Please use private git repository, if anyhow public repository is used, then place SECURITY_KEY in .en

Here, in this doc, for domain name examples: example.com, ccstiet.com, escalade.ccstiet.com, etc.

This doc assumes that already server is set up, if setting up server for first time https://www.digitalocea

And run following commands to install necessary packages

```
sudo apt-get update
```

```
sudo apt-get install python3-pip python3-dev nginx docker python-certbot-nginx
```

Go to settings.py file

Add your domain name and localhost to ALLOWED_HOSTS

Set DEBUG=False

Change ‘NAME’ in ‘DATABASES’ to BASE_DIR / 'database/db.sqlite3'

If using Django4, add CSRF_TRUSTED_ORIGINS = ['https://domain-name']

Add whitenoise

Link for reference: http://whitenoise.evans.io/en/stable/#installation

In settings.py file

```
import os
```

```
STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')
```

Remember whitenoise should also be present in requirements.txt

In project’s urls.py add the content

```
from django.conf import settings
```

```
from django.conf.urls.static import static
```

Then modify urlpatterns to following

urlpatterns = [

#your urls…….

```
] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
```

Add the following to .gitignore

database

staticfiles

privkey.pem

cert.pem

First create the following files for the containerization:

Dockerfile

Add following contents:

```
# For more information, please refer to https://aka.ms/vscode-docker-python
```

```
FROM python:3.8-slim-buster
```

```
# Keeps Python from generating .pyc files in the container
```

```
ENV PYTHONDONTWRITEBYTECODE=1
```

```
# Turns off buffering for easier container logging
```

```
ENV PYTHONUNBUFFERED=1
```

Install pip requirements

```
COPY requirements.txt .
```

```
RUN pip install gunicorn[gevent]
```

```
RUN python -m pip install -r requirements.txt
```

```
WORKDIR /app
```

```
COPY . /app
```

```
# Creates a non-root user with an explicit UID and adds permission to access the /app folder
```

```
# For more info, please refer to https://aka.ms/vscode-docker-python-configure-containers
```

```
RUN adduser -u 5678 --disabled-password --gecos "" appuser && chown -R appuser /app
```

```
USER root
```

```
# During debugging, this entry point will be overridden. For more information, please refer to https://aka
```

```
CMD ["sh", "run.sh"]
```

docker-compose.yml

Add following contents:

```
version: '3.4'
```

services:

project-name:

image: ccs/project-name

build:

context: .

dockerfile: ./Dockerfile

ports:

- "6541:6541"

volumes:

- ./database/:/app/database/

```
restart: always
```

Note: Here please replace project-name by your project name. Also, I assume that the repository name

run.sh

Add following contents:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

```
python manage.py collectstatic
```

```
gunicorn  --worker-class gevent --certfile= cert.pem --keyfile=privkey.pem --bind 0.0.0.0:6541 django-pr
```

Here django-project-name-folder means the name of folder which contains settings.py and wsgi.py files

SSH to gcp, in user’s folder, clone the project repository.

Go to project’s folder

Run command

```
sudo nano /etc/nginx/sites-available/project-name
```

Replace project-name with your project name

Now nano editor will be opened up, add the following content

server {

```
listen 80;
```

server_name domain-name;

```
return 301 https://$server_name$request_uri;■
```

```
}
```

server {

```
listen 443 ssl;
```

server_name domain-name;

```
location / {
```

```
proxy_pass https://localhost:6541;
```

```
}
```

```
}
```

Replace domain-name with domain name

Run command

```
sudo ln -s /etc/nginx/sites-available/project-name /etc/nginx/sites-enabled
```

Please replace project-name with your project name

Run command:

```
sudo certbot --nginx -d domain-name
```

Replace domain-name with your domain name

If it asks for an option, enter 2, i.e. Redirect

Run command:

```
sudo cp /etc/letsencrypt/live/domain-name/cert.pem /etc/letsencrypt/live/domain-name/privkey.pem ./
```

Replace domain-name with domain name

Run command:

```
sudo docker-compose up –build
```

After the container is successfully built, then hit Ctrl + Z

Run command

```
sudo docker-compose up -d –build
```

Run commands

```
sudo nginx -t
```

```
sudo systemctl restart nginx
```

References:

https://medium.com/developer-student-clubs-tiet/hosting-multiple-containerized-websites-on-a-single-se

https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-gunico

https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-16

https://serverfault.com/questions/67316/in-nginx-how-can-i-rewrite-all-http-requests-to-https-while-main