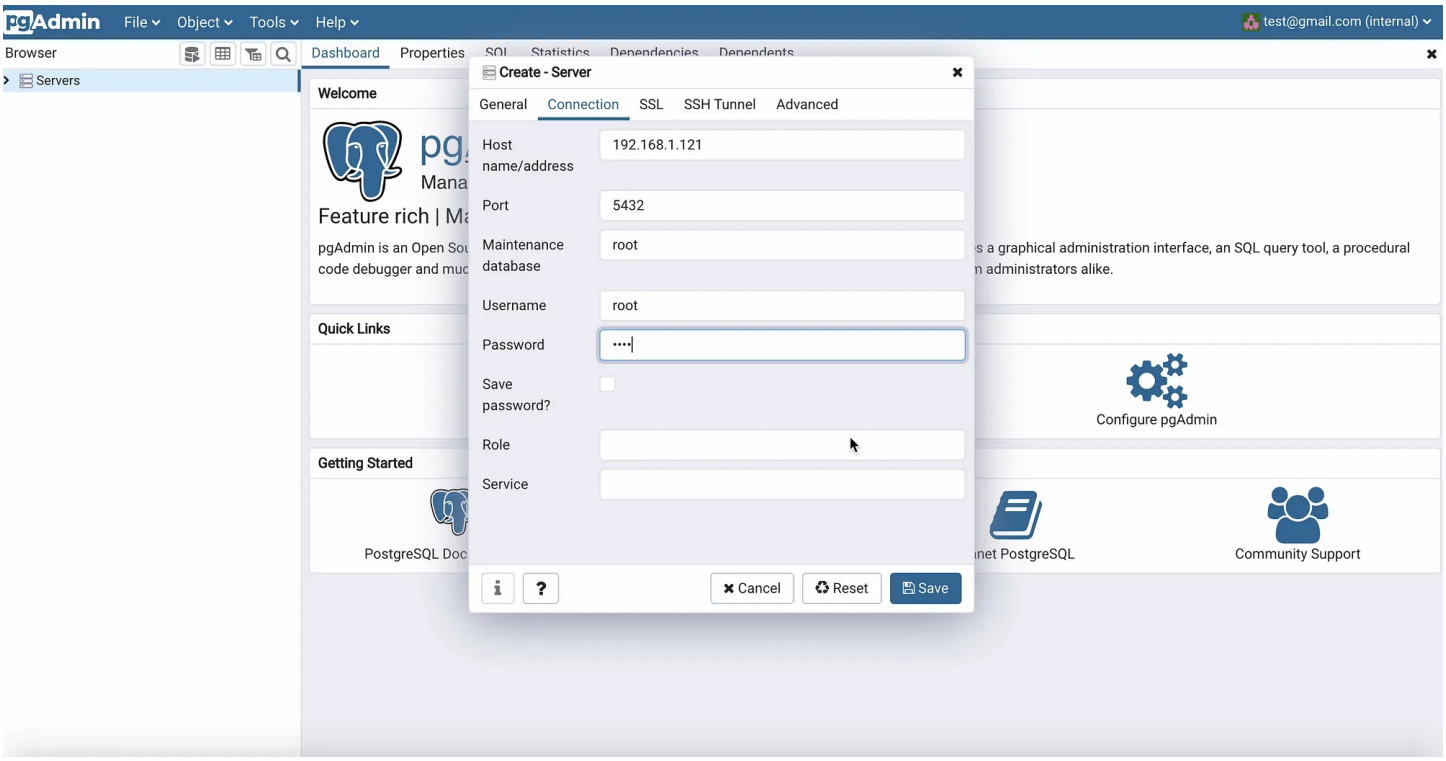
TIMELINE

then type

Database: root

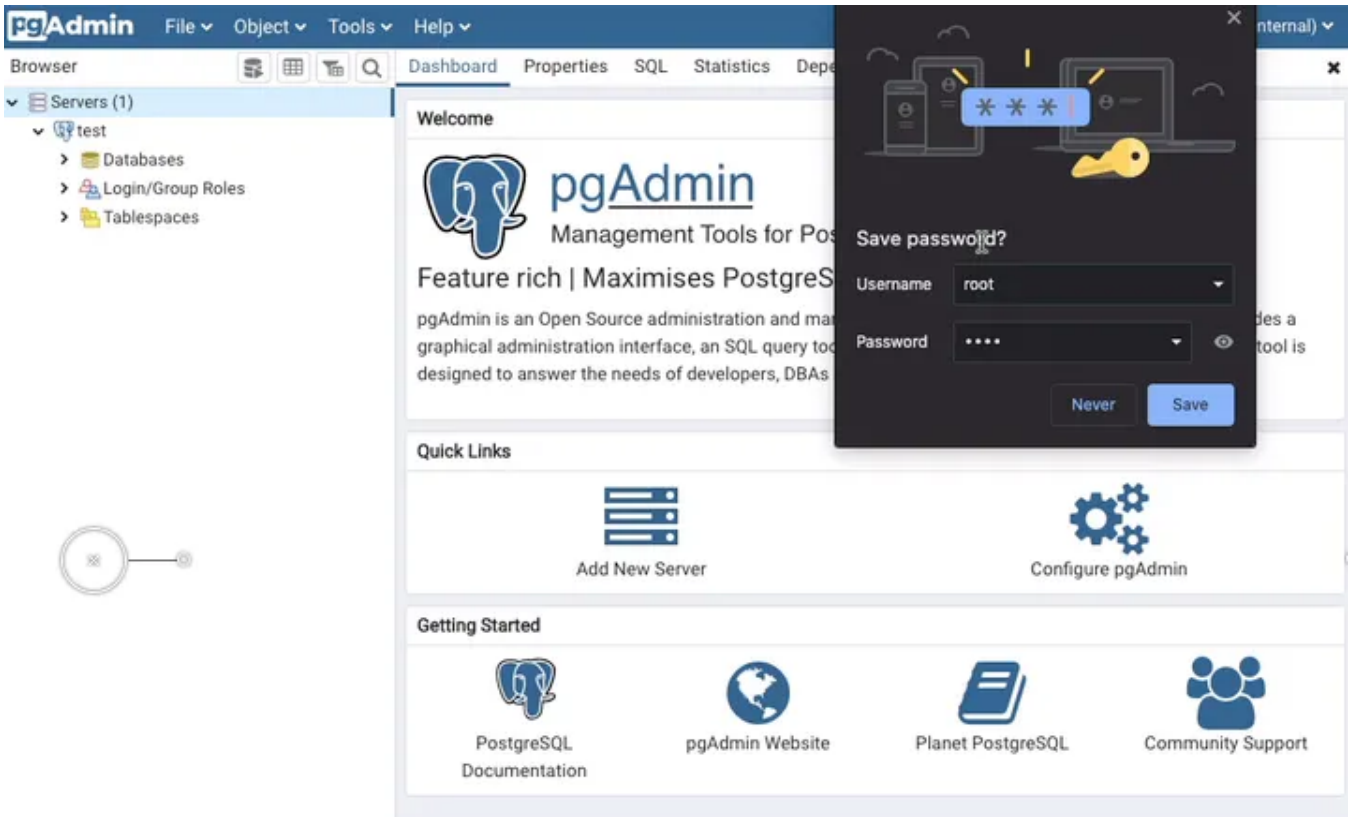
Username: root

Password: root

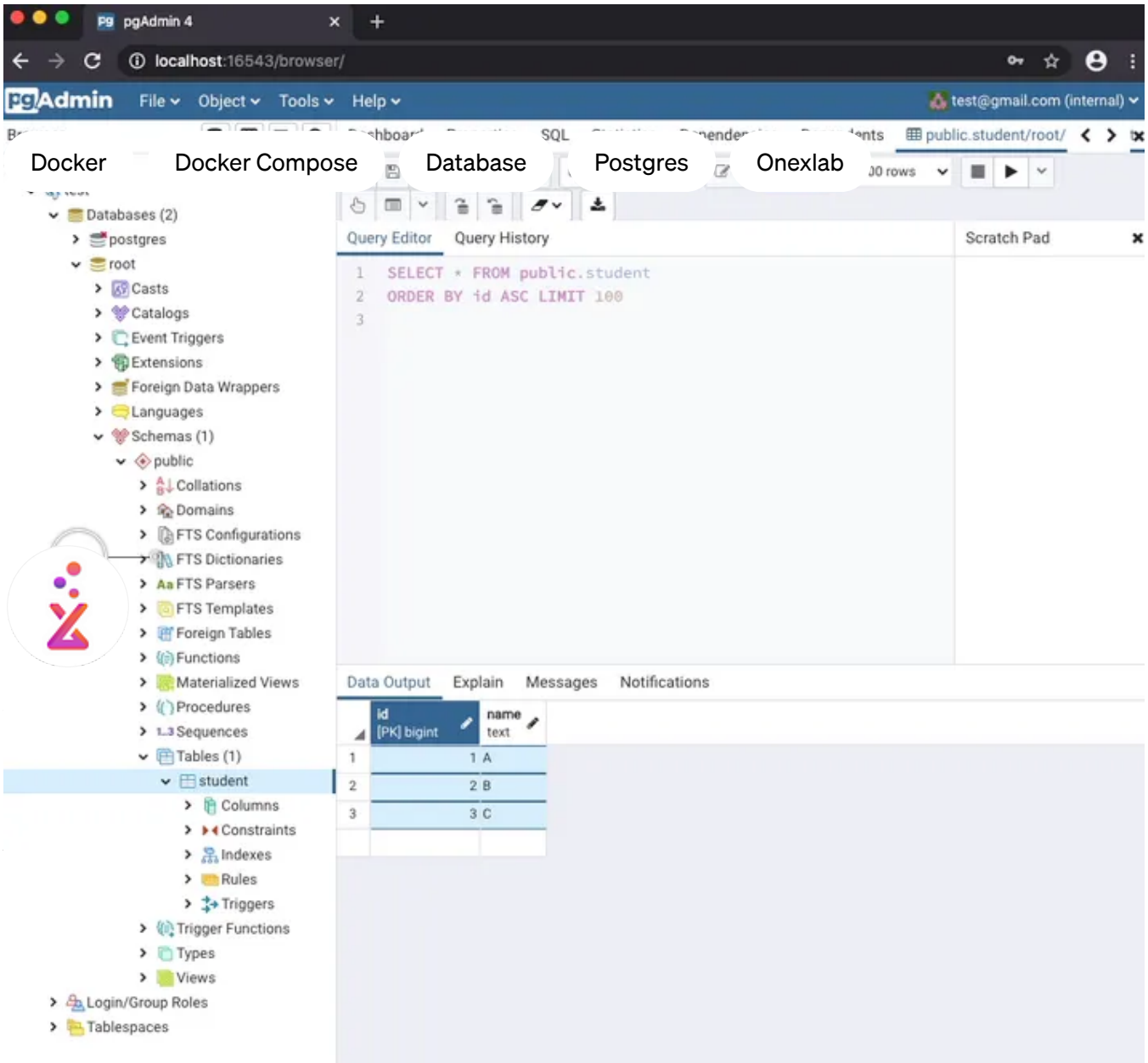


then click on to **Save Button**

then you will see the connection successful on the left side of the Pgadmin 4 navigation as shown below.



then go to **Databases->root->Schema->Tables->students** you will see there are 3 records as shown below in the screenshot.



Github

oxlb/docker-postgres-seeding

Docker Compose Postgres Seeding. Contribute to oxlb/docker-postgres-seeding development by creating an account on...

github.com

Thank you!