Errors that occurred before lab 4

django.core.exceptions.ImproperlyConfigured: Could not find the GDAL library (tried "gdal304", "gdal303", "gdal302", "gdal301", "gdal300", "gdal204", "gdal203", "gdal202"). Is GDAL installed? If it is, try setting GDAL_LIBRARY_PATH in your settings.

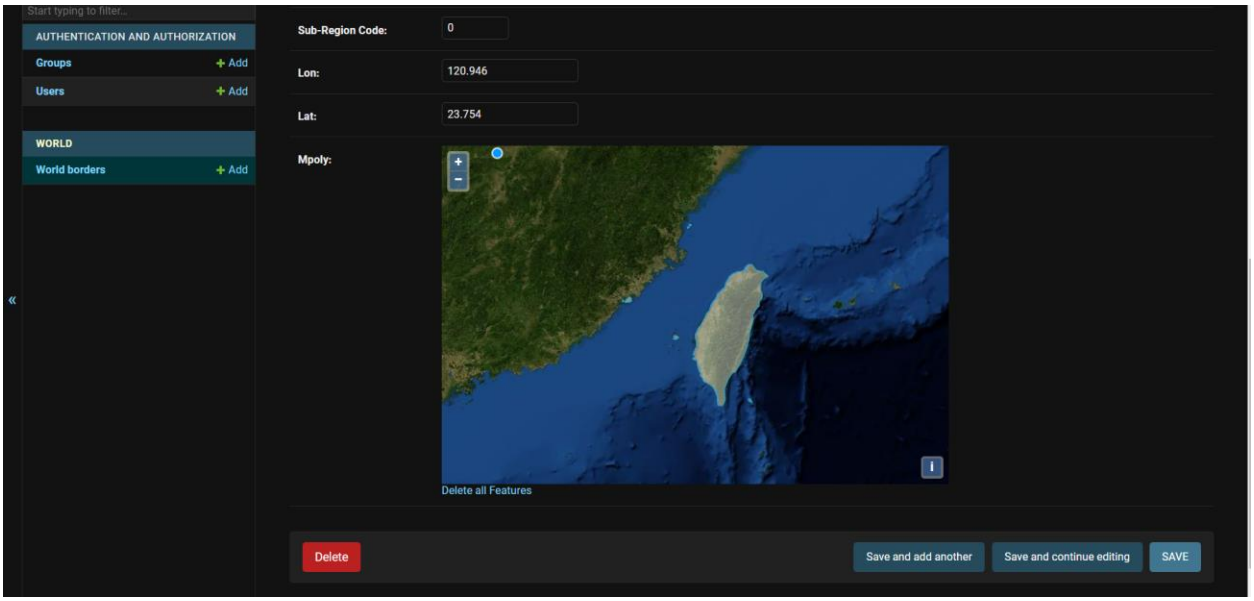
Tried to set a path using `os.environ.get('CONDA_PREFIX','')` and then `os.path.join()`

