

Homework: Network, Security, and Orchestration of Docker

[Description](#)

[Getting Started](#)

[System Design](#)

[Dependencies](#)

[Installing](#)

[Executing](#)

[Demo](#)

[Execute greeting](#)

[Akses pgAdmin melalui Nginx](#)

[Call service `springboot-greeting` melalui Nginx](#)

[Restart `docker-compose`](#)

[Memastikan perubahan data dari `pgAdmin` terlihat pada `springboot-greeting`](#)

[Screenshot](#)

Description

Sebuah startup ingin membangun aplikasi berbasis web dengan arsitektur berikut:

1. Frontend: Menggunakan Nginx untuk melayani halaman statis.
2. Backend: API sederhana berbasis Python Flask.
3. Database: PostgreSQL untuk menyimpan data aplikasi.
4. Networking: Backend dan database harus saling terhubung dengan jaringan privat.
5. Volume: Database PostgreSQL memerlukan penyimpanan data yang persisten.
6. Scaling: Backend harus bisa di-scale hingga 3 replika.

Tugas Peserta

1. Buat Docker Compose file untuk menjalankan semua service.
2. Pastikan Nginx dapat diakses melalui browser.
3. Gunakan Docker Volume untuk menyimpan data PostgreSQL.
4. ~~Gunakan Docker Swarm untuk scaling backend menjadi 3 replika.~~
5. Pastikan semua container saling terhubung melalui jaringan

Getting Started

System Design

Rancangan sistem terdiri dari 4 komponen (service) sbb:

▼ **pg-greeting**

Database PostgreSQL

ref: https://hub.docker.com/_/postgres

▼ **pgadmin-greeting**

Web-based app untuk me-manage atau administrasi database postgres

ref: <https://hub.docker.com/r/dpage/pgadmin4>

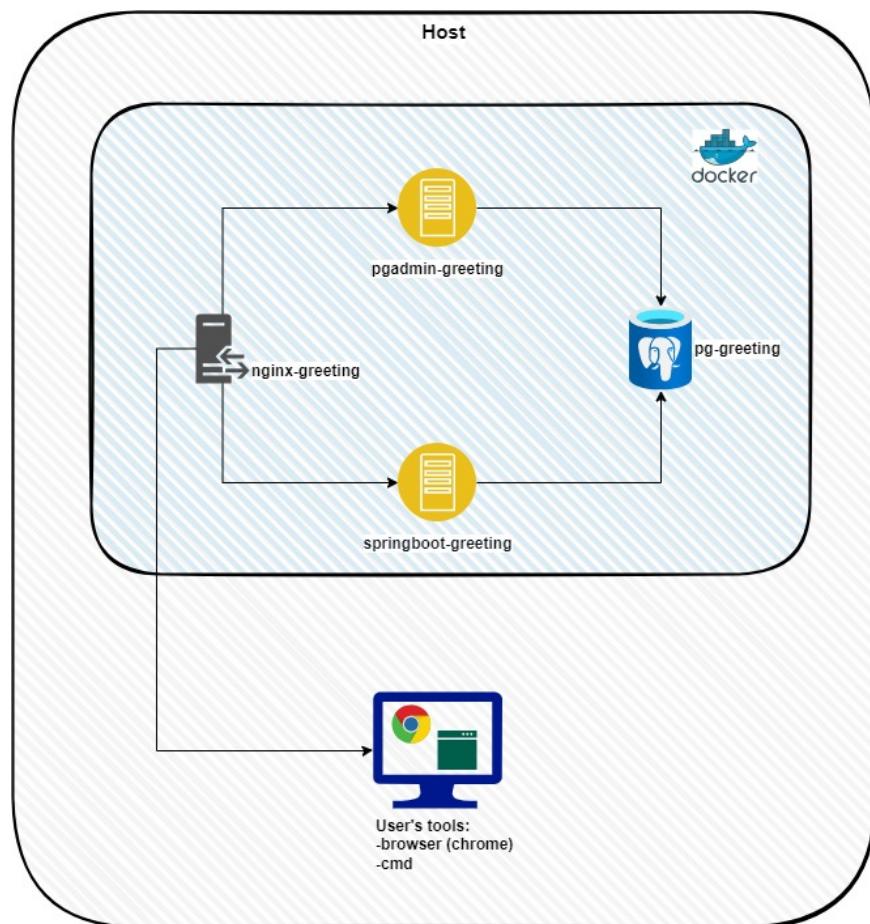
▼ **springboot-greeting**

Aplikasi rest-api berbasis Java-Springboot yang akan menampilkan pesan "hello <nama>" dari database

