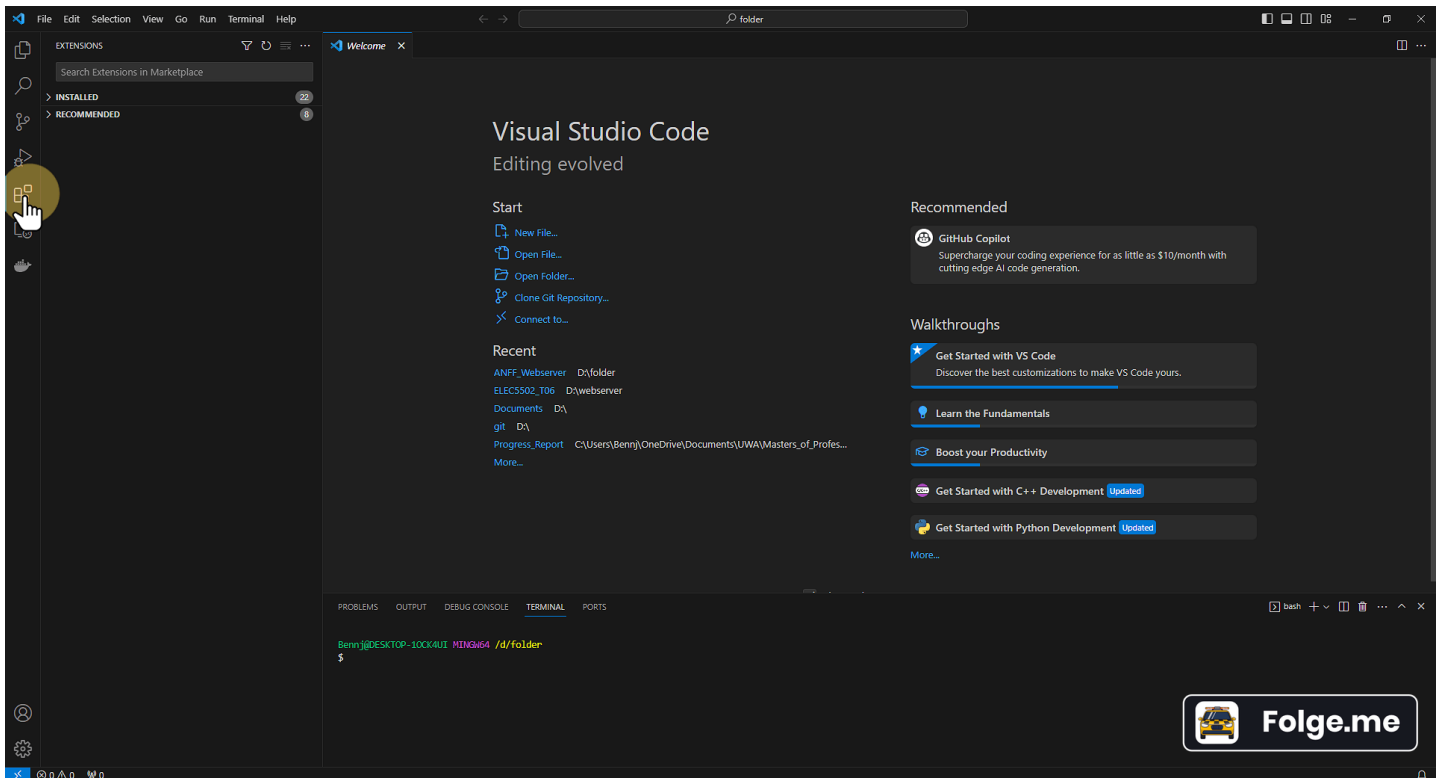
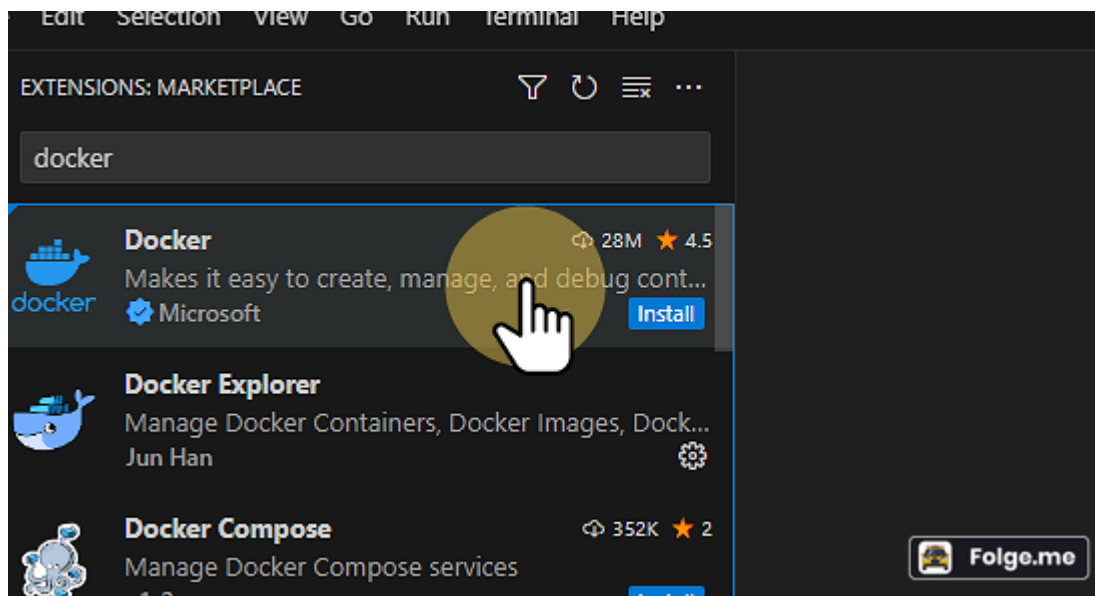