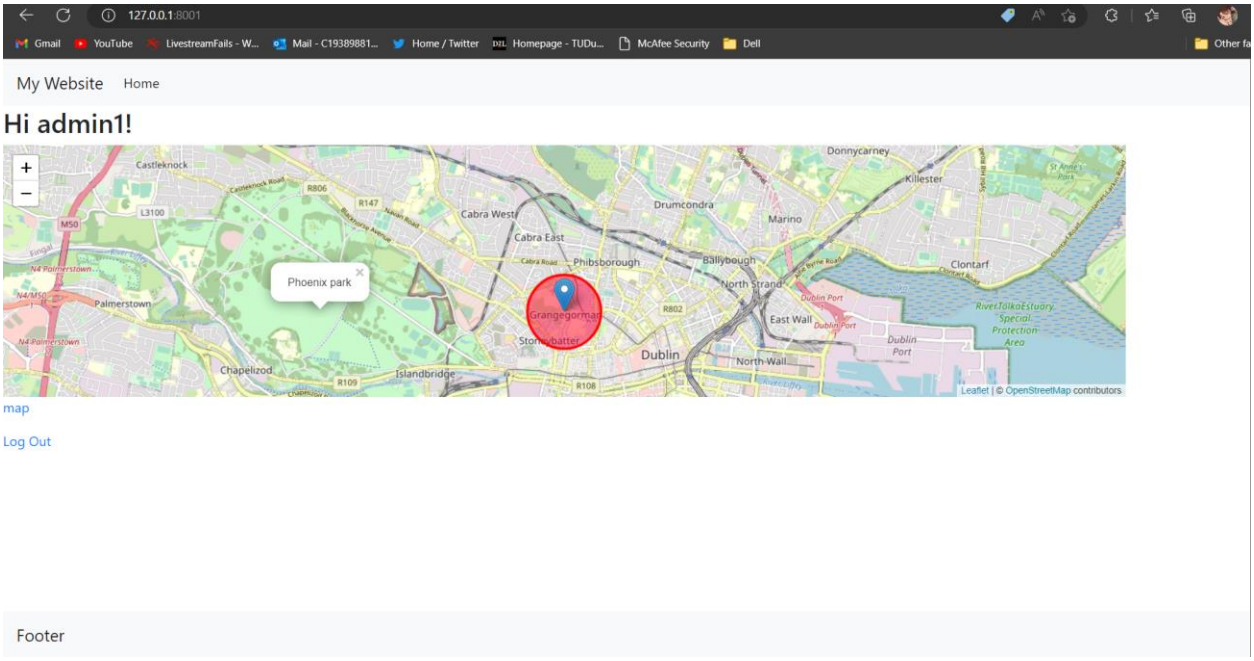
However that didn't fix it the solution ended up being going into a folder and manually entering 403 and 504 for versions. This error caused me a lot of hassle before I could do lab 4.

Another major error was a conflict between anaconda and miniconda not allowing base environments to be seen

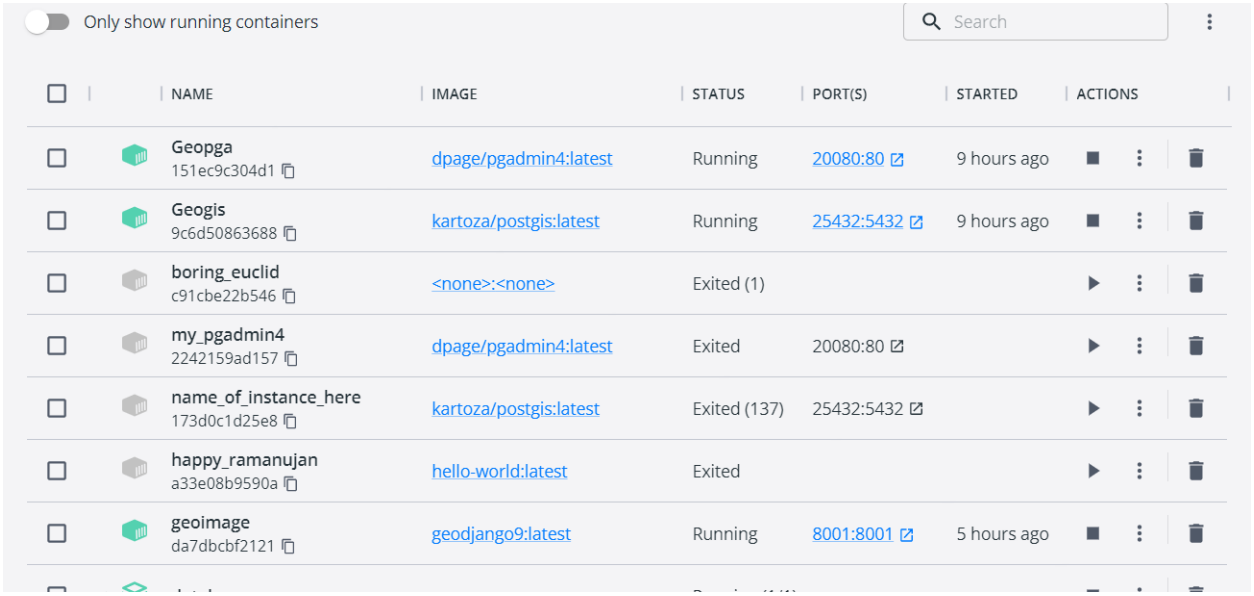
However all these were fixed and I could progress with the labs

LAB 4 DOCKER DELPLOYMENT





NOTE: PORT 8001



LAB 5

1.

```python
class Profile(models.Model):
    class Meta:
        verbose_name = ("user profile")
        verbose_name_plural =("user profiles")

    user = models.OneToOneField(get_user_model(), on_delete=models.CASCADE)
    created = models.DateTimeField(auto_now_add=True, editable=False)
    modified = models.DateTimeField(auto_now=True, editable=False)

    last_location = models.PointField(
        verbose_name=("last location"),
        editable=False,
        blank=True,
        null=True,
        default=None,
        help_text=(
            "Geographic coordinates (lon/lat) as Point. Can be serialised from WKT (well-known text) representation")
    )
```

2. Login

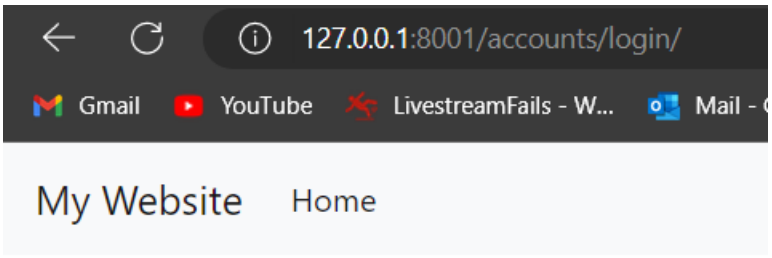
```
</head>
<body>
    <!-- templates/registration/login.html -->


    {% block title %}Login{% endblock %}

    {% block content %}

        <h2>Log In</h2>
        <form method="post">
            {% csrf_token %}
            {{ form.as_p }}
            <button type="submit">Log In</button>
        </form>
    {% endblock %}
</body>
</html>
```

Logout/login

```
        <p><a href="">map</a></p>
      <p><a href="{% url 'logout' %}">Log Out</a></p>
    {% else %}
      <p>You are not logged in</p>
      <a href="{% url 'login' %}">Log In</a>
    {% endif %}
{% endblock %}
```

You are not logged in

Log In



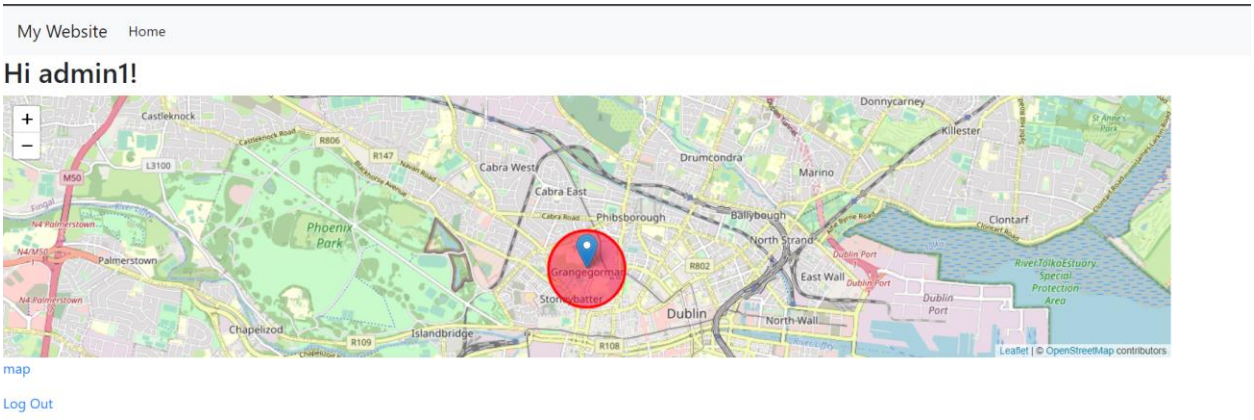
Log In

Username: admin1

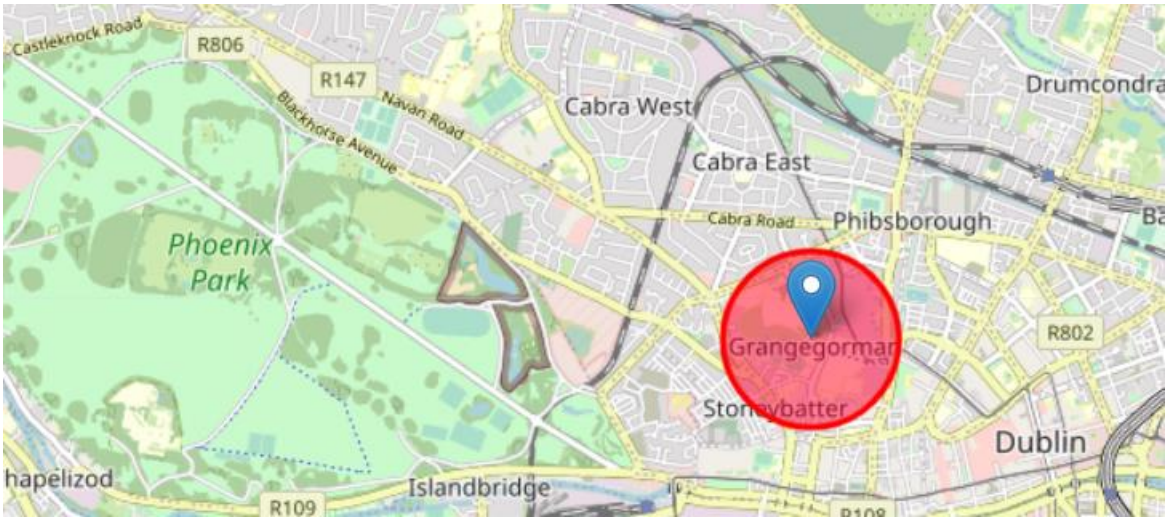
Password: ••••••

Log In

3. Map

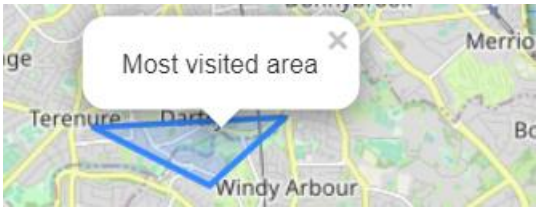


4.



5.

Polygon popup:



```
var polygon = L.polygon([
    [53.309, -6.28],
    [53.303, -6.26],
    [53.31, -6.247]
]).addTo(map);
polygon.bindPopup("Most visited area")}
```

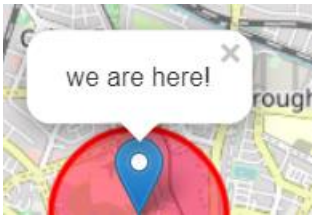
Standalone popup

```
var popup = L.popup()
    .setLatLng([53.355, -6.33])
    .setContent("Phoenix park")
    .openOn(map);
```

Circle Popup:

```
var circle = L.circle([53.355, -6.28], {
    color: 'red',
    fillColor: '#f03',
    fillOpacity: 0.5,
    radius: 500
    }).addTo(map);
circle.bindPopup("Area of college")
```

Marker popup




```
cript>
  function map_init_basic (map, options) {
      var marker = L.marker([53.355, -6.28]).addTo(map);
      marker.bindPopup("we are here!").openPopup();
```

6. base.html

```
<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <a class="navbar-brand" href="#">My Website</a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded=
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
        <div class="navbar-nav">
          <a class="nav-item nav-link active" href="#">Home <span class="sr-only">(current)</span></a>
        </div>
      </div>
    </nav>
  <main>
```

```
{% extends 'base.html' %}
```

7.database

Java script

```
function update_db(pos) {
        let locString = pos.coords.longitude + ", " + pos.coords.latitude;

        $.ajax({
            type: "POST",
            headers: {
                'X-CSRFToken': getCookie('csrftoken')
            },
            url: HOST + "/updatedb/",
            data: {
                point: locString
            }
        }).done(function (data, status, xhr) {
            console.log(data["message"])
            let originalMsg = $(".toast-body").html();
            $(".toast-body").html(originalMsg + "<br/>Updated database<br/>" + data["message"]);

        }).fail(function (xhr, status, error) {
            console.log(error);
            let originalMsg = $(".toast-body").html();
            $(".toast-body").html(originalMsg + "<br/>" + error);
        }).always(function () {
            console.log("find_loc_ed finished");
            $(".toast").toast('show');
        });
    }
```

Views.py

```python
from django.contrib.auth.decorators import login_required
from django.contrib.gis.geos import Point
from django.http import JsonResponse


@login_required
def update_database(request):
    my_location = request.POST.get("point", None)
    if not my_location:
        return JsonResponse({"message": "No location found."}, status=400)

    try:
        my_coords = [float(coord) for coord in my_location.split(", ")]
        my_profile = request.user.profile
        my_profile.last_location = Point(my_coords)
        my_profile.save()

        message = f"Updated {request.user.username} with {f'POINT({my_location})'}"

        return JsonResponse({"message": message}, status=200)
    except:
        return JsonResponse({"message": "No profile found."}, status=400)
```

Models.py

```python
class Profile(models.Model):
    class Meta:
        verbose_name = ("user profile")
        verbose_name_plural = ("user profiles")

    user = models.OneToOneField(get_user_model(), on_delete=models.CASCADE)
    created = models.DateTimeField(auto_now_add=True, editable=False)
    modified = models.DateTimeField(auto_now=True, editable=False)

    last_location = models.PointField(
        verbose_name=("last location"),
        editable=False,
        blank=True,
        null=True,
        default=None,
        help_text=(
            "Geographic coordinates (lon/lat) as Point. Can be serialised from WKT (well-known text) representation")
    )

    def __str__(self):
        return f"{self.user}"
```

```python
@receiver(post_save, sender=get_user_model())
def manage_user_profile(sender, instance, created, **kwargs):
    try:
        my_profile = instance.profile
        my_profile.save()
    except Profile.DoesNotExist:
        Profile.objects.create(user=instance)
```

Urls.py

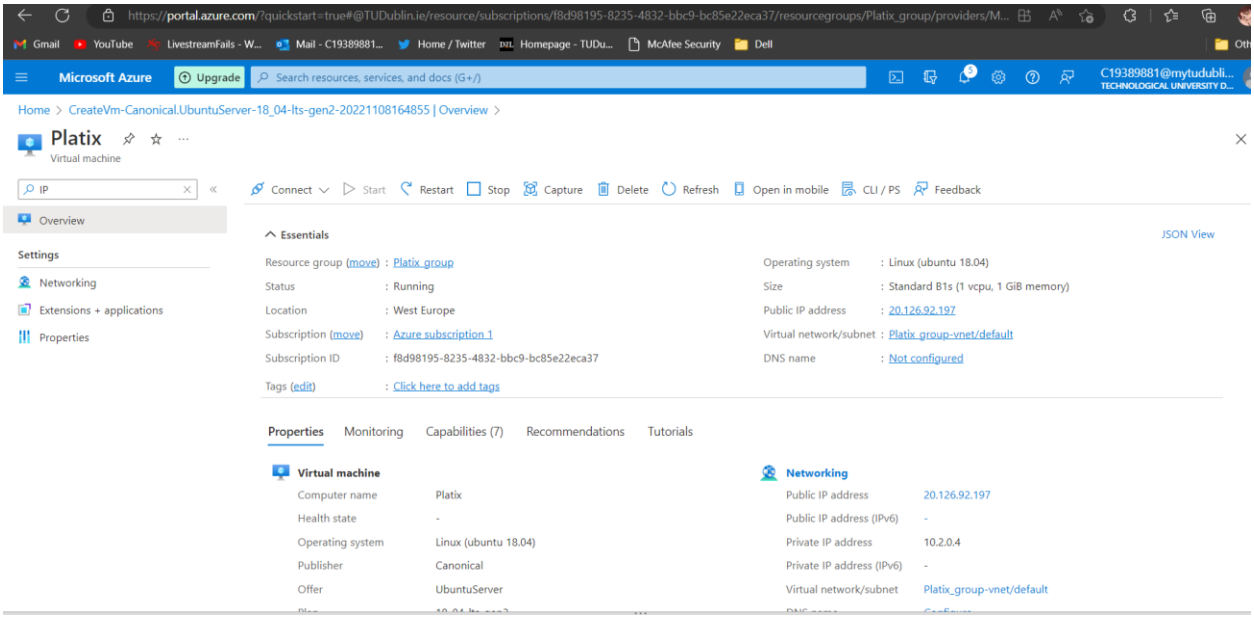
```python
# django_project/urls.py
from builtins import map
from django.contrib import admin
from django.urls import path, include
from django.views.generic.base import TemplateView # new
from world import views
from . import Views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('accounts/', include('django.contrib.auth.urls')),
    path("", TemplateView.as_view(template_name="index.html"), name="home"),
    path('updatedb/', views.update_database, name='update_db')
]
```

Cloud deployment:

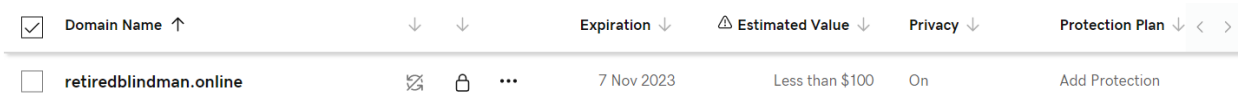
1. Microsoft Azure docker install

```
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: Platix
adduser: Please enter a username matching the regular expression configured
via the NAME_REGEX[_SYSTEM] configuration variable.  Use the `--force-badname'
option to relax this check or reconfigure NAME_REGEX.
Enter new UNIX username: platix
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Installation successful!
Windows Subsystem for Linux is now available in the Microsoft Store!
You can upgrade by running 'wsl.exe --update' or by visiting https://aka.ms/wslstorepage
Installing WSL from the Microsoft Store will give you the latest WSL updates, faster.
For more information please visit https://aka.ms/wslstoreinfo

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

platix@Platix:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/platix/.ssh/id_rsa): yes
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in yes.
Your public key has been saved in yes.pub.
The key fingerprint is:
SHA256:UqrhVrT5lVdN/2Z2HJ2j77rSKJSxiUxAlufGOyuwgZM platix@Platix
The key's randomart image is:
+---[RSA 2048]----+
|    .o.         .|
|    .o .       o+|
|      * .     .++|
|     . X . . ...+|
|  o . X S B ..  B|
| E + + B * .  .+.|
|  . B   =   o  . |
|   o . . . o ..  |
|      .   . .oo. |
+----[SHA256]-----+
platix@Platix:~$ c^C^C
platix@Platix:~$ ^C
platix@Platix:~$ cat yes.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCaEtDzow6cEMcFvya3KaZ2yc0WIEhWx+YvF2tstajBCx6t16QBZJ5js9OO/E+vgrwZdgzShf6EfBr2Jmam
EREdv95lmTzHy9MTHbBjkrCIK0+7pNQjG7I1aQOjSqlXcjYSf8DxLl1oN8gprV3sMzdWiNJpTt/0XLnm6V5jqswoDo+sUqv/5vjpJXWqyqbydMM1OsBHoHG6
wuWPo5Tg3DrMXO6AH65L/AN5jO733H6W+dO7A7iCr4n2otu7/1c1NqhnO3O8ytNsv19fowYmV+WlmMcvavD8zGMzURd5x/LLyeZN7jdNuPd96o5EwF13o2Qk
+tyPtnNNB+W0oYpeVWFJ platix@Platix
platix@Platix:~$ sudo apt update
[sudo] password for platix:
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2449 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [424 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [957 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [133 kB]
Get:10 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1238 kB]
Get:11 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [284 kB]
Get:12 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [19.0 kB]
Get:13 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [3836 B]
Get:14 http://archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [2790 kB]
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [513 kB]
Get:19 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [987 kB]
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [137 kB]
Get:21 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1853 kB]
Get:22 http://archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [401 kB]
Get:23 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [30.0 kB]
Get:24 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [7532 B]
Get:25 http://archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [53.2 kB]
```

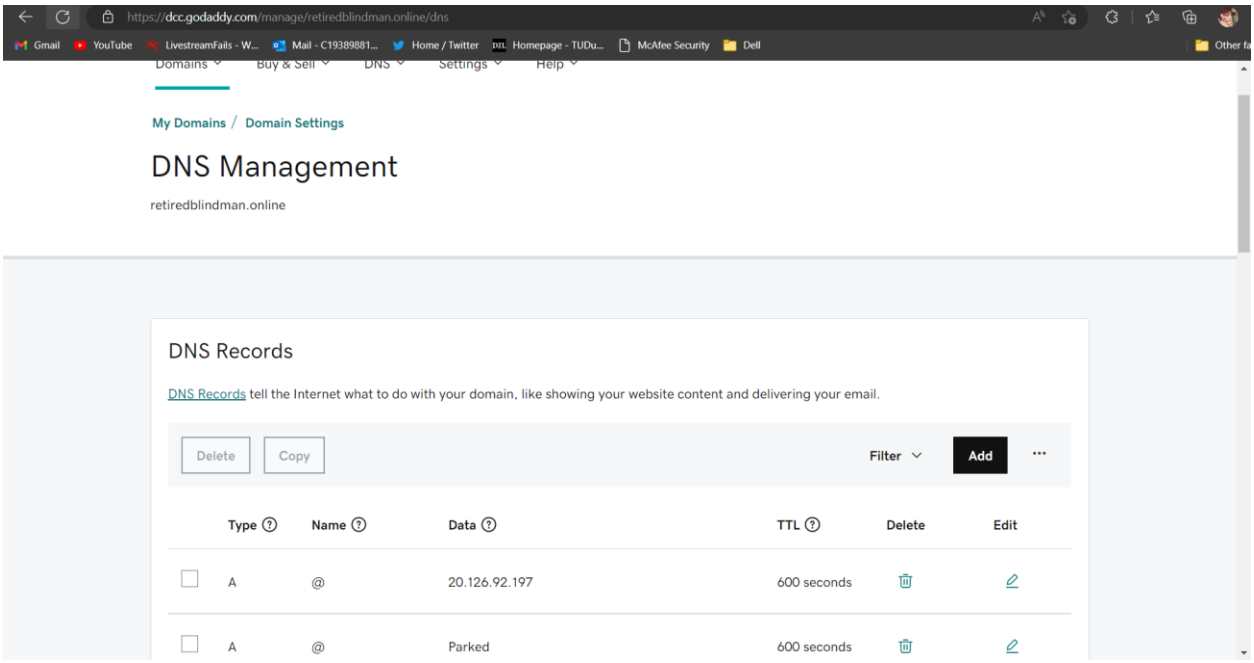
```
platix@Platix:~$ sudo apt-get update
Get:27 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [18.1 kB]
Get:28 http://archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [8668 B]
Fetched 26.4 MB in 13s (2036 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
116 packages can be upgraded. Run 'apt list --upgradable' to see them.
platix@Platix:~$ sudo apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
platix@Platix:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd docker.io pigz runc ubuntu-fan
0 upgraded, 6 newly installed, 0 to remove and 116 not upgraded.
Need to get 74.2 MB of archives.
After this operation, 360 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 bridge-utils amd64 1.5-15ubuntu1 [30.1 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 runc amd64 1.0.1-0ubuntu2~18.04.1 [4155 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 containerd amd64 1.5.5-0ubuntu3~18.04.2 [33.0 MB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 docker.io amd64 20.10.7-0ubuntu5~18.04.3 [36.9 MB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 ubuntu-fan all 0.12.10 [34.7 kB]
Fetched 74.2 MB in 27s (2790 kB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 29603 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.5-15ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.5-15ubuntu1) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.0.1-0ubuntu2~18.04.1_amd64.deb ...
Unpacking runc (1.0.1-0ubuntu2~18.04.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.5.5-0ubuntu3~18.04.2_amd64.deb ...
Unpacking containerd (1.5.5-0ubuntu3~18.04.2) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../4-docker.io_20.10.7-0ubuntu5~18.04.3_amd64.deb ...
Unpacking docker.io (20.10.7-0ubuntu5~18.04.3) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../5-ubuntu-fan_0.12.10_all.deb ...
Unpacking ubuntu-fan (0.12.10) ...
Setting up runc (1.0.1-0ubuntu2~18.04.1) ...
Setting up containerd (1.5.5-0ubuntu3~18.04.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up bridge-utils (1.5-15ubuntu1) ...
Setting up ubuntu-fan (0.12.10) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
invoke-rc.d: could not determine current runlevel
Setting up pigz (2.4-1) ...
Setting up docker.io (20.10.7-0ubuntu5~18.04.3) ...
Adding group `docker' (GID 115) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
invoke-rc.d: unknown initscript, /etc/init.d/docker not found.
invoke-rc.d: could not determine current runlevel
Processing triggers for systemd (237-3ubuntu10.53) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
platix@Platix:~$ docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu5~18.04.3
```



2. Create domain name



3. Modifying DNS entry



4 cerbot

Kept getting errors on normal cerbot installations tried : sudo apt install certbot python3-certbot-nginx

And it worked

```
platix@Platix:~$ sudo snap
The snap command lets you install, configure, refresh and remove snaps.
Snaps are packages that work across many different Linux distributions,
enabling secure delivery and operation of the latest apps and utilities.

Usage: snap <command> [<options>...]

Commonly used commands can be classified as follows:

        Basics: find, info, install, remove, list
       ...more: refresh, revert, switch, disable, enable, create-cohort
       History: changes, tasks, abort, watch
       Daemons: services, start, stop, restart, logs
   Permissions: connections, interface, connect, disconnect
 Configuration: get, set, unset, wait
    App Aliases: alias, aliases, unalias, prefer
       Account: login, logout, whoami
     Snapshots: saved, save, check-snapshot, restore, forget
        Device: model, reboot, recovery
     ... Other: warnings, okay, known, ack, version
   Development: download, pack, run, try

For more information about a command, run 'snap help <command>'.
For a short summary of all commands, run 'snap help --all'.
platix@Platix:~$ sudo snap install
error: the required argument `<snap> (at least 1 argument)` was not provided
platix@Platix:~$ sudo apt install certbot python3-certbot-nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
```
7

5.1 had problems gettings this error

```
platix@Platix:~$ sudo systemctl status docker
System has not been booted with systemd as init system (PID 1). Can't operate.
```

However I fixed it by adding user to docker group

```
platix@Platix:~$ sudo systemctl status docker
System has not been booted with systemd as init system (PID 1). Can't operate.
platix@Platix:~$ sudo usermod -aG docker platix
platix@Platix:~$ su - platix
Password:
platix@Platix:~$ id -nG
platix adm dialout cdrom floppy sudo audio dip video plugdev netdev docker
platix@Platix:~$ docker network create wmap_network
692b7e92096e0bdc4ac27b4ddd19f880532d56f5af55a4f6263aeda843fc075d
platix@Platix:~$
```

5.2

```
platix@Platix:~$ pwd
/home/platix
platix@Platix:~$ mkdir certs
platix@Platix:~$ ls certs/
platix@Platix:~$ cd certs/
platix@Platix:~/certs$ touch Dockerfile
platix@Platix:~/certs$ nano Dockerfile
platix@Platix:~/certs$ nano Dockerfile
platix@Platix:~/certs$
```

Made a dockerfile

Builing image

```
platix@Platix:~/certs$ docker build -t wmap-nginx-certbot .
Sending build context to Docker daemon  2.048kB
Step 1/3 : From nginx
latest: Pulling from library/nginx
e9995326b091: Downloading [====>                                              ]  2.948MB/31.42MB
71689475aec2: Downloading [====>                                              ]  2.105MB/25.41MB
f88a23025338: Download complete
0df440342e26: Download complete
eef26ceb3309: Download complete
8e3ed6a9e43a: Waiting
```

Build successfully

```
Processing triggers for dbus (1.12.24-0+deb11u1) ...
Removing intermediate container 2b8fb98a6199
 ---> ee8fd494db7d
Successfully built ee8fd494db7d
Successfully tagged wmap-nginx-certbot:latest
platix@Platix:~/certs$
```

Kept getting these errors

Error response from daemon: pull access denied for wmap_nginx_certbot, repository does not exist or may require 'docker login': denied: requested access to the resource is denied

Unable to find image 'wmap_nginx_certbot:latest' locally

Changed it to this

docker create --name wmap_nginx_certbot --network wmap_network --network-alias wmap-nginx-certbot -p 80:80 -p 443:443 -t -v wmap_web_data:/usr/share/nginx/html -v $HOME/wmap_nginx_certbot/conf:/etc/nginx/conf.d -v /etc/letsencrypt:/etc/letsencrypt -v /var/www/certbot -v html_data:/usr/share/nginx/html/static wmap-nginx-certbot

```
platix@Platix:~$ docker images
REPOSITORY          TAG       IMAGE ID       CREATED         SIZE
wmap-nginx-certbot  latest    ee8fd494db7d   3 minutes ago   250MB
nginx               latest    76c69feac34e   2 weeks ago     142MