

Advanced Web Mapping: Assignment 1

Student number: D18130495

Student name: Yushun Zeng

Website link: <https://www.awm2023.site>

Pgadmin4 link: <https://www.awm2023.site/pgadmin4>

Username: D18130495@mytudublin.ie

Password: 123456

Connection:

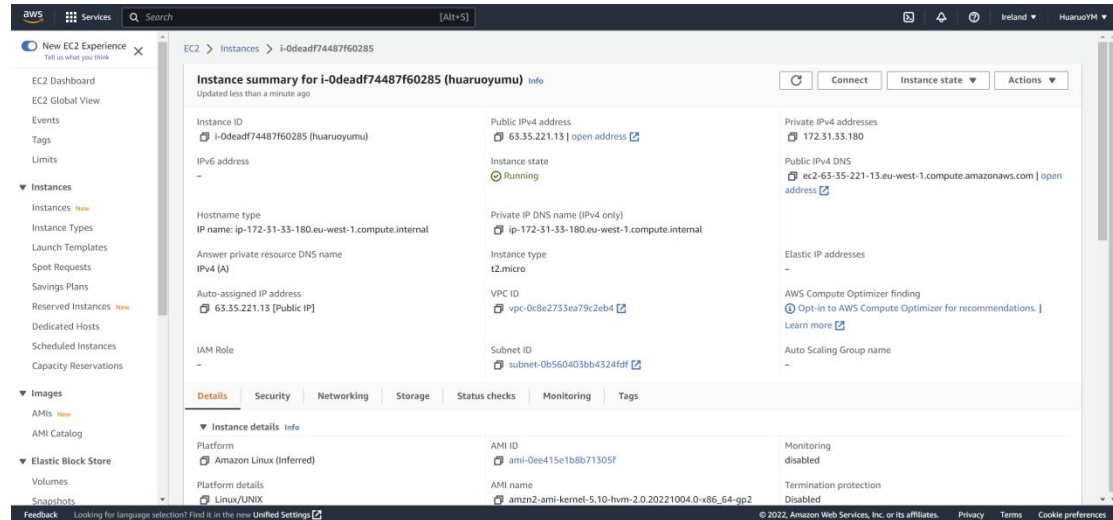
Host: 63.35.221.13 Port: 25432

Database: gis Username: postgres Password: 123456

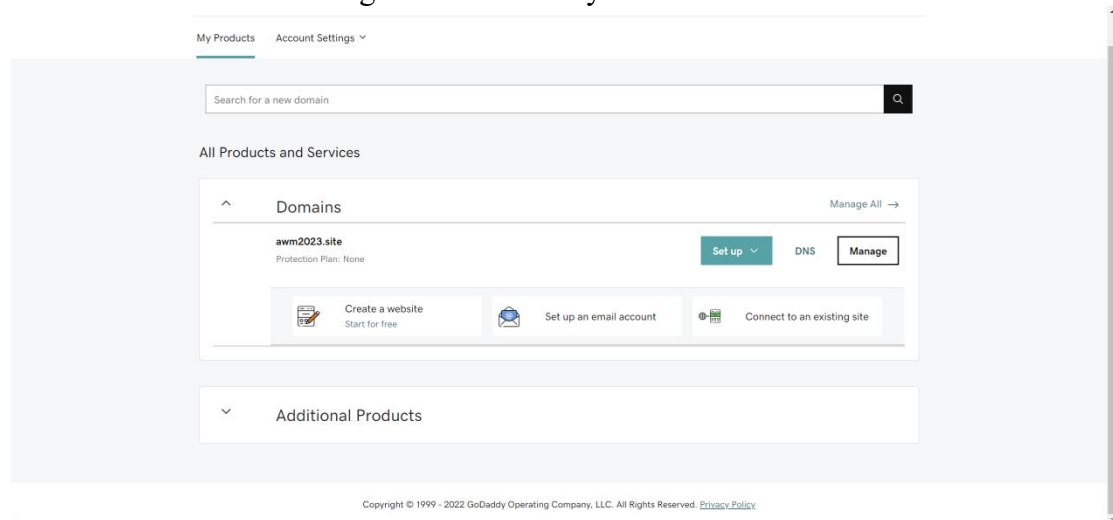
Github link: <https://github.com/D18130495/Advanced-Web-Mapping>

Deploy step:

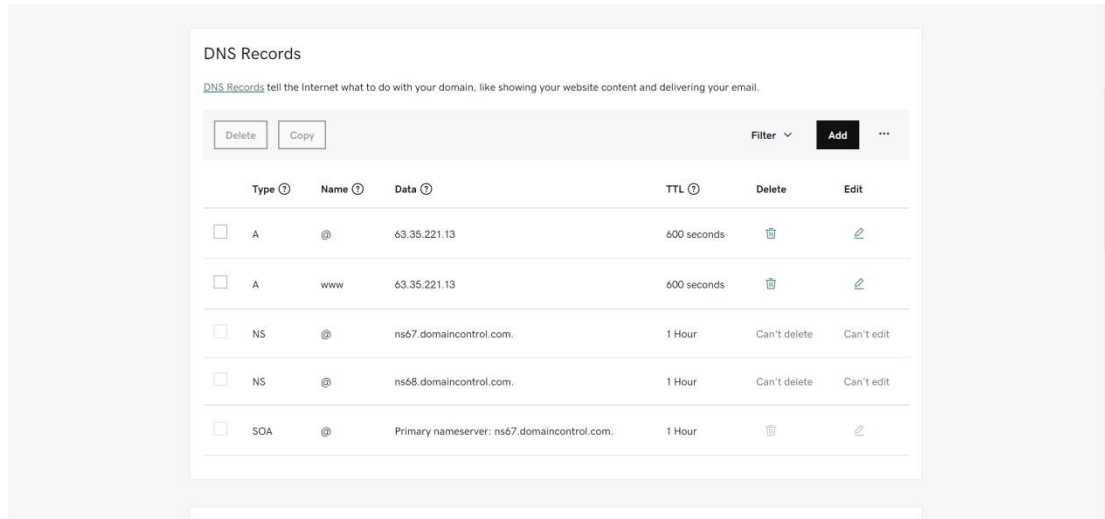
1. The could server I used is Amazon AWS.



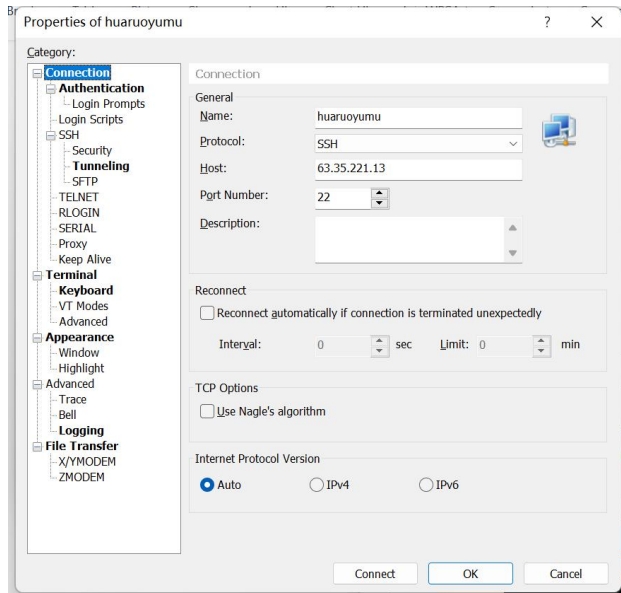
2. The domain name I brought from GoDaddy.



3. Added the domain name's DNS.



4. Connect to the cloud server var the ssh connection.



```
Xshell 7 (Build 0100)
Copyright (c) 2020 NetSarang Computer, Inc. All rights reserved.

Type 'help' to learn how to use Xshell prompt.
[C:\-]$

Connecting to 63.35.221.13:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+j'.

WARNING! The remote SSH server rejected X11 forwarding request.
Last failed login: Tue Nov  8 11:46:33 UTC 2022 from 72.22.164.247 on ssh:notty
There were 688 failed login attempts since the last successful login.
Last login: Fri Nov  4 18:21:54 2022 from 64.43.50.21

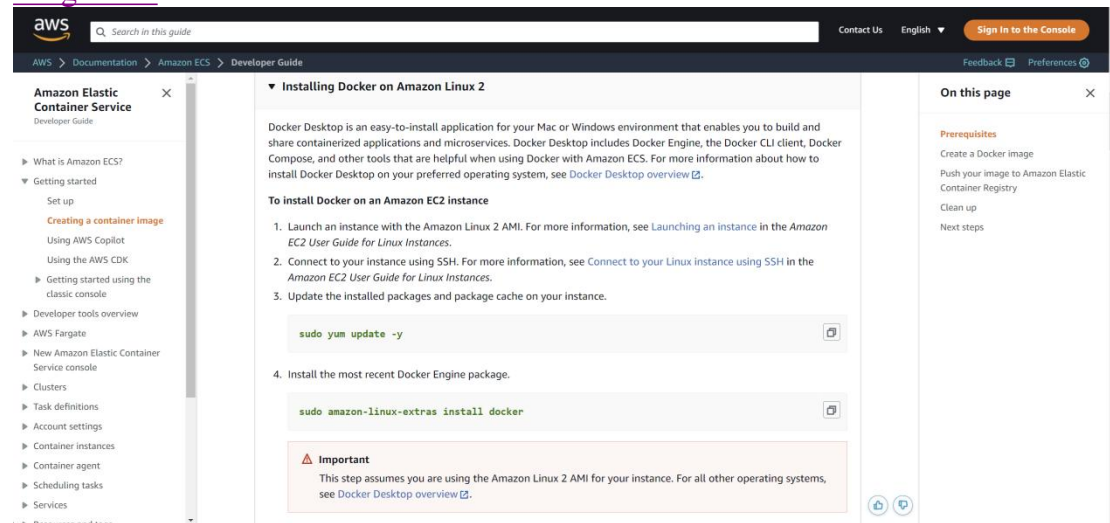
 _ _ _ _ _
| |   | |   | |   | |   | |
|_|  |_|  |_|  |_|  |_|  |_|

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[root@ip-172-31-33-180 ~]#
```

5. Install Docker and create network.

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/create-container-image.html>



Create new docker network by using docker network create awm2023

```
[root@ip-172-31-33-180 ~]# docker network list
NETWORK ID          NAME        DRIVER  SCOPE
8b3c87a221d1        awm2023    bridge  local
b608fc7e3b58        bridge     bridge  local
b842df81f6f2        host       host    local
1f714291e778        none       null    local
[root@ip-172-31-33-180 ~]#
```

6. Create docker file for get Nginx Certbot image.

```
[root@ip-172-31-33-180 awm2023]# cat Dockerfile
FROM nginx
MAINTAINER Yushun Zeng
RUN apt-get -y update && apt-get -y upgrade && apt-get -y install software-properties-common certbot python3-certbot-nginx
[root@ip-172-31-33-180 awm2023]#
```

Run the docker file to get Nginx_Certbot image and also get other needed images(pgadmin4, postgis).

```
[root@ip-172-31-33-180 awm2023]# docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
awm2023_nginx_certbot latest      ebbae36a92a0 3 days ago   249MB
nginx                latest      76c69feac34e 2 weeks ago  142MB
dpage/pgadmin4       latest      173beb2d4f13 3 weeks ago  370MB
kartoza/postgis      latest      acb761fa7225 16 months ago 1.65GB
[root@ip-172-31-33-180 awm2023]#
```

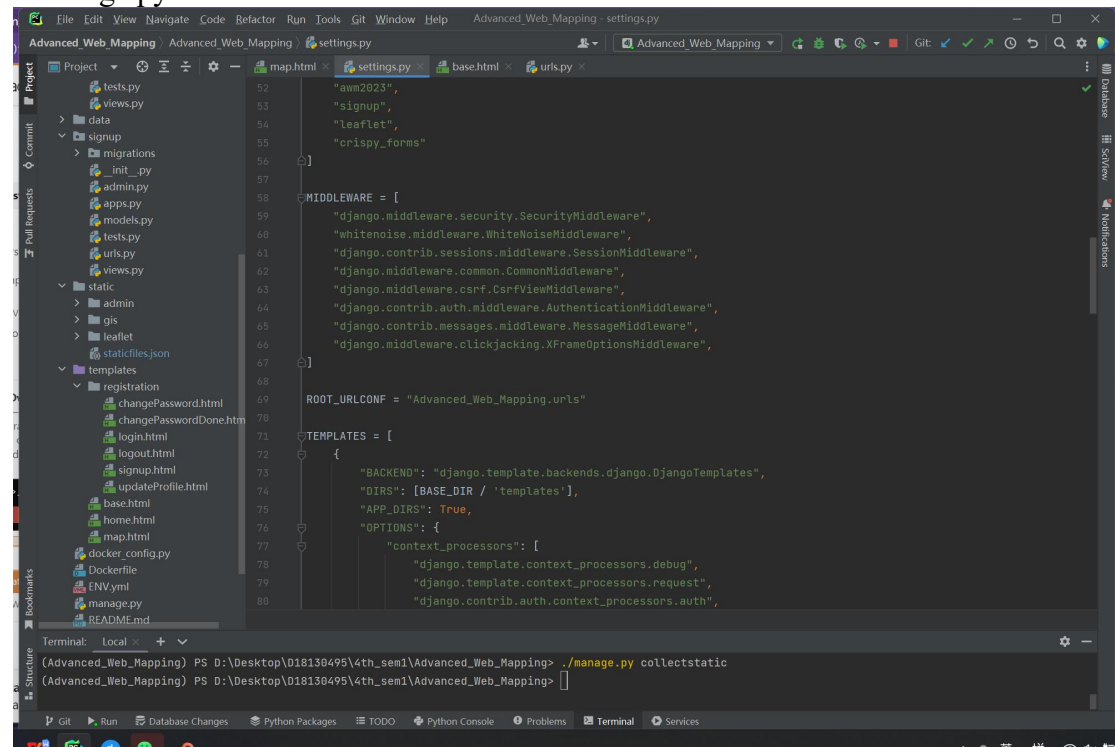
Then create the container for Nginx_Certbot, pgadmin4 and postgis under the awm2023 network.

```
[root@ip-172-31-33-180 awm2023]# docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED    STATUS    PORTS                               NAMES
1adfb57719b5   awm2023_nginx_certbot              "/docker-entrypoint..." 3 days ago Up 3 days 0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp  awm2023_nginx_certbot
564a22f704c    dpage/pgadmin4                     "/entrypoint.sh"         4 days ago Up 3 days 443/tcp, 0.0.0.0:20080->80/tcp, :::20080->80/tcp  awm2023_pgadmin4
1080901cfb76   kartoza/postgis                    "/bin/sh -c /scripts..." 4 days ago Up 3 days 0.0.0.0:25432->5432/tcp, :::25432->5432/tcp  awm2023_postgis
[root@ip-172-31-33-180 awm2023]#
```

7. Update Django project for deployment.

First, update whitenoise setting and generate static folder by using:

`./manage.py collectstatic`



Second, update setting.py to detect if the application run on the local host or cloud server and to enable some setting.

```
142 if socket.gethostname() == "192.168.192.1":
143     DATABASES["default"]["HOST"] = "localhost"
144     DATABASES["default"]["PORT"] = docker_config.POSTGIS_PORT
145 else:
146     DATABASES["default"]["HOST"] = docker_config.CLOUD_POSTGIS_HOST
147     DATABASES["default"]["PORT"] = docker_config.CLOUD_POSTGIS_PORT
148
149 # Set DEPLOY_SECURE to True only for LIVE deployment
150 if docker_config.DEPLOY_SECURE:
151     DEBUG = False
152     TEMPLATES[0]["OPTIONS"]["debug"] = False
153     ALLOWED_HOSTS = ['www.awm2023.site', '63.35.221.13', '127.0.0.1']
154     CSRF_COOKIE_SECURE = True
155     SESSION_COOKIE_SECURE = True
156 else:
157     DEBUG = True
158     TEMPLATES[0]["OPTIONS"]["debug"] = True
159     ALLOWED_HOSTS = ['*', ]
160     CSRF_COOKIE_SECURE = False
161     SESSION_COOKIE_SECURE = False
162
```

```
map.html x settings.py x docker_config.py x base.html x urls.py x
1 POSTGIS_PORT = 25432
2 CLOUD_POSTGIS_HOST = "63.35.221.13"
3 CLOUD_POSTGIS_PORT = 25432
4 DEPLOY_SECURE = True
```

Third, update SECRET_KEY for the application for security reason.

```
33 # SECURITY WARNING: keep the secret key used in production secret!
34 with open('secret_key.txt') as f:
35     SECRET_KEY = f.read().strip()
```

```
map.html x settings.py x secret_key.txt x base.html x urls.py x
1 django-insecure--66)l$3pa!$xk$i^@k=^isi-1qns5sv1y!-9)6@cd^o7%f)34k^
```

8. Build the project image and push to the Docker Hub.

The docker file that I used to build project image:

<https://github.com/D18130495/Advanced-Web-Mapping/blob/main/Dockerfile>

Build project image by using docker build -t huaruoyumu/awm2023 .

```
[+] Building 10.9s (17/18)
[+] Building 14.8s (19/19) FINISHED
=> [internal] load build definition from Dockerfile 0.1s
=> == transferring dockerfile: 2.00kB 0.0s
=> [internal] load .dockerignore 0.0s
=> == transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/continuumio/miniconda3:latest 2.6s
=> [ 1/14] FROM docker.io/continuumio/miniconda3@sha256:977263e8d1e476972fddab1c75fe050dd3cd17626390e874448bd92721fd659b 0.0s
=> [internal] load build context 0.2s
=> == transferring context: 12.37MB 0.2s
=> CACHED [ 2/14] RUN apt-get -y update && apt-get -y upgrade 0.0s
=> CACHED [ 3/14] RUN conda update -n base conda && conda update -n base --all 0.0s
=> CACHED [ 4/14] RUN mkdir -p /usr/src/app 0.0s
=> CACHED [ 5/14] WORKDIR /usr/src/app 0.0s
=> CACHED [ 6/14] RUN apt-get -y install build-essential python3-cffi libcairo2 libpango-1.0-0 libpangocairo-1.0-0 libgdk-pixbuf2.0-0 libffi-dev shar 0.0s
=> CACHED [ 7/14] COPY ENV.yml /usr/src/app 0.0s
=> CACHED [ 8/14] RUN conda env create -n Advanced_Web_Mapping --file ENV.yml 0.0s
=> CACHED [ 9/14] RUN echo "conda activate Advanced_Web_Mapping" >> ~/.bashrc 0.0s
=> CACHED [10/14] RUN conda config --add channels conda-forge && conda config --set channel_priority strict 0.0s
=> CACHED [11/14] RUN cat ~/.condarc 0.0s
=> CACHED [12/14] RUN conda install uwsgi 0.0s
=> [13/14] COPY . /usr/src/app 0.5s
=> [14/14] RUN python manage.py collectstatic --no-input 10.9s
=> exporting to image 0.4s
=> == exporting layers 0.4s
=> == writing image sha256:e106fa804042a59dfc3e8f4a3a934fdd50f3cb94c2b219c24f071854af97071f 0.0s
=> == naming to docker.io/library/awm2023 0.0s
```

```
(Advanced_Web_Mapping) PS D:\Desktop\D18130495\4th_sem1\Advanced_Web_Mapping> docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
huaruoyumu/awm2023  latest         c8a75926f3a6   6 seconds ago  3.89GB
advanced_web_mapping latest         64a83950b02e   2 weeks ago    3.86GB
postgres            latest         901a82b310d3   3 weeks ago    377MB
dpage/pgadmin4      latest         94c0924749b6   7 weeks ago    366MB
kartoza/postgis     latest         acb761fa7225   16 months ago  1.65GB
(Advanced_Web_Mapping) PS D:\Desktop\D18130495\4th_sem1\Advanced_Web_Mapping>
```


Push the image to the Docker Hub:

First, log in the docker account on the image machine

```
root@HuaruoYM: ~  
root@HuaruoYM: # docker login  
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.  
Username: huaruoyumu  
Password:  
Login Succeeded  
  
Logging in with your password grants your terminal complete access to your account.  
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/  
root@HuaruoYM: ~#
```

Second, push the image to the Docker Hub by using:

`docker push huaruoyumu/awm2023:latest`

```
root@HuaruoYM: ~  
root@HuaruoYM: # docker push huaruoyumu/awm2023:latest  
The push refers to repository [docker.io/huaruoyumu/awm2023]  
5fc86e8c66a1: Layer already exists  
6c8c0e6d727d: Layer already exists  
0b47c9be0d13: Layer already exists  
0fbd30e77edb: Layer already exists  
6b8f14079087: Layer already exists  
29774d0898f3: Layer already exists  
450102cdea60: Layer already exists  
4c66ca6caeb6: Layer already exists  
24304bd82bf6: Layer already exists  
5f70bf18a086: Layer already exists  
9aa8cd744673: Layer already exists  
fc4217835d04: Layer already exists  
45991f52fe0d: Layer already exists  
ab2731ec3f53: Layer already exists  
6falf4185aa2: Layer already exists  
ad6562704f37: Layer already exists  
latest: digest: sha256:6ee147a448155c7e5cae7c3820190b4526808a69154578e918b4bee6c6051a5a size: 3683  
root@HuaruoYM: ~#
```

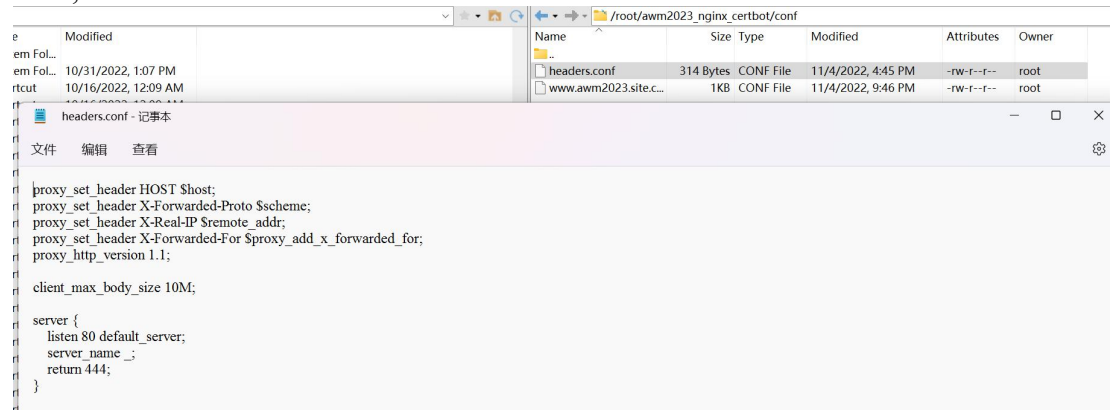
9. Get the pushed image to cloud server and create the container.

Use: `docker create --name awm2023 --network awm2023 --network-alias awm2023 -t huaruoyumu/awm2023`

```
REPOSITORY          TAG          IMAGE ID        CREATED         SIZE
awm2023/nginx_certbot latest       ebbae36a92a0    3 days ago     249MB
nginx               latest       76c69feac34e    2 weeks ago    142MB
dpkg/pgadmin4       latest       173beb2d4f13    3 weeks ago    370MB
kartoza/postgis      latest       acb761fa7225    16 months ago  1.65GB
[root@ip-172-31-33-180 ~]# docker create --name awm2023 --network awm2023 --network-alias awm2023 -t huaruoyumu/awm2023
Unable to find image 'huaruoyumu/awm2023:latest' locally
latest: Pulling from huaruoyumu/awm2023
42c077c10790: Pull complete
1a23c9d790a3: Pull complete
22a6fc63b9b5: Pull complete
6eb175953651: Pull complete
f8fde90e0fc5: Pull complete
5aecac2bd381: Pull complete
4f4fb700ef54: Pull complete
ae3cdf94dbcf: Pull complete
89e222ecd352: Pull complete
b9ee81e5263d: Pull complete
a105fdaa4dae: Pull complete
6c4433b3aeac: Pull complete
7623207220c1: Pull complete
21af924c38a6: Pull complete
82596472c0bd: Pull complete
cbdcbc4427fc: Pull complete
Digest: sha256:6ee147a448155c7e5cae7c3820190b4526808a69154578e918b4bee6c6051a5a
Status: Downloaded newer image for huaruoyumu/awm2023:latest
83b7e13acb2575adfd71875b6a16f9c28212d21d2e675c2320bdb9d1b2b0df1c
[root@ip-172-31-33-180 ~]# docker images
REPOSITORY          TAG          IMAGE ID        CREATED         SIZE
huaruoyumu/awm2023  latest       c8a75926f3a6    14 minutes ago  3.89GB
awm2023/nginx_certbot latest       ebbae36a92a0    3 days ago     249MB
nginx               latest       76c69feac34e    2 weeks ago    142MB
dpkg/pgadmin4       latest       173beb2d4f13    3 weeks ago    370MB
kartoza/postgis      latest       acb761fa7225    16 months ago  1.65GB
[root@ip-172-31-33-180 ~]#
```


10. Config the Nginx.

First, create the headers.conf.



Second, create www.awm2023.site.conf.

```
server {
    listen 80;
    server_name www.awm2023.site;

    location / {
        return 301 https://$host$request_uri;
    }

    location /.well-known/acme-challenge/ {
        root /var/www/certbot;
    }
}

server {
    listen 443 ssl;

    root /usr/share/nginx/html;
    index index.html;

    server_name www.awm2023.site;

    ssl_certificate /etc/letsencrypt/live/awm2023.site/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/awm2023.site/privkey.pem;

    #include /etc/letsencrypt/options-ssl-nginx.conf;
    #ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

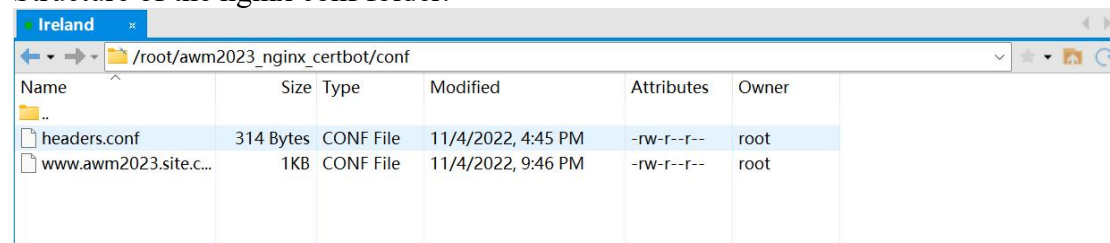
    location = /favicon.ico { access_log off; log_not_found off; }

    location /pgadmin4 {
        proxy_set_header X-Script-Name /pgadmin4;
        proxy_pass http://awm2023_pgadmin4;
    }

    # Some updates here
    # include uwsgi_params;
    location / {
        proxy_pass http://awm2023:8001;
    }

    # uwsgi_param Host $host;
    # uwsgi_param X-Real-IP $remote_addr;
    # uwsgi_param X-Forwarded-For $proxy_add_x_forwarded_for;
    # uwsgi_param X-Forwarded-Proto $http_x_forwarded_proto;
```

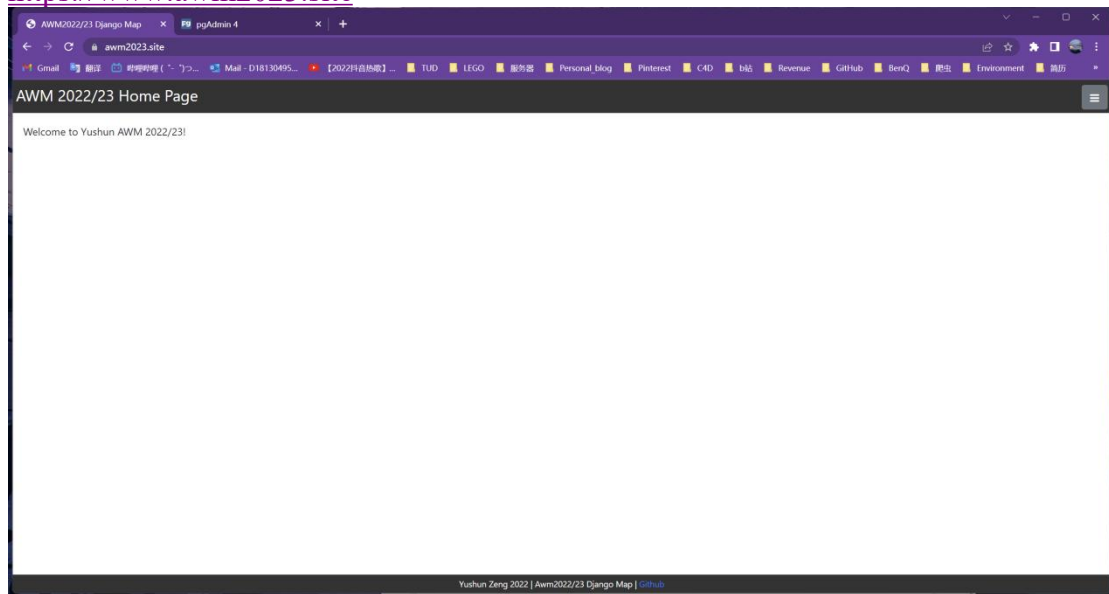
Structure of the nginx conf folder.



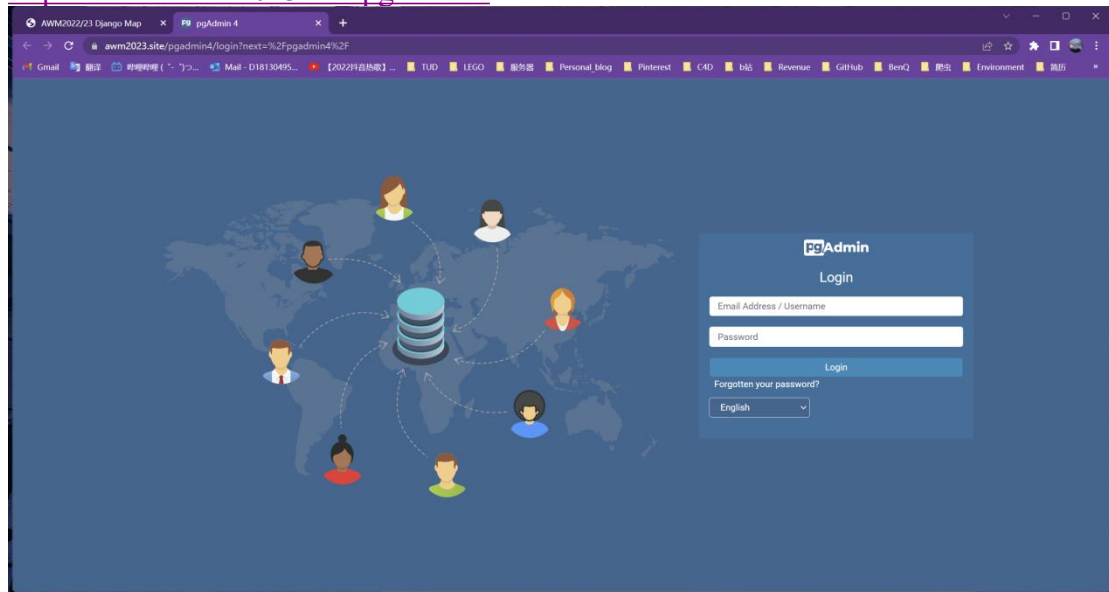
11. Run the postgres, pgadmin4, Django project and Nginx Certbot.

```
[root@ip-172-31-33-180 ~]# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
83b7e13eeb25   huaweiyun/awm2023                  /bin/bash --login -s    9 minutes ago Up 2 minutes  8001/tcp                                awm2023
1adf857719b5   awm2023/nginx_certbot              /docker-entrypoint...    3 days ago    Up 3 days    0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp  awm2023_nginx_certbot
564a22fa704c   dpape/pgadmin4                      /entrypoint.sh          4 days ago    Up 3 days    443/tcp, 0.0.0.0:20080->80/tcp, :::20080->80/tcp  awm2023_pgadmin4
1880901cfb76   kartoz/postgis                      /bin/sh -c /scripts...  4 days ago    Up 3 days    0.0.0.0:25432->5432/tcp, :::25432->5432/tcp  awm2023_postgis
[root@ip-172-31-33-180 ~]#
```

12. Visit <https://www.awm2023.site> and <https://www.awm2023.site/pgadmin4>



<https://www.awm2023.site/pgadmin4>



Connected to the postgres

The screenshot shows the pgAdmin 4 web interface in a browser. The browser's address bar displays the URL `awm2023.site/pgadmin4/browser/`. The pgAdmin interface includes a left-hand sidebar with a tree view of database objects, a central query editor, and a bottom section for query results and messages.

Database Object Tree (Left Sidebar):

- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (13)
 - auth_group
 - auth_group_permission
 - auth_permission
 - auth_user
 - auth_user_groups
 - auth_user_user_perm
 - awm2023_profile**
 - awm2023_worldbord
 - django_admin_log
 - django_content_type
 - django_migrations
 - django_session
 - spatial_ref_sys
- Trigger Functions
- Types
- Views
- topology
- Subscriptions

Query Editor (Center):

The query editor shows the following SQL query:

```
SELECT * FROM public.awm2023_profile
ORDER BY id ASC
```

Data Output (Bottom):

The query results are displayed in a table with 5 rows. The columns are: `id` (integer), `created` (timestamp with time zone), `modified` (timestamp with time zone), `last_location` (geometry), and `user_id` (integer).

id	created	modified	last_location	user_id
1	2022-10-18 13:52:31.651098+00	2022-11-08 15:04:45.056268+00	0101000020E6100000722E6A9CF22519C0BC3901F2CAAE4A40	1
2	2022-11-02 23:17:28.762489+00	2022-11-08 13:32:13.662126+00	0101000020E6100000A527767E9B0F19C0B901B650E8A8A4A0	2
3	2022-11-04 22:04:56.849931+00	2022-11-08 14:52:52.616065+00	0101000020E61000007591F5F48B0C19C0B381C448DDA4A4A0	3
4	2022-11-05 23:39:33.874758+00	2022-11-07 11:53:02.988173+00	0101000020E6100000C5C0F04AC2019C0784E69B3A0A04A...	4
5	2022-11-08 15:01:28.188319+00	2022-11-08 15:04:21.52871+00	0101000020E6100000722E6A9CF22519C0BC3901F2CAAE4A40	5

Total rows: 5 of 5 Query complete 00:00:00.091 Ln 1, Col 1