# Web Server Quick Setup Guide

## 1 Open VS Code and navigate to extensions



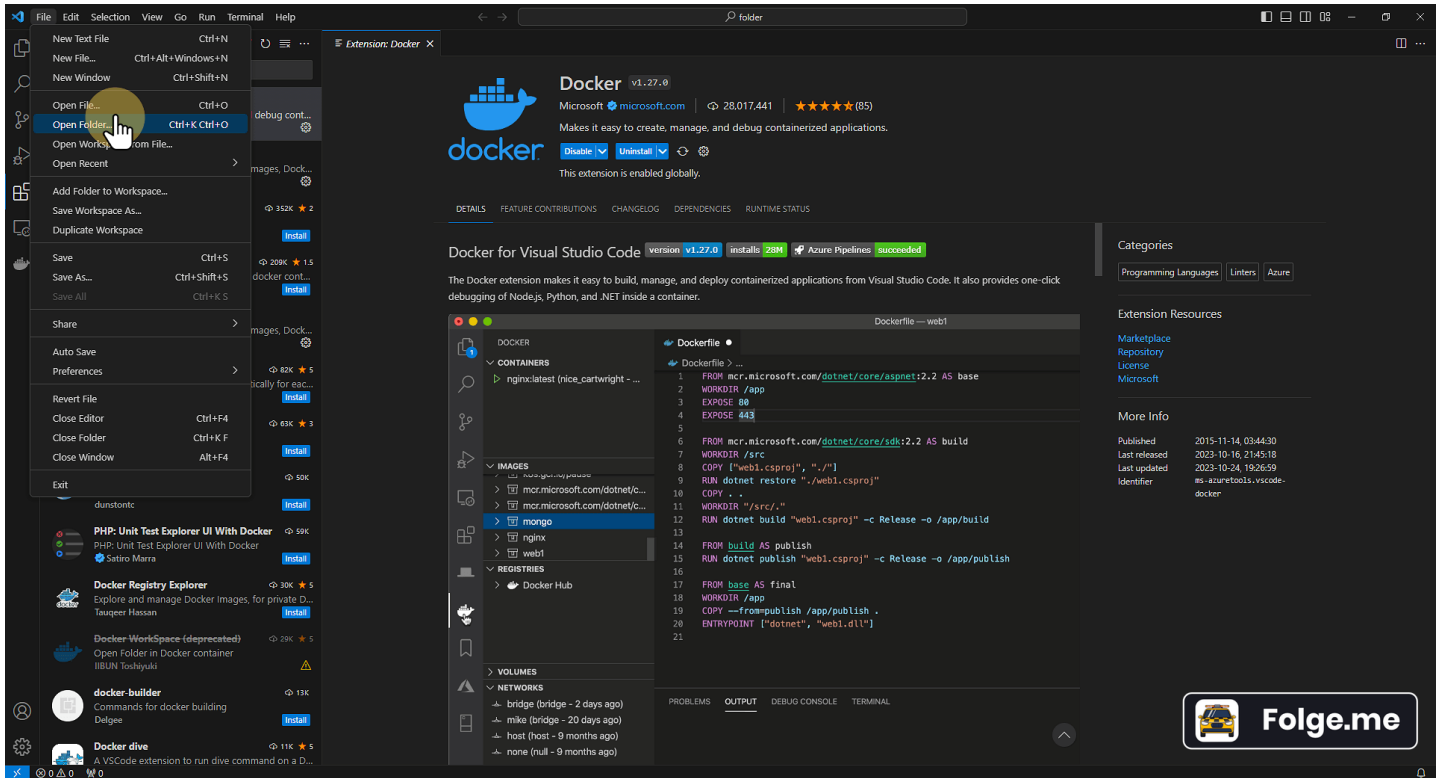
## 2 Search for Docker



## 2.1 Install Docker extension



### 3 Open folder in VS code that you want to clone the web server into



## 3.1

The screenshot displays the Visual Studio Code interface during the Docker setup process. On the left, the 'Open Folder' dialog is open, showing a file explorer view of 'Secondary storage (D:)'. A folder named 'webserver' is highlighted. Below this, the 'Extensions' view shows a list of Docker-related extensions, including 'Docker Runner', 'vscode-docker-syntax', 'PHP: Unit Test Explorer UI With Docker', 'Docker Registry Explorer', 'Docker-WorkSpace (deprecated)', 'docker-builder', and 'Docker dive'. The 'Docker Runner' extension is selected. In the center, the 'Dockerfile' editor shows a Dockerfile for a .NET application. The Dockerfile content is as follows:

```
Dockerfile -- web1
1 FROM mcr.microsoft.com/dotnet/core/aspnet:2.2 AS base
2 WORKDIR /app
3 EXPOSE 80
4 EXPOSE 443
5
6 FROM mcr.microsoft.com/dotnet/core/sdk:2.2 AS build
7 WORKDIR /src
8 COPY ["web1.csproj", "."]
9 RUN dotnet restore "/web1.csproj"
10 COPY . .
11 WORKDIR "/src/"
12 RUN dotnet build "web1.csproj" -c Release -o /app/build
13
14 FROM build AS publish
15 RUN dotnet publish "web1.csproj" -c Release -o /app/publish
16
17 FROM base AS final
18 WORKDIR /app
19 COPY --from=publish /app/publish .
20 ENTRYPOINT ["dotnet", "web1.dll"]
21
```

On the right, the 'Docker' view shows the 'Dockerfile' tab selected, displaying the Dockerfile content. Below the Dockerfile, the 'PROBLEMS' and 'OUTPUT' tabs are visible. The 'Docker' view also shows the 'Dockerfile' tab selected, displaying the Dockerfile content. The 'Docker' view also shows the 'Dockerfile' tab selected, displaying the Dockerfile content.

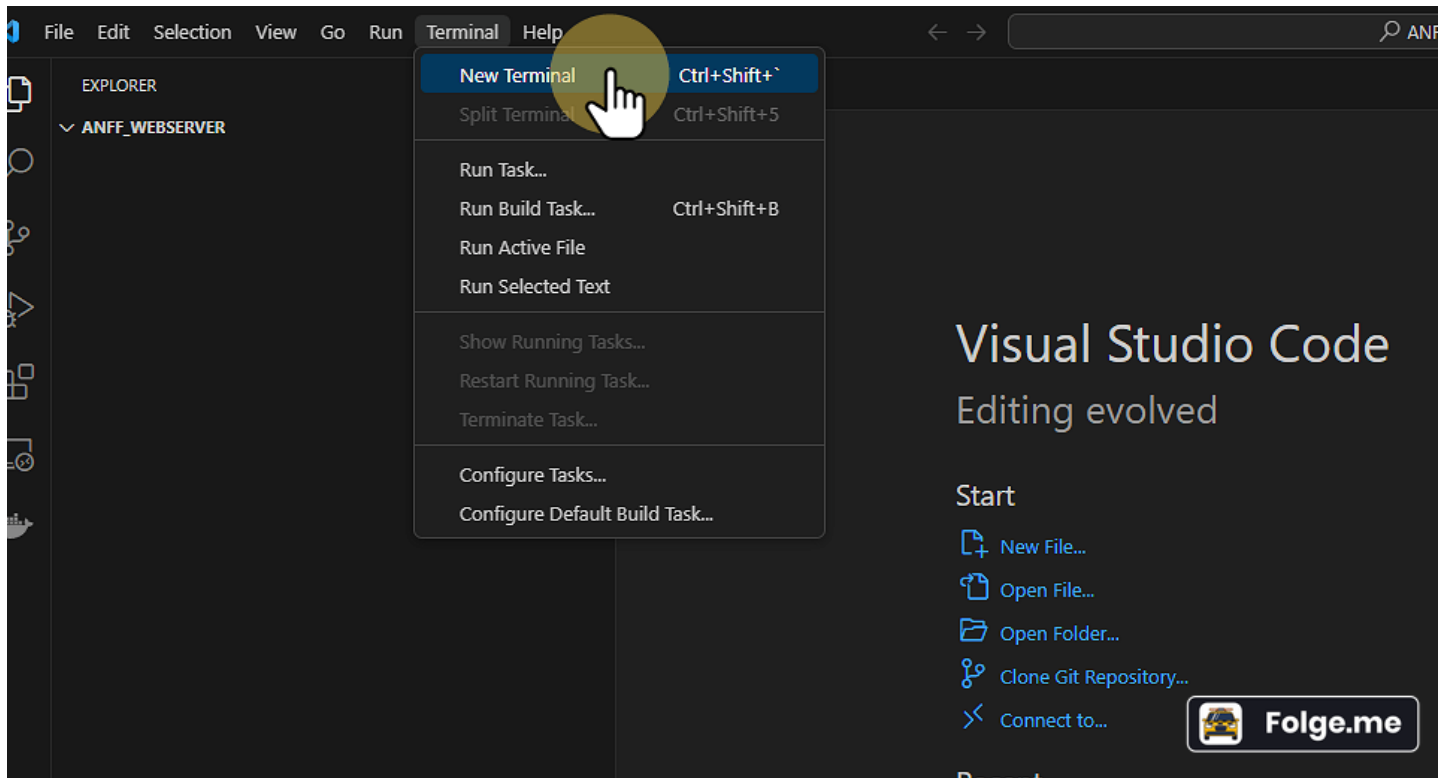
## 3.2

The screenshot shows the Visual Studio Code interface with a Dockerfile being edited. A yellow circle highlights the 'Select Folder' button in the 'Folder' dialog box. The Dockerfile content is as follows:

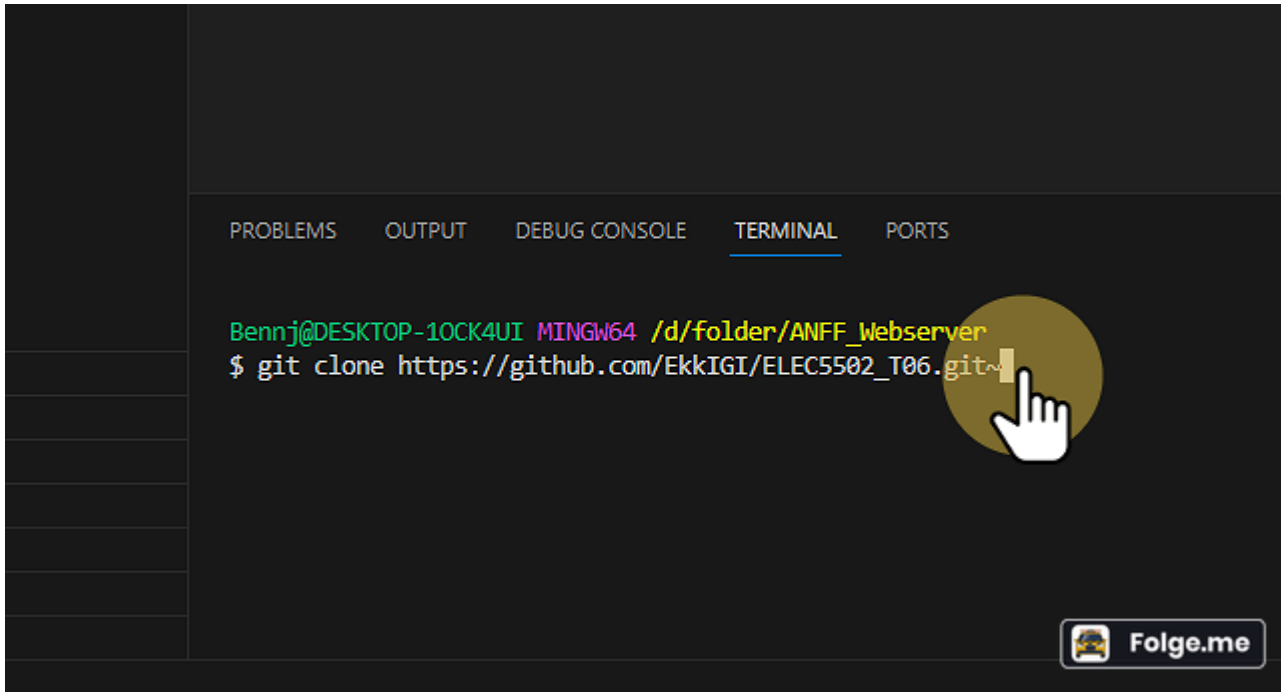
```
Dockerfile > ...
1 FROM mcr.microsoft.com/dotnet/core/aspnet:2.2 AS base
2 WORKDIR /app
3 EXPOSE 80
4 EXPOSE 443
5
6 FROM mcr.microsoft.com/dotnet/core/sdk:2.2 AS build
7 WORKDIR /src
8 COPY ["web1.csproj", "."]
9 RUN dotnet restore "./web1.csproj"
10 COPY . .
11 WORKDIR "/src/"
12 RUN dotnet build "web1.csproj" -c Release -o /app/build
13
14 FROM build AS publish
15 RUN dotnet publish "web1.csproj" -c Release -o /app/publish
16
17 FROM base AS final
18 WORKDIR /app
19 COPY --from=publish /app/publish .
20 ENTRYPOINT ["dotnet", "web1.dll"]
21
```

The interface also shows a list of Docker images on the left, including 'nginx:latest' and 'mongo'. The 'REGISTRIES' section shows 'Docker Hub'.