▼ **nginx-greeting**

Reverse proxy server, sebagai entry-point untuk mengakses sistem. **nginx-greeting** ini satu-satunya service yang akan di-ekspose port-nya ke public/host

Diagram



Dependencies

- Docker
- Browser (chrome, etc)
- cmd, powershell, etc
- Java 17, maven

Installing

- Download source code:

<https://github.com/galih15/devops-digital-skola-greeting>

```
greeting
|   .env
|   docker-compose.yml
+-- backend
|   |   Dockerfile
|   +-- greeting
|       +-- target
|           greeting-0.0.1.jar
+-- db
+-- pgadmin
+-- proxy
    default.conf
```

- Add `127.0.0.1 pgadmin.localhost` to file `hosts`

Executing

1. build `springboot-greeting`

```
cd backend\greeting
mvn clean package
```

```
user@Galih14 E:\Documents\Projects\DevOps\greeting\backend\greeting
$ mvn clean package
[INFO] Scanning for projects...
[INFO] [INFO] ----- < com.gaw:greeting > -----
[INFO] [INFO] Building greeting 0.0.1
[INFO] [INFO] from pom.xml
[INFO] [INFO] -----[ jar ]-----
[INFO] [INFO] --- clean:3.4.0:clean (default-clean) @ greeting ---
[INFO] [INFO] Deleting E:\Documents\Projects\DevOps\greeting\backend\greeting\target
[INFO] [INFO] --- resources:3.3.1:resources (default-resources) @ greeting ---
[INFO] [INFO] Copying 1 resource from src\main\resources to target\classes
[INFO] [INFO] Copying 0 resource from src\main\resources to target\classes
[INFO] [INFO] --- compiler:3.13.1:compile (default-compile) @ greeting ---
[INFO] [INFO] Recompiling the module because of changed source code.
[INFO] [INFO] Compiling 4 source files with javac [debug parameters release 17] to target\classes
[INFO] [INFO] --- resources:3.3.1:testResources (default-testResources) @ greeting ---
[INFO] [INFO] skip non existing resourceDirectory E:\Documents\Projects\DevOps\greeting\backend\greeting\src\test\resources
[INFO] [INFO] --- compiler:3.13.1:testCompile (default-testCompile) @ greeting ---
[INFO] [INFO] Recompiling the module because of changed dependency.
[INFO] [INFO] Compiling 1 source file with javac [debug parameters release 17] to target\test-classes
[INFO] [INFO] --- surefire:3.5.2:test (default-test) @ greeting ---
[INFO] [INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO] [INFO] -----
[INFO] [INFO] T E S T S
[INFO] [INFO] -----
[INFO] [INFO] Results:
[INFO] [INFO] Tests run: 0, Failures: 0, Errors: 0, Skipped: 0
[INFO] [INFO] -----
[INFO] [INFO] --- jar:3.4.2:jar (default-jar) @ greeting ---
[INFO] [INFO] Building jar: E:\Documents\Projects\DevOps\greeting\backend\greeting\target\greeting-0.0.1.jar
[INFO] [INFO] --- spring-boot:3.4.1:repackage (repackage) @ greeting ---
[INFO] [INFO] Replacing main artifact E:\Documents\Projects\DevOps\greeting\backend\greeting\target\greeting-0.0.1.jar with repackaged archive, adding nested dependencies in BOOT-INF/.
[INFO] [INFO] The original artifact has been renamed to E:\Documents\Projects\DevOps\greeting\backend\greeting\target\greeting-0.0.1.jar.original
[INFO] [INFO] BUILD SUCCESS
[INFO] [INFO] -----
[INFO] [INFO] Total time: 4.364 s
[INFO] [INFO] Finished at: 2025-01-05T04:37:37+07:00
[INFO] [INFO] -----
```

2. Run `docker-compose`

```
docker-compose up -d
```

```

PS E:\Documents\Projects\DevOps\greeting> docker-compose up -d
[*] Running 2/2
  ✓ node:latest
  ✓ db:latest
  ✓ backend Pulled
[*] Running 0/1
[*] Building 1.6s (2/3)
[*] Building 2.4s (16/10) FINISHED
=> [backend internal] load build definition from Dockerfile
=> => transferring dockerfile: 211B
=> [backend internal] load metadata for docker.io/library/openjdk:17-jdk-slim
=> [backend auth] library/openjdk:pull token for registry-1.docker.io
=> [backend internal] validate https://registry-1.docker.io/v2/
=> => transferring context: 2B
=> [backend 1/3] FROM docker.io/library/openjdk:17-jdk-slim@sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f68c3f62f9774
=> [backend internal] load build context
=> => transferring context: 121B
=> CACHED [backend 2/3] WORKDIR /app
=> CACHED [backend 3/3] COPY greeting/target/greeting-0.0.1.jar /app/app.jar
=> [backend] exporting to image
=> exporting layers
=> => layer sha256:30e93915c838e4989b6e162d67ce2f7867a8211381cfb4e0de2702eee2509bd16
[*] Running 5/5 docker.io/library/greeting-backend
  ✓ Service backend      Built
  ✓ Container pg-greeting Started
  ✓ Container springboot-greeting Started
  ✓ Container pgadmin-greeting Started
  ✓ Container nginx-greeting Started

```

3. Verify that services is running

```
docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2fec21b2ce8366	nginx:latest	"./docker-entrypoint..."	11 seconds ago	Up 10 seconds	0.0.0.0:8080->80/tcp	nginx-greeting
9cb3c2cece93	dpage/pgadmin4:latest	"/entrypoint.sh"	11 seconds ago	Up 11 seconds	80/tcp, 443/tcp	pgadmin-greeting
12c44f796e66	greeting-backend	"java -jar /app/app..."	11 seconds ago	Up 11 seconds	8081/tcp	springboot-greeting
226cd62c1955	postgres:17.2	"docker-entrypoint.s..."	11 seconds ago	Up 11 seconds	5432/tcp	pg-greeting

Demo

Demo project ini bisa dilihat di

<https://youtu.be/lLqZYiP6j38>

Execute greeting



Akses pgAdmin melalui Nginx

- Akses <http://pgadmin.localhost:8080/> melalui browser
- Login sesuai email/password di [.env](#)
- Create connection to [pg-greeting](#)
- Create table [users](#)

```

CREATE TABLE public.users (
    user_id serial4 NOT NULL,
    username varchar(50) NOT NULL,
    created_at timestamp DEFAULT CURRENT_TIMESTAMP NULL,
    CONSTRAINT users_pkey PRIMARY KEY (user_id),
    CONSTRAINT users_username_key UNIQUE (username)
);

```



```
insert into users (username) values ('John');
```

Call service [springboot-greeting](#) melalui Nginx

- Call <http://localhost:8080/hello> melalui web browser

Restart docker-compose

Memastikan bahwa data terakhir masih tersimpan setelah docker di-restart

- ✓ `docker-compose down` lalu `docker-compose up -d`
- ✓ login ke `pgAdmin` untuk melihat data "John" masih ada
- ✓ Call `http://localhost:8080/hello` melalui web browser

Memastikan perubahan data dari pgAdmin terlihat pada springboot-greeting

- ✓ Call `http://localhost:8080/hello` multiple times (execute bat script dari cmd dengan delay 1 detik)

```
@ECHO OFF
ECHO "call http://localhost:8080/hello with delay 1 sec"

for /l %%x in (1,1,100) do (
    curl http://localhost:8080/hello
    timeout /t 1 /nobreak >nul
    ECHO .
)

ECHO "DONE"
```

- ✓ Insert new value melalui `pgAdmin` -

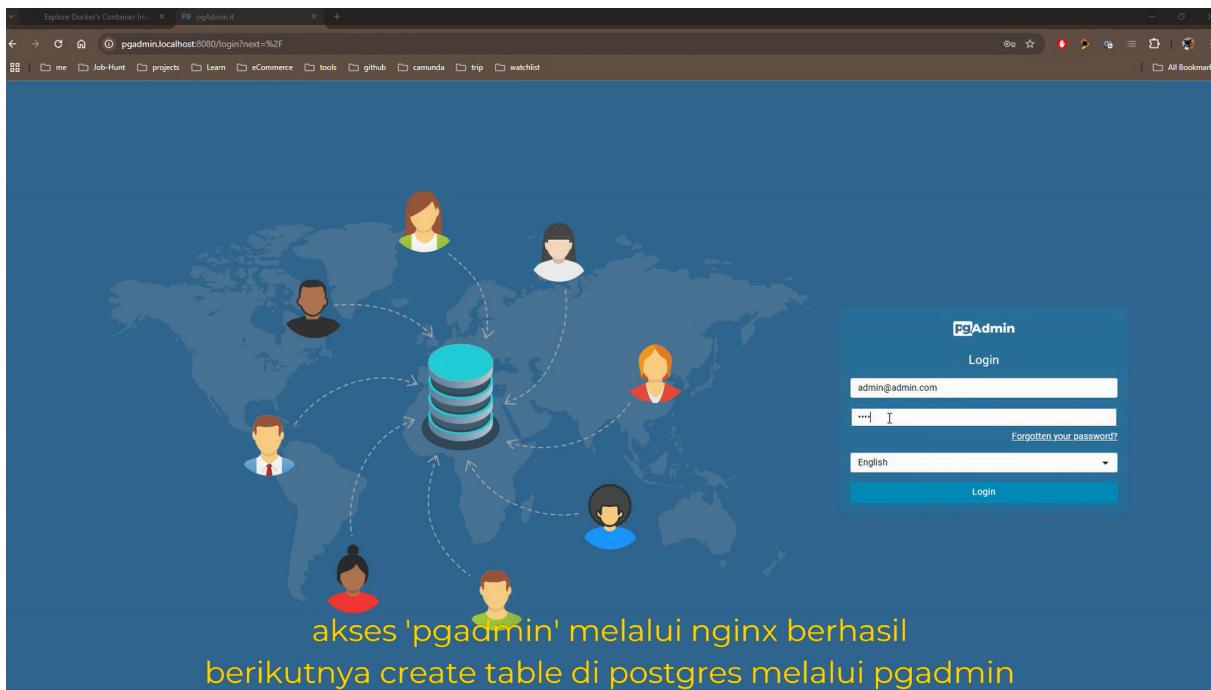
```
insert into users (username) values ('Doe');
```

- ✓ Perubahan response langsung terlihat pada hasil `curl`

Screenshot

```
PS E:\Documents\Projects\DevOps\greeting> docker-compose up -d
[...]
✓ Container pg-greeting      Started
✓ Container springboot-greeting  Started
✓ Container pgadmin-greeting   Started
✓ Container nginx-greeting    Started
$ E:\Documents\Projects\DevOps\greeting>
PS E:\Documents\Projects\DevOps\greeting> docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
e348077f7c3        dpage/pgadmin4:latest   "/entrypoint.sh"   9 minutes ago     Up 7 seconds          0.0.0.0:8080->80/tcp   pgadmin-greeting
8028c67c7f1f2       postgres:17.2          "docker-entrypoint..." 9 minutes ago     Up 8 seconds          80/tcp, 443/tcp      nginx-greeting
pg-greeting
```

container 'springboot-greeting' belum bisa up karena db-object belum ada



pgAdmin 4

pgAdmin is an Open Source administration tool for PostgreSQL. It provides a graphical interface to manage databases, users, and other PostgreSQL objects.

Welcome

Register - Server

General Connection Parameters SSH Tunnel Advanced

Host name/address: pg-greeting
Port: 5432
Maintenance database: postgres
Username: admin
Kerberos authentication?
Password:
Save password?
Role:
Service:

Configure pgAdmin

PostgreSQL Documentation

Community Support

connect ke database sesuai credential yang ada di file `.env`

pgAdmin 4

pgAdmin is an Open Source administration tool for PostgreSQL. It provides a graphical interface to manage databases, users, and other PostgreSQL objects.

Object Explorer

Servers (1) pg-greeting Databases (2) greeting

- greeting
 - Casts
 - Catalogs
 - Check Triggers
 - Extensions
 - Foreign Data Wrappers
 - Languages
 - Publications
 - Schemas (1)
 - Tables (1)
 - Views (1)
- public
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Types

Dashboard Properties SQL Statistics Dependencies Dependents Processes

Query Query History

```
1 < CREATE TABLE public.users (
2   id serial NOT NULL,
3   username varchar(50),
4   created_at timestamp DEFAULT CURRENT_TIMESTAMP NULL,
5   CONSTRAINT users_pkey PRIMARY KEY (user_id),
6   CONSTRAINT users_username_key UNIQUE (username)
7 );
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 243 msec.

The screenshot shows the pgAdmin 4 interface. In the Object Explorer, under the 'greeting' database, a new table named 'users' is being created. The table has three columns: 'user_id' (PK integer), 'username' (character varying(50)), and 'created_at' (timestamp without time zone). A query in the main pane runs the command: `select * from users;`. The Data Output pane shows the result of the query, which is empty at this point. A success message at the bottom right indicates: "Successfully run. Total query runtime: 91 msec. 0 rows affected."

This screenshot shows the pgAdmin 4 interface again. The 'users' table now contains one row with the value 'John'. The Data Output pane displays the result of the query: `select * from users;`. The result table shows a single row with user_id 1, username 'John', and created_at '2025-01-05 06:22:04.553105'. A success message at the bottom right indicates: "Total query runtime: 74 msec. 1 rows affected."

```

Windows PowerShell x gw@gwD14:~/mtncUser/v + ~
$ E:\Documents\Projects\DevOps\greeting> docker-compose up -d
✓ Container pg-greeting Started
✓ Container springboot-greeting Started
✓ Container pgadmin-greeting Started
✓ Container nginx-greeting Started
$ E:\Documents\Projects\DevOps\greeting>
$ E:\Documents\Projects\DevOps\greeting> docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
c3a6ff8860d nginx:latest "/docker-entrypoint..." 9 minutes ago Up 7 seconds 0.0.0.0:8080->80/tcp nginx-greeting
c3a6ff8860d dpage/pgadmin:latest "/entrypoint.sh" 9 minutes ago Up 8 seconds 80/tcp, 443/tcp pgadmin-greeting
028e67c7f2 postgres:17.2 "docker-entrypoint.s..." 9 minutes ago Up 8 seconds 5432/tcp pg-greeting
$ E:\Documents\Projects\DevOps\greeting>
$ E:\Documents\Projects\DevOps\greeting> docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
c3a6ff8860d nginx:latest "/docker-entrypoint..." 13 minutes ago Up 4 minutes 0.0.0.0:8080->80/tcp nginx-greeting
c3a6ff8860d dpage/pgadmin:latest "/entrypoint.sh" 13 minutes ago Up 4 minutes 80/tcp, 443/tcp pgadmin-greeting
028e67c7f2 postgres:17.2 "docker-entrypoint.s..." 13 minutes ago Up 4 minutes 5432/tcp pg-greeting
$ E:\Documents\Projects\DevOps\greeting>
$ E:\Documents\Projects\DevOps\greeting> docker-compose down
[*] Stopping 5/5
✓ Container nginx-greeting Removed
✓ Container springboot-greeting Removed
✓ Container pgadmin-greeting Removed
✓ Container pg-greeting Removed
✓ Network greeting_network Removed
$ E:\Documents\Projects\DevOps\greeting>
$ E:\Documents\Projects\DevOps\greeting> docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
$ E:\Documents\Projects\DevOps\greeting>
$ E:\Documents\Projects\DevOps\greeting> docker-compose up -d
[*] Starting 5/5
✓ Container nginx-greeting Created
✓ Container springboot-greeting Started
✓ Container pgadmin-greeting Started
✓ Container pg-greeting Started
✓ Container nginx-greeting Started
$ E:\Documents\Projects\DevOps\greeting>

```

menguji persistent volume postgres:
 - docker-compose down
 - docker compose up -d

```

PS E:\Documents\Projects\DevOps\greeting>
$ E:\Documents\Projects\DevOps\greeting> docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ef86049a3d71 nginx:latest "/docker-entrypoint..." 7 seconds ago Up 6 seconds 0.0.0.0:8080->80/tcp nginx-greeting
96ba1657b65d dpage/pgadmin4:latest "/entrypoint.sh" 7 seconds ago Up 6 seconds 80/tcp, 443/tcp pgadmin-greeting
58c31a1fd8d greeting-backend "java -jar /app/app..." 7 seconds ago Up 6 seconds 8081/tcp springboot-greeting
0e7aef3f92dc postgres:17.2 "docker-entrypoint.s..." 7 seconds ago Up 6 seconds 5432/tcp pg-greeting
$ E:\Documents\Projects\DevOps\greeting>

```

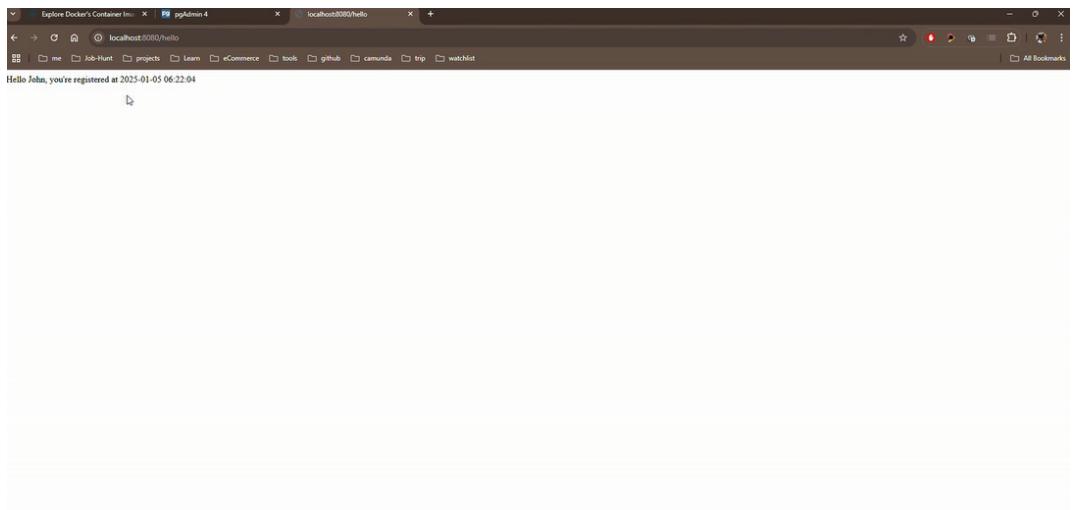
kali ini container 'springboot-greeting' sudah up
 karena table 'users' sudah dibuat

```

Explore Docker's Container Instances pgAdmin 4 x pgAdmin 4
pgAdmin File Object Tools Help *
Dashboard x Properties x SQL x Statistics x Dependencies x Dependents x Processes x greeting/admin@pg-greeting* x
admin@admin.com (internal) 
Object Explorer Servers (1)
pgAdmin pg-greeting
  Databases (2)
    greeting
      > Casts
      > Catalogs
      > Extension
      > Foreign Data Wrappers
      > Languages
      > Publications
      > Schemas (1)
        > public
          > Aggregates
          > Functions
          > Domains
          > FTS Configurations
          > FTS Dictionaries
          > FTS Parsers
          > FTS Templates
          > Foreign Tables
          > Functions
          > Materialized Views
          > Operators
          > Procedures
          > Sequences
        > Tables (1)
          > users
            > Columns
            > Constraints
            > Indexes
            > RLS Policies
            > Rules
            > Triggers
            > Trigger Functions
            > Triggers
            > Views
            > Subscriptions
          > Postgres
        > Login/Group Roles
  Data Output Messages Notifications
  user_id [PK] integer username character varying(50) created_at timestamp without time zone
  1 1 John 2025-01-05 06:22:04.553105
  Total rows: 1 of 1 Query complete 00:00:00.118
  Successfully run. Total query runtime: 118 msec. 1 rows affected. X
  Ln 1, Col 21

```

data dengan username 'John' masih ada



call service 'springboot-greeting' melalui nginx berhasil

Setelah insert data 'Doe' melalui `pgAdmin`
`springboot-greeting` langsung menampilkan nama 'Doe'