Deploying a Streamlit app on AWS EC2 with Docker, Nginx, and HTTPS

This guide explains how to deploy a Streamlit application (a word search application in our case) on an Amazon Web Services (AWS) Elastic Compute Cloud (EC2) instance by containerizing it with Docker and using Nginx as a reverse proxy to serve the app over HTTPS using self-signed certificates. Streamlit is a Python library for rapidly creating web applications, especially suited for data science and machine learning projects.

If you do not know how to create an AWS EC2 instance, please see this guide.

1. Launch an EC2 Instance: Go to the AWS Management Console, select EC2, and launch a new instance. Choose an Amazon Machine Image (AMI) that supports Docker, such as an Amazon Linux 2 AMI.

2. Configure Security Group: Ensure your security group allows inbound traffic on port 443 to enable HTTPS access.

3. Connect to Your Instance: Use SSH to connect to your instance after it is up and running.

4. Install Docker and Docker Compose. There are many guides online on how to do this, such as this one and this one.

5. Transfer files to your AWS EC2 instance: In our example, transfer docker-compose.yml, Dockerfile.app, Dockerfile.nginx, nginx.conf, and word_search_st.py to your home directory in your EC2 instance.

6. Create Certs: Enter the following commands to create your certificates. In this example, we use OpenSSL.

```
[ec2-user@ip-172-31-40-155 ~]$ mkdir certs
```

```
[ec2-user@ip-172-31-40-155 ~]$ cd certs
```

`openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout selfsigned.key -out selfsigned.crt`

Return to your home directory by entering `cd ..`.

7. Build Docker images: I built the Docker images using the following commands and then tagged each image.

```
[ec2-user@ip-172-31-40-155 ~]$ sudo docker buildx build -f Dockerfile.app .
[+] Building 29.4s (9/9) FINISHED
```

```
=> => writing image sha256:9bdf05c2bccb7dbe8a07bb7dcf9d3ec5d8c8d922fd1bed664ee
[ec2-user@ip-172-31-40-155 ~]$ sudo docker buildx build -f Dockerfile.nginx .
```

```
[ec2-user@ip-172-31-40-155 ~]$ sudo docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED             SIZE
<none>       <none>    fe0d6a62ec2e   About an hour ago   47MB
<none>       <none>    9bdf05c2bccb   About an hour ago   1.47GB
```

```
[ec2-user@ip-172-31-40-155 ~]$ sudo docker image tag fe0d6a62ec2e nginx_image
```

```
[ec2-user@ip-172-31-40-155 ~]$ sudo docker image tag 9bdf05c2bccb app_image
```

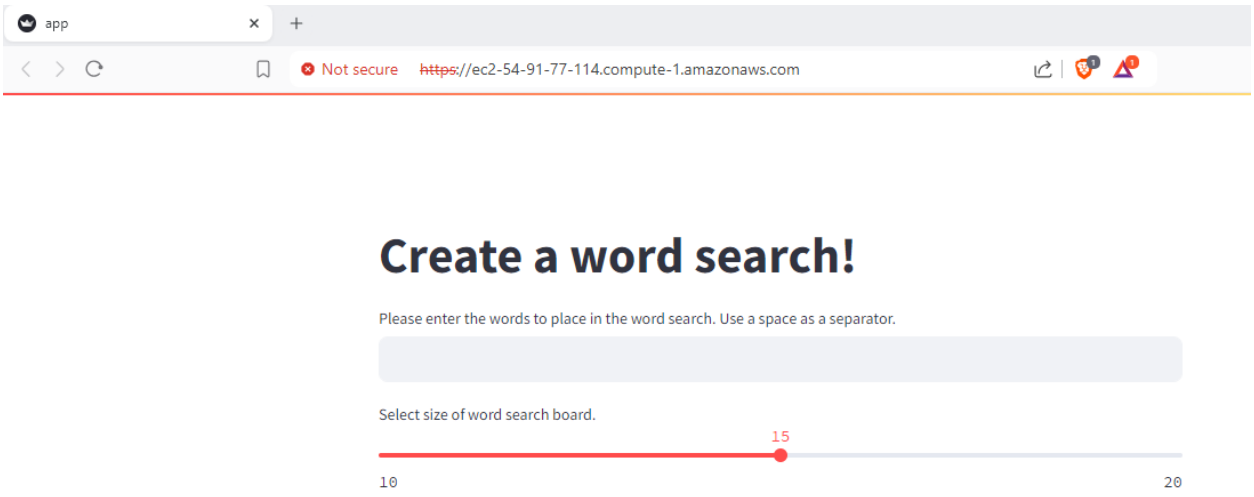
8. Run your Docker containers: Use Docker Compose to run your containers with the command `sudo docker-compose up –d`.

```
[ec2-user@ip-172-31-40-155 ~]$ sudo docker-compose up -d
[+] Running 2/2
 ⠿ Container ec2-user-app-1    Started
 ⠿ Container ec2-user-nginx-1  Started
```

Check that the containers are running:

```
[ec2-user@ip-172-31-40-155 ~]$ sudo docker container ls
CONTAINER ID   IMAGE         COMMAND                  CREATED         STATUS         PORTS                                                NAMES
ee73470a69c9   nginx_image   "/docker-entrypoint.…"   8 seconds ago   Up 7 seconds   80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp   ec2-user-nginx-1
ef315b5b19b5   app_image     "streamlit run app.p…"   8 seconds ago   Up 7 seconds   8501/tcp                                             ec2-user-app-1
```

9. Verify deployment: Navigate in your browser to `https://your_domain_or_IP`. You will receive a warning error because of the self-signed certificates.


app
Not secure
https://ec2-54-91-77-114.compute-1.amazonaws.com
Create a word search!
Please enter the words to place in the word search. Use a space as a separator.
Select size of word search board.
15
10
20