## 4 Open a new terminal in VS Code

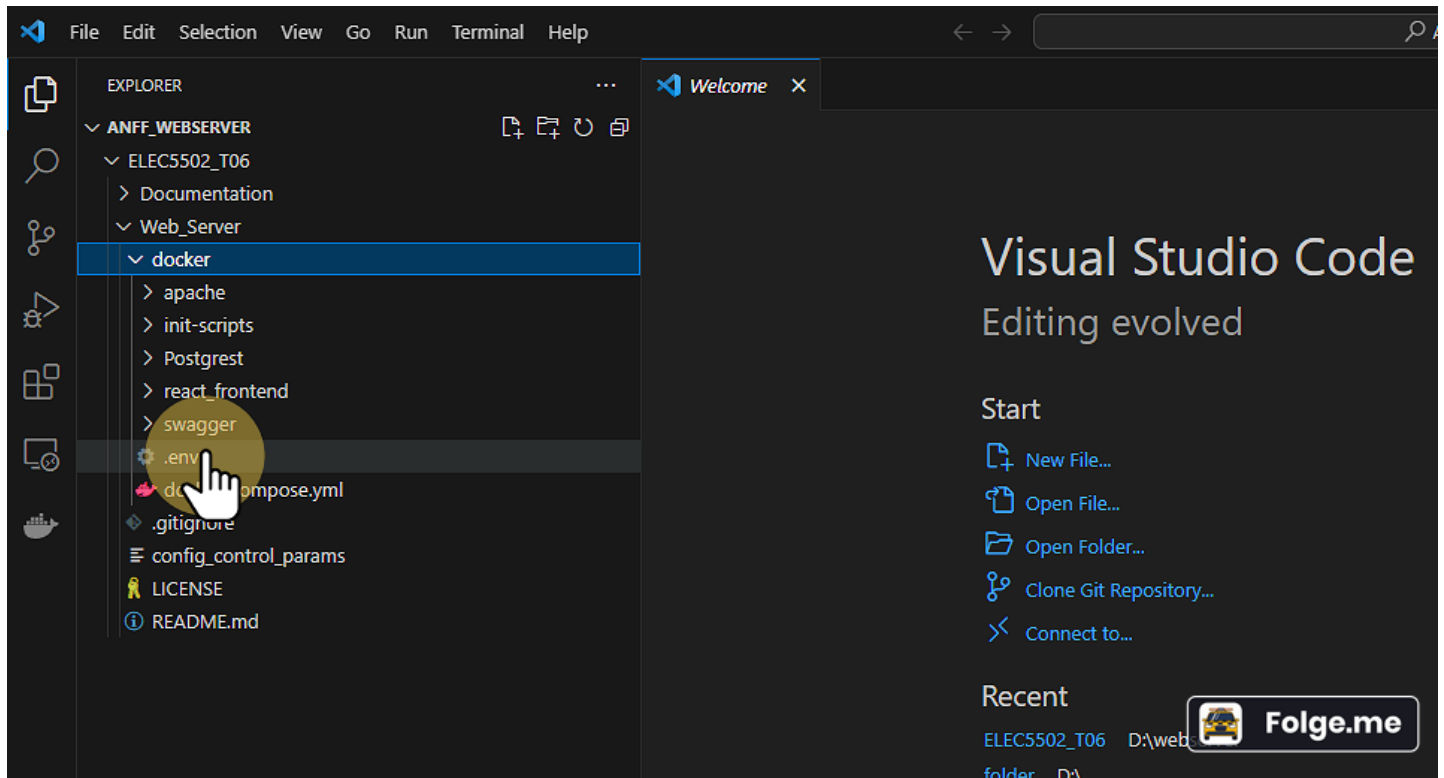


- 5 Run the command : git clone  
`https://github.com/EkkIGI/ELEC5502_T06.git`

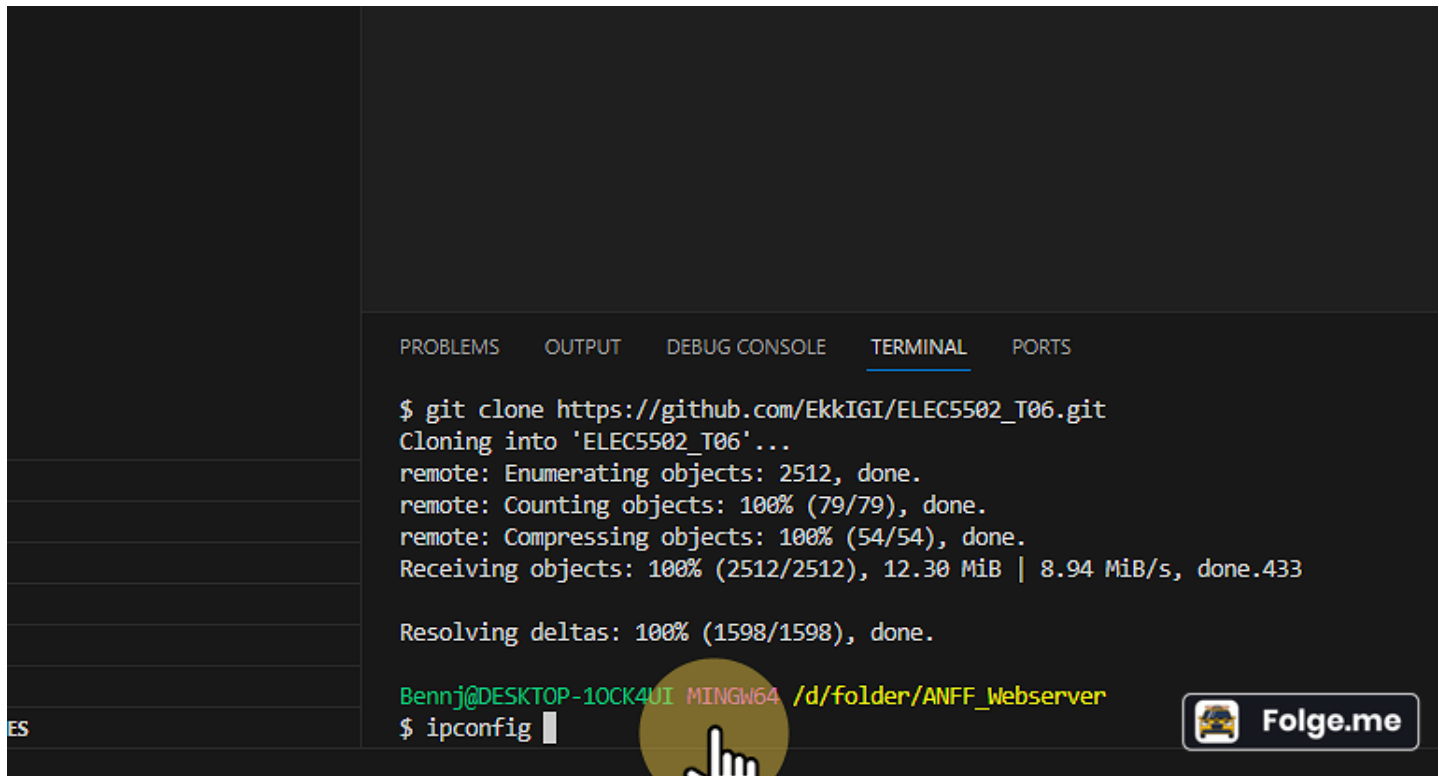




## 6 open the .env file in the docker folder in web\_server



## 7 Run the command ipconfig in the terminal



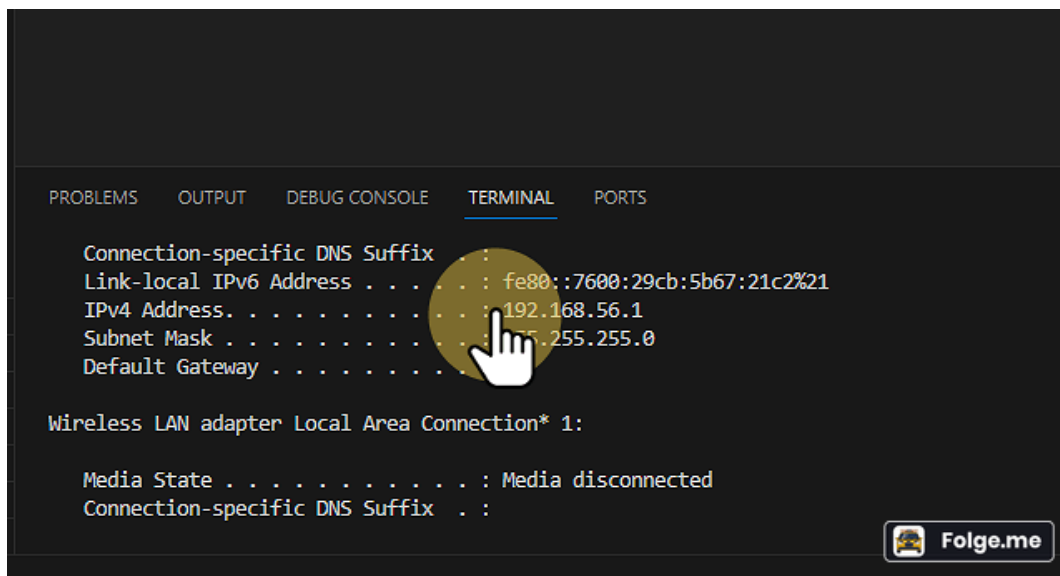
The screenshot shows a terminal window with a dark background. At the top, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS. The terminal output shows the execution of a git clone command, followed by progress bars for cloning, enumerating, counting, and compressing objects, and finally receiving objects. Below this, it shows the resolution of deltas. The prompt then changes to Bennis@DESKTOP-10CK4UI MINGW64 /d/folder/ANFF\_Webserver, and the command \$ ipconfig is entered. A yellow circle with a hand icon points to the command. In the bottom right corner, there is a logo for Folge.me.

```
$ git clone https://github.com/EkkIGI/ELEC5502_T06.git
Cloning into 'ELEC5502_T06'...
remote: Enumerating objects: 2512, done.
remote: Counting objects: 100% (79/79), done.
remote: Compressing objects: 100% (54/54), done.
Receiving objects: 100% (2512/2512), 12.30 MiB | 8.94 MiB/s, done.433

Resolving deltas: 100% (1598/1598), done.

Bennis@DESKTOP-10CK4UI MINGW64 /d/folder/ANFF_Webserver
$ ipconfig
```

## 8 Locate the IPv4 Address and copy it



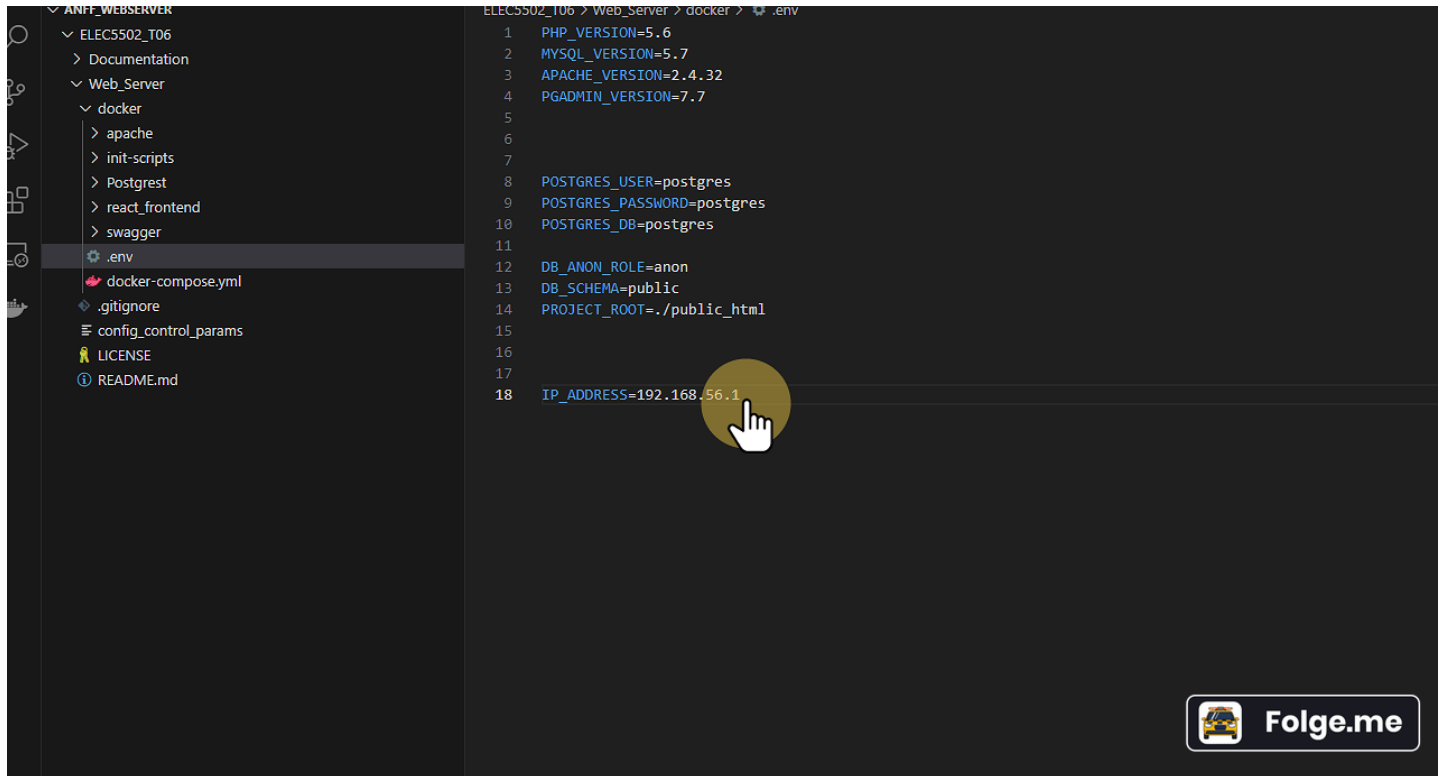
The screenshot shows a terminal window with a dark background. At the top, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS. The terminal output shows the output of the ipconfig command, displaying network configuration details for a wireless LAN adapter. A yellow circle with a hand icon points to the IPv4 address, 192.168.56.1. In the bottom right corner, there is a logo for Folge.me.

```
Connection-specific DNS Suffix . : 
Link-local IPv6 Address . . . . . : fe80::7600:29cb:5b67:21c2%21
IPv4 Address. . . . . : 192.168.56.1
Subnet Mask . . . . . : 255.255.0
Default Gateway . . . . . : 

Wireless LAN adapter Local Area Connection* 1:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
```

## 9 Paste the IPv4 address into the IP\_ADDRESS line

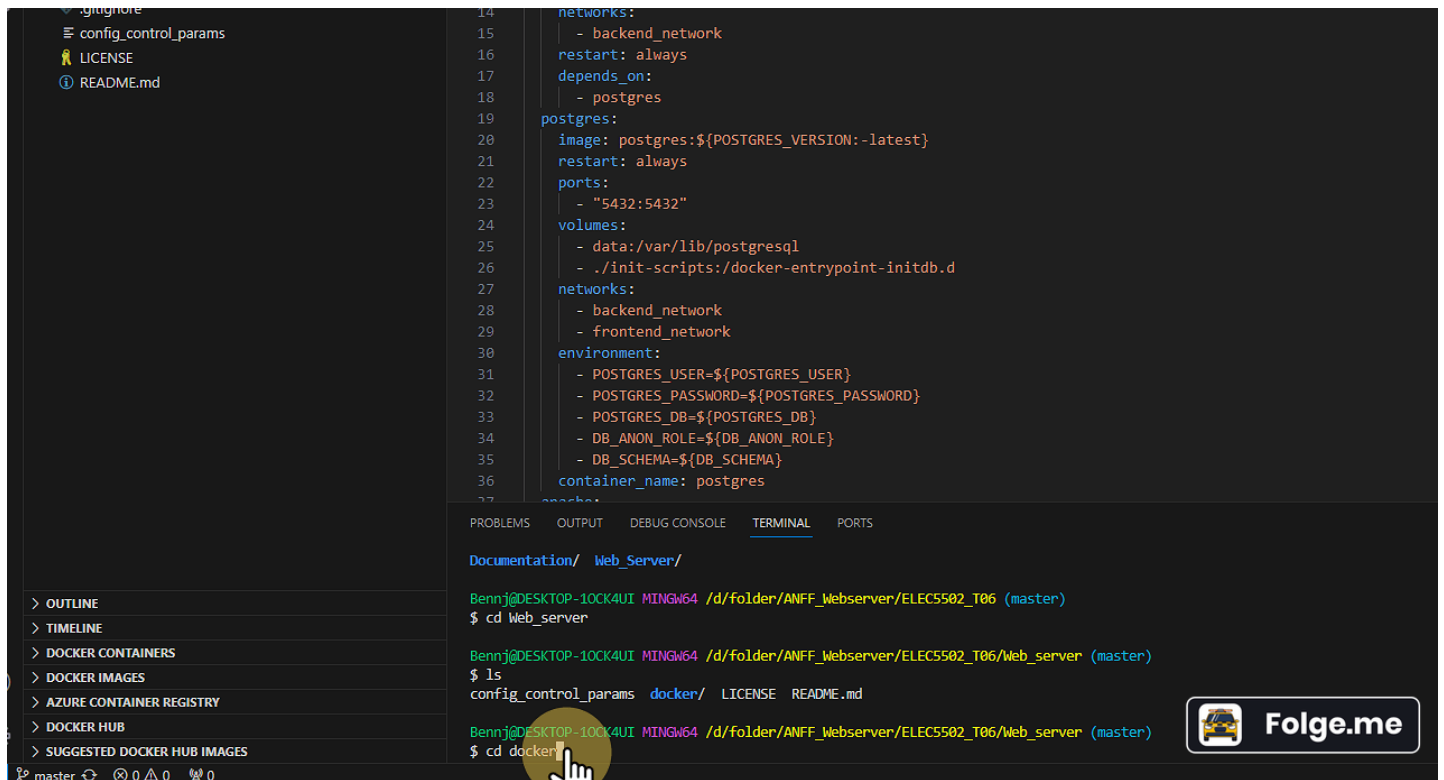


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'Documentation', 'Web\_Server', and 'docker'. The 'docker' folder is expanded, showing subfolders like 'apache', 'init-scripts', 'Postgres', 'react\_frontend', and 'swagger'. The '.env' file is selected. The code editor shows the contents of the '.env' file, which includes configuration for PHP, MySQL, Apache, and PostgreSQL. A hand cursor is pointing to the 'IP\_ADDRESS' line, which is currently set to '192.168.56.1'.

```
1 PHP_VERSION=5.6
2 MYSQL_VERSION=5.7
3 APACHE_VERSION=2.4.32
4 PGADMIN_VERSION=7.7
5
6
7
8 POSTGRES_USER=postgres
9 POSTGRES_PASSWORD=postgres
10 POSTGRES_DB=postgres
11
12 DB_ANON_ROLE=anon
13 DB_SCHEMA=public
14 PROJECT_ROOT=./public_html
15
16
17
18 IP_ADDRESS=192.168.56.1
```

Folge.me

## 10 In the terminal navigate to the docker directory



The screenshot shows a VS Code editor with a Docker Compose file open. The file defines a service named 'postgres' with the following configuration:

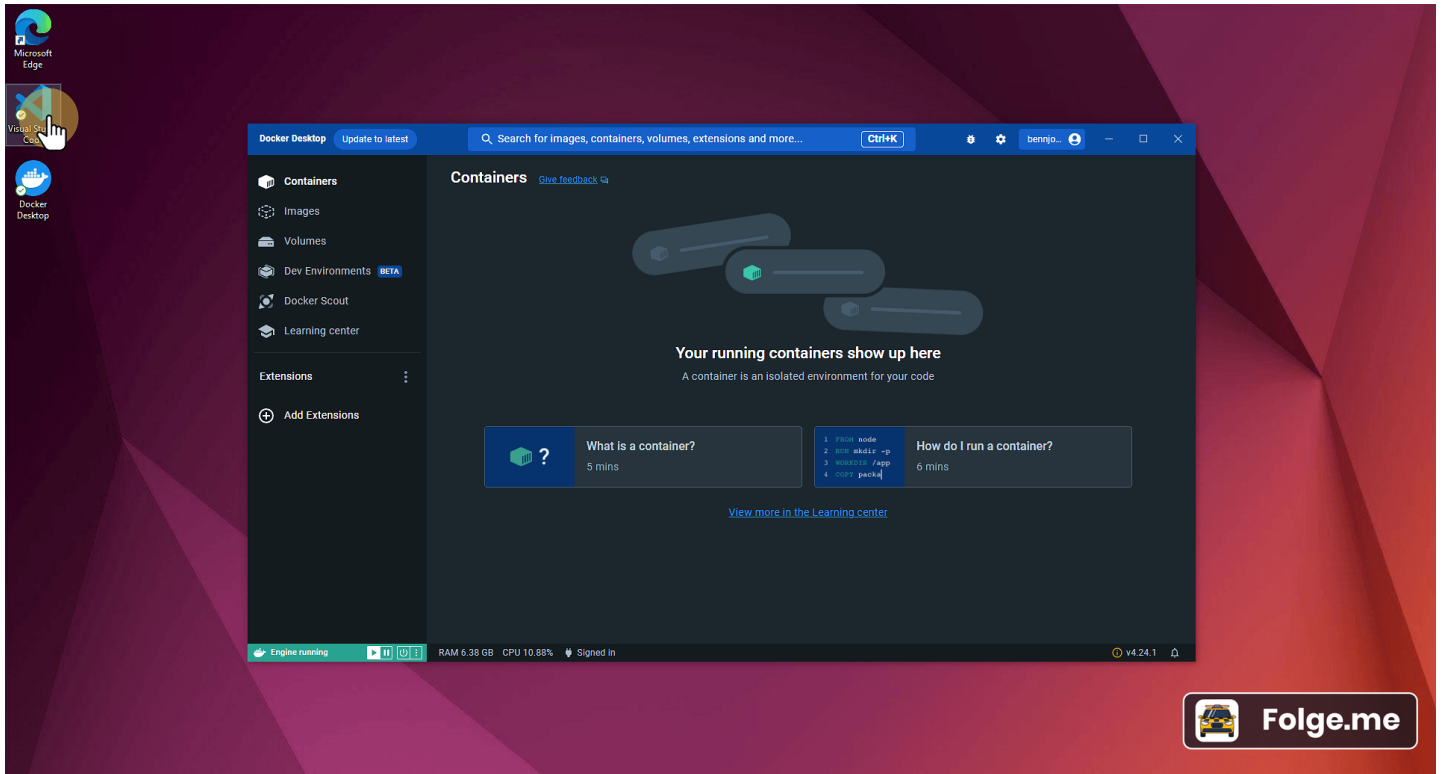
```
14 networks:
15   - backend_network
16 restart: always
17 depends_on:
18   - postgres
19 postgres:
20   image: postgres:${POSTGRES_VERSION:-latest}
21   restart: always
22   ports:
23     - "5432:5432"
24   volumes:
25     - data:/var/lib/postgresql
26     - ./init-scripts:/docker-entrypoint-initdb.d
27   networks:
28     - backend_network
29     - frontend_network
30   environment:
31     - POSTGRES_USER=${POSTGRES_USER}
32     - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
33     - POSTGRES_DB=${POSTGRES_DB}
34     - DB_ANON_ROLE=${DB_ANON_ROLE}
35     - DB_SCHEMA=${DB_SCHEMA}
36   container_name: postgres
```

The terminal window at the bottom shows the following commands and output:

```
Documentation/ Web_Server/
Bennj@DESKTOP-10CK4UI MINGW64 /d/folder/ANFF_Webserver/ELEC5502_T06 (master)
$ cd Web_server
Bennj@DESKTOP-10CK4UI MINGW64 /d/folder/ANFF_Webserver/ELEC5502_T06/Web_server (master)
$ ls
config_control_params docker/ LICENSE README.md
Bennj@DESKTOP-10CK4UI MINGW64 /d/folder/ANFF_Webserver/ELEC5502_T06/Web_server (master)
$ cd docker
```

A hand cursor icon is pointing at the 'cd docker' command in the terminal. A 'Folge.me' logo is visible in the bottom right corner of the terminal window.

## 11 Open Docker Desktop



## 12 Run the following command in the terminal : docker-compose up -d

```
23     - 5432:5432
24     volumes:
25       - data:/var/lib/postgresql
26       - ./init-scripts:/docker-entrypoint-initdb.d
27     networks:
28       - backend_network
29       - frontend_network
30     environment:
31       - POSTGRES_USER=${POSTGRES_USER}
32       - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
33       - POSTGRES_DB=${POSTGRES_DB}
34       - DB_ANON_ROLE=${DB_ANON_ROLE}
35       - DB_SCHEMA=${DB_SCHEMA}
36     container_name: postgres
37     #networks:


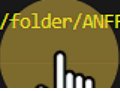
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

$ cd Web_server

Bennj@DESKTOP-10CK4UI MINGW64 /d/folder/ANFF_Webserver/ELEC5502_T06/Web_server (master)
$ ls
config_control_params  docker/  LICENSE  README.md

Bennj@DESKTOP-10CK4UI MINGW64 /d/folder/ANFF_Webserver/ELEC5502_T06/Web_server (master)
$ cd docker

Bennj@DESKTOP-10CK4UI MINGW64 /d/folder/ANFF_Webserver/ELEC5502_T06/Web_server/docker (master)
$ docker-compose up -d
```



## 13

The screenshot shows a Visual Studio Code editor with a Docker Compose file open. The file is named `docker-compose.yml` and is located in the `docker` directory. The file contains the following configuration:

```
14 networks:
15   - backend_network
16 restart: always
17 depends_on:
18   - postgres
19 postgres:
20   image: postgres:${POSTGRES_VERSION:-latest}
21   restart: always
22   ports:
23     - "5432:5432"
24   volumes:
25     - data:/var/lib/postgresql
26     - ./init-scripts:/docker-entrypoint-initdb.d
27 networks:
28   - backend_network
29   - frontend_network
30 environment:
31   - POSTGRES_USER=${POSTGRES_USER}
32   - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
33   - POSTGRES_DB=${POSTGRES_DB}
34   - DB_ANON_ROLE=${DB_ANON_ROLE}
35   - DB_SCHEMA=${DB_SCHEMA}
36 container_name: postgres
37
```

The status bar at the bottom shows the following information:

- PROBLEMS: 0
- OUTPUT: 0
- DEBUG CONSOLE: 0
- TERMINAL: 0
- PORTS: 0

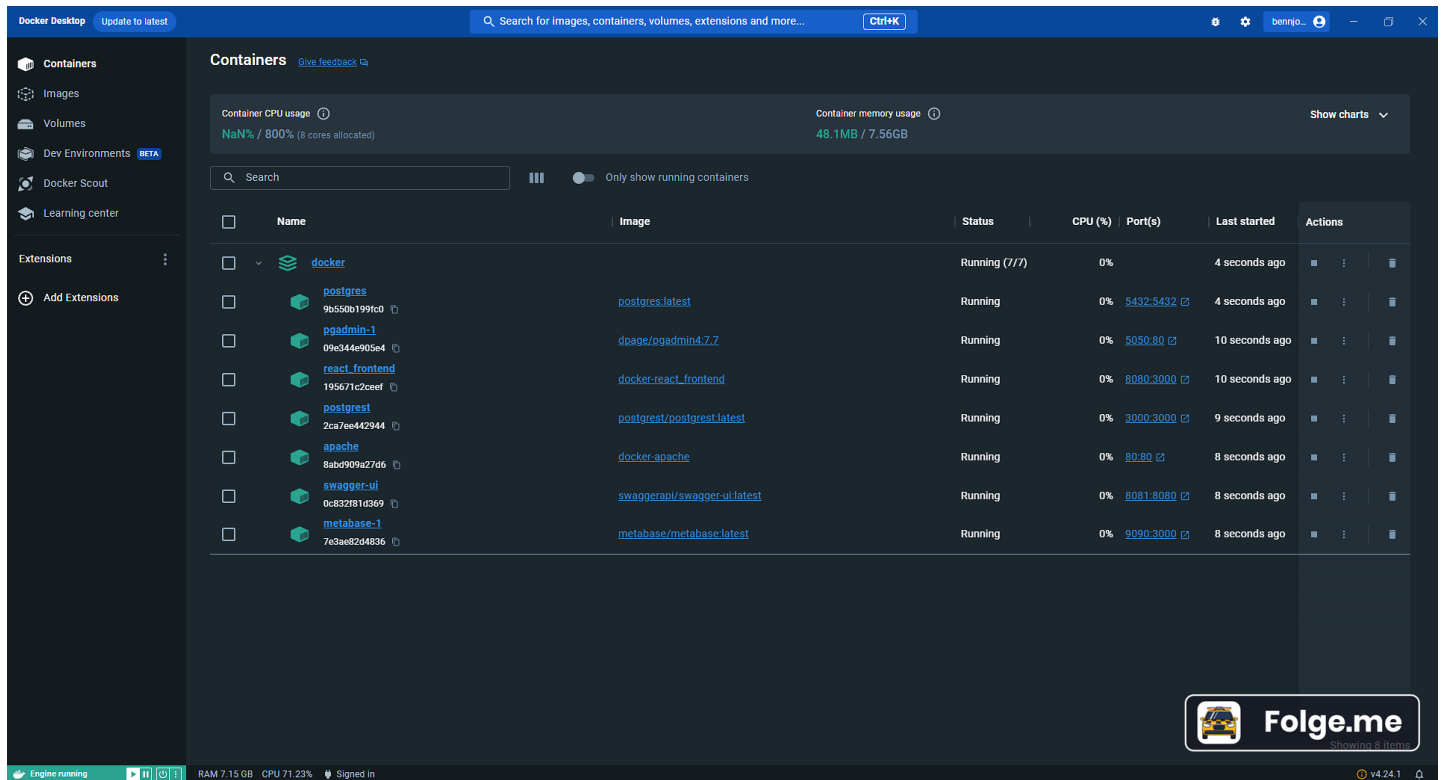
The terminal output shows the following status:

- ✓ Network docker\_backend\_network Created
- ✓ Container react\_frontend Started
- ✓ Container docker-pgadmin-1 Started
- ✓ Container postgres Started
- ✓ Container postgrest Started
- ✓ Container docker-metabase-1 Started
- ✓ Container swagger-ui Started
- ✓ Container apache Started

The terminal prompt is `Bennj@DESKTOP-10CK4UI MINGW64 /d:/folder/ANFF_Webserver/ELEC5502_T06/Web_server/docker (master)`.

A **Folge.me** logo is visible in the bottom right corner of the terminal window.

## 14 Open Docker to see if all the containers are running

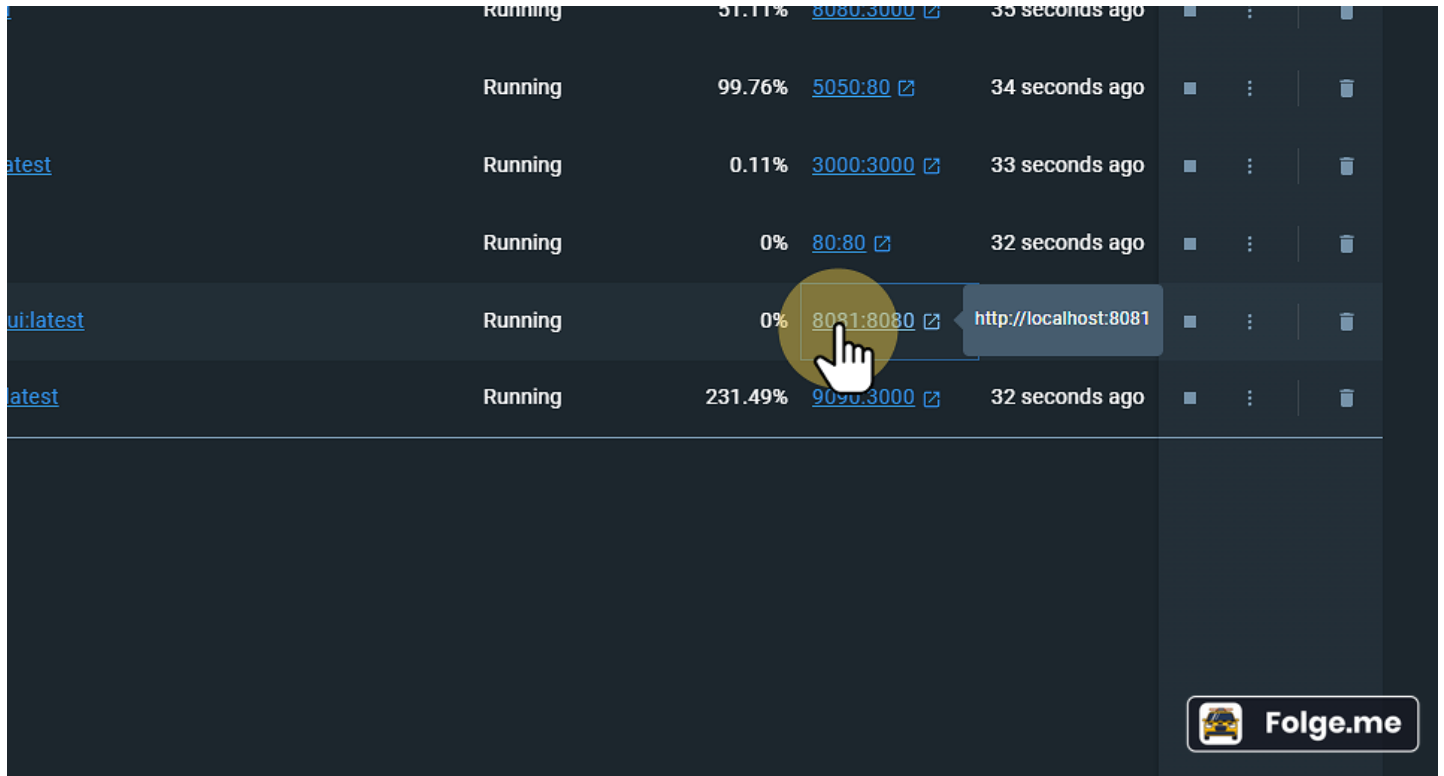


The screenshot shows the Docker Desktop application window. The left sidebar contains navigation options: Containers, Images, Volumes, Dev Environments (marked BETA), Docker Scout, and Learning center. The main area is titled 'Containers' and displays a table of running containers. Above the table, there are summary statistics: 'Container CPU usage' at NaN% / 800% (8 cores allocated) and 'Container memory usage' at 48.1MB / 7.56GB. A search bar and a toggle for 'Only show running containers' are also present. The table lists several containers, including 'docker', 'postgres', 'pgadmin-1', 'react\_frontend', 'postgres', 'apache', 'swagger-ui', and 'metabase-1', all with a status of 'Running'. A 'Folge.me' watermark is visible in the bottom right corner of the interface.

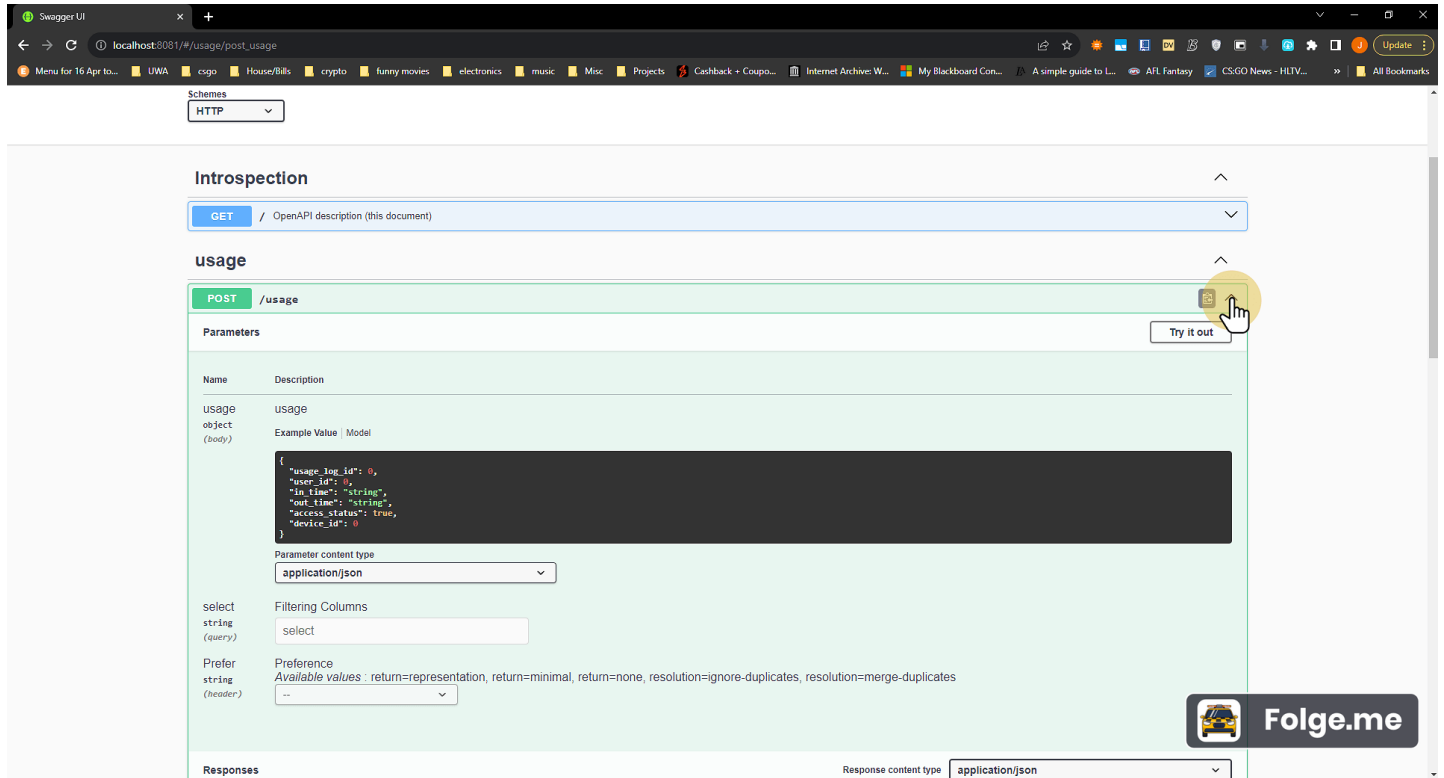
Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
docker		Running (7/7)	0%		4 seconds ago	
postgres	postgres:latest	Running	0%	5432:5432	4 seconds ago	
pgadmin-1	dpape/pgadmin4:7.7	Running	0%	5050:80	10 seconds ago	
react_frontend	docker-react_frontend	Running	0%	8080:3000	10 seconds ago	
postgres	postgres/postgres:latest	Running	0%	3000:3000	9 seconds ago	
apache	docker-apache	Running	0%	80:80	8 seconds ago	
swagger-ui	swaggerapi/swagger-ui:latest	Running	0%	8081:8080	8 seconds ago	
metabase-1	metabase/metabase:latest	Running	0%	9090:3000	8 seconds ago	



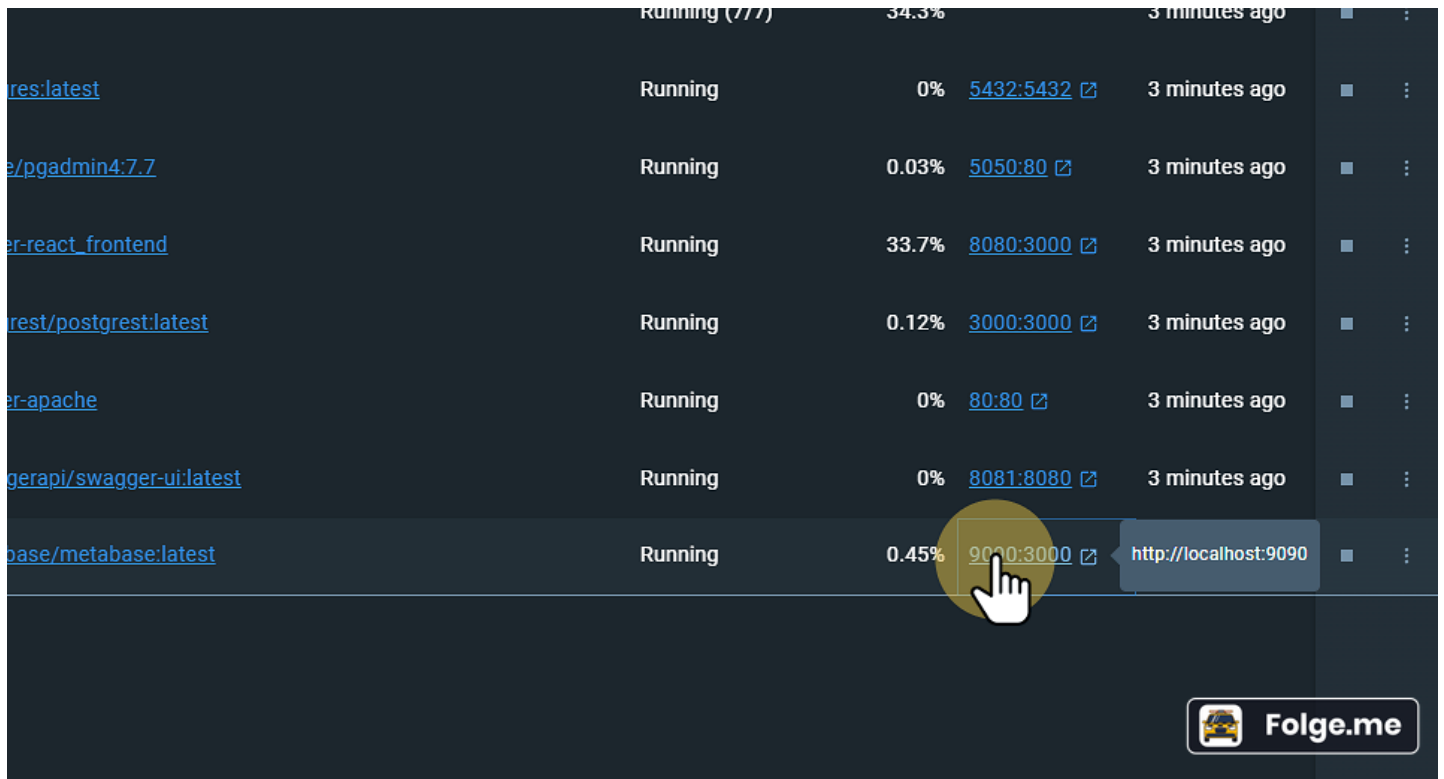
## 15 Click on the port address of Swagger-UI to open the webpage



## 16 Swagger-UI will open the API endpoint webpage in a new browser window



## 17 In Docker desktop, click on the metabase port address to open the Metabase webpage



Running (7/7)	34.3%	3 minutes ago	
<a href="#">postgres:latest</a>	Running	0%	<a href="#">5432:5432</a> 3 minutes ago
<a href="#">postgres/pgadmin4:7.7</a>	Running	0.03%	<a href="#">5050:80</a> 3 minutes ago
<a href="#">metabase-react_frontend</a>	Running	33.7%	<a href="#">8080:3000</a> 3 minutes ago
<a href="#">metabase/postgrest:latest</a>	Running	0.12%	<a href="#">3000:3000</a> 3 minutes ago
<a href="#">metabase-apache</a>	Running	0%	<a href="#">80:80</a> 3 minutes ago
<a href="#">metabase-api/swagger-ui:latest</a>	Running	0%	<a href="#">8081:8080</a> 3 minutes ago
<a href="#">metabase/metabase:latest</a>	Running	0.45%	<a href="#">9090:3000</a> <a href="#">http://localhost:9090</a> 3 minutes ago

Folge.me

18



## Welcome to Metabase

Looks like everything is working. Now let's get to know you, connect to your data, and start finding you some answers!



## 19

This language will be used throughout Metabase and will be the default for new users.

Chinese (China)  
Chinese (Hong Kong SAR China)  
Chinese (Taiwan)  
Czech  
Dutch  
English  
French  
German  
Indonesian  
Italian



2 What should we call you?

3 Add your data



Folge.me

## 20 Fill in personal details

2

### What should we call you?

First name

postgres

Last name

postgres

Email

22373566@student.uwa.edu.au

Company or team name

uwa

Create a password

••••••••

Confirm your password

••••••••



Folge.me

## 21 Select the PostgreSQL database

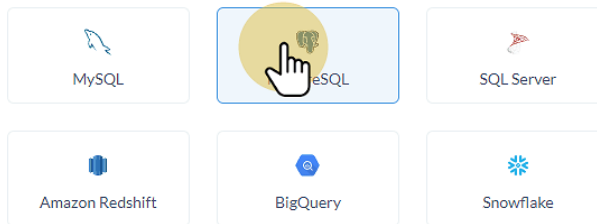
✓ Hi, postgres. Nice to meet you!

3

### Add your data

Are you ready to start exploring your data? Add it below.  
Not ready? Skip and play around with our Sample Database.

Search for a database...



Show more options ▾

[I'll add my data later](#)



## 22 Fill out database details.

Display name : ANFF web server

Host : postgres

Port : 5432

Database name : postgres

Username : postgres

Password : postgres

These can be changed in the .env file

PostgreSQL

Display name  
postgres

Host  
postgres

Port  
5432

Database name  
postgres

Username  
postgres

Password  
••••••••

Schemas  
All

Use a secure connection (SSL) ☐

Use an SSH tunnel ☐  
If a direct connection to your database isn't possible, you may want to use an SSH tunnel. [Learn more.](#)

Show advanced options ▾

Skip Connect database

Folge.me



## 23 Follow the prompts



Connecting to postgres

4

### Usage data preferences

In order to help us improve Metabase, we'd like to collect certain data about product usage. [Here's a full list of what we track and why.](#)



Allow Metabase to anonymously collect usage events

- Metabase never collects anything about your data or question results.
- All collection is completely anonymous.
- Collection can be turned off at any point in your admin settings.

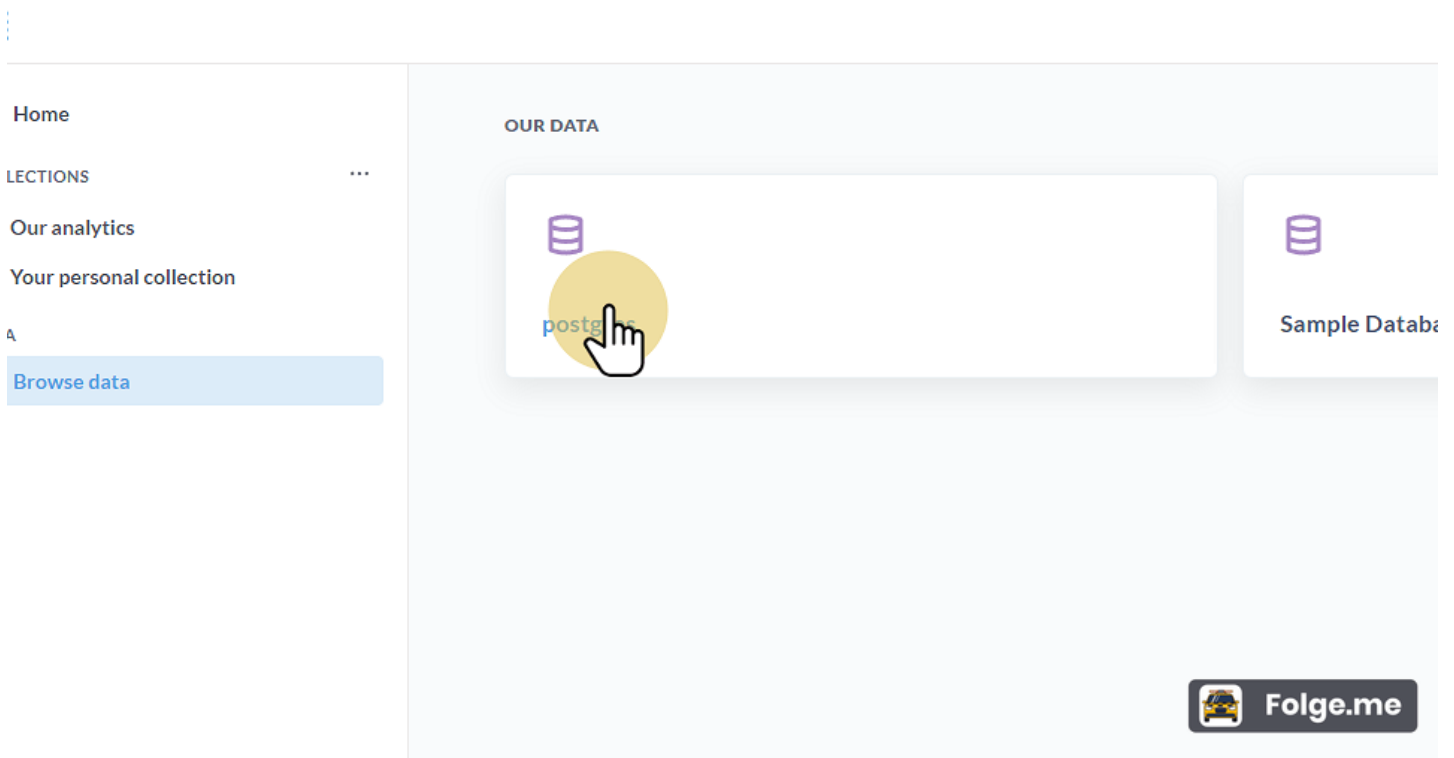


If you feel stuck, [our getting started guide](#) is just a click away.



































Folge.me

## 24 Navigate to the correct database



## 25 Open docker and click on the PGAdmin port address to open the PGAdmin webpage

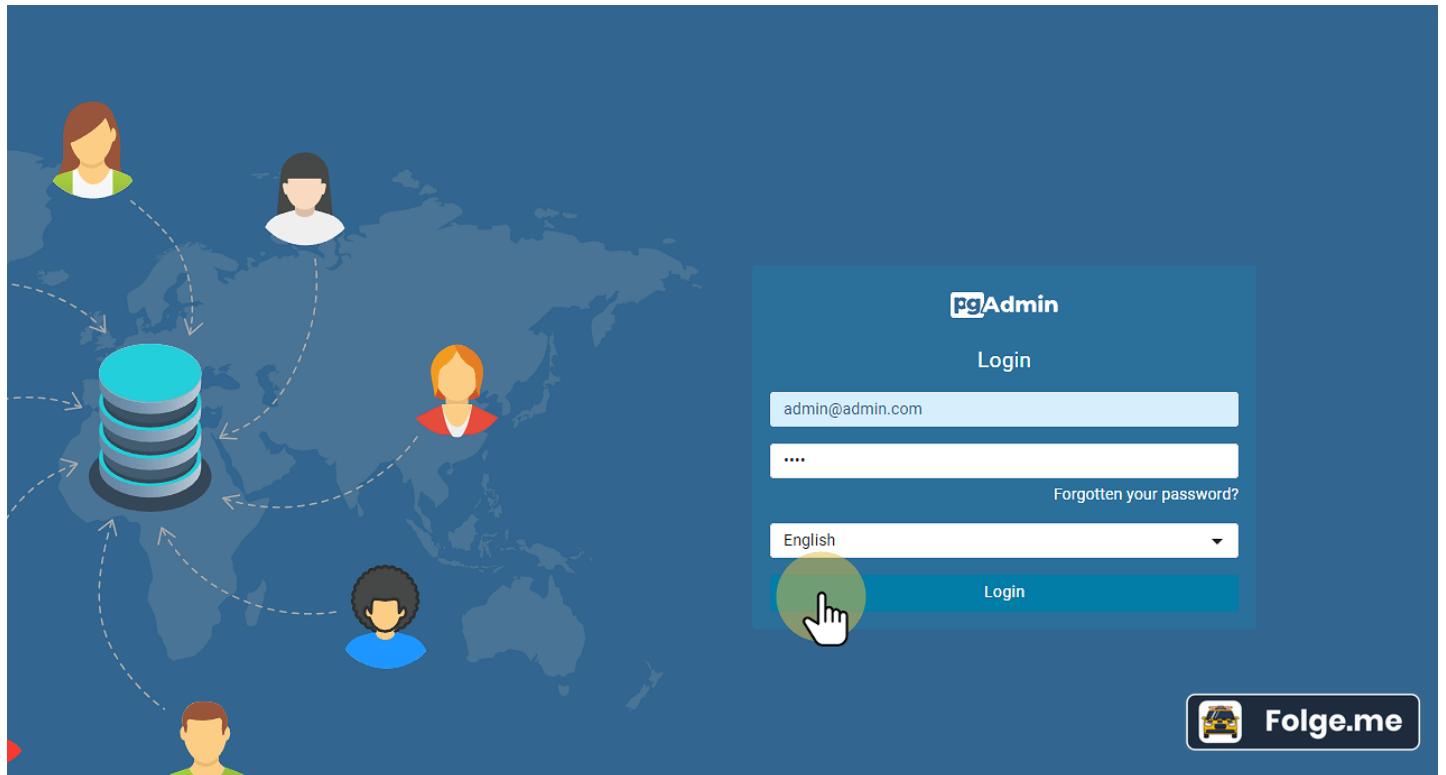
	Status	CPU (%)	Port(s)	Last started	Actions
	Running (7/7)	23.6%		5 minutes ago	  
	Running	0.02%	<a href="#">5432:5432</a> 	5 minutes ago	  
	Running	0.04%	<a href="#">5050:5050</a>  <a href="http://localhost:5050">http://localhost:5050</a>		  
	Running	23.06%	<a href="#">8080:3000</a> 	5 minutes ago	  
<a href="#">latest</a>	Running	0.14%	<a href="#">3000:3000</a> 	5 minutes ago	  
	Running	0.01%	<a href="#">80:80</a> 	5 minutes ago	  
<a href="#">ui:latest</a>	Running	0%	<a href="#">8081:8080</a> 	5 minutes ago	  
<a href="#">latest</a>	Running	0.33%	<a href="#">9090:3000</a> 	5 minutes ago	  

 Folge.me

## 26 Fill out the log in details

Username : admin@admin.com

PW : root




## 27 Add server

explorer servers

Dashboard Properties SQL Statistics Dependencies Dependents Processes

Welcome

 **pgAdmin**  
Management Tools for PostgreSQL

Feature rich | Maximises PostgreSQL | Open Source


pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, and a set of utilities for developers, DBAs and system administrators alike.

Quick Links


 [Add New Server](#)

Getting Started

 PostgreSQL Documentation

 pgAdmin Website

 Planet PostgreSQL

 Folge.me

## 28 Fill out server name

The screenshot shows the pgAdmin 4 web interface with the 'Register - Server' dialog box open. The 'Name' field contains 'postgres', which is highlighted by a yellow circle and a hand cursor. The 'Server group' dropdown is set to 'Servers'. The 'Background' and 'Foreground' checkboxes are unchecked, while 'Connect now?' and 'Shared?' are checked. A red error message at the bottom of the dialog states: 'Either Host name or Service must be specified.' The background shows the pgAdmin 4 dashboard with various links and a 'Configure pgAdmin' button.

Register - Server

General Connection Parameters SSH Tunnel Advanced

Name postgres

Server group Servers

Background ☐

Foreground ☐

Connect now? ☒

Shared? ☒

Comments

⚠ Either Host name or Service must be specified.

## 29 Fill in the server details.


Host name/address : postgres

Username : postgres

Password : postgres

Dashboard Properties SQL Statistics Dependencies Dependents Processes

Welcome


 **pgAdmin**  
Management Tools for PostgreSQL

Feature rich | Maximises PostgreSQL productivity

pgAdmin is an Open Source administration and development tool, a procedural code debugger and a query editor.

Quick Links

Getting Started

 PostgreSQL Documentation

**Register - Server**

General Connection Parameters SSH Tunnel Advanced

Host name/address: postgres

Port: 5432

Maintenance database: postgres

Username: postgres

Kerberos authentication? ☐

Password: .....

Save password? ☐

Role:

Service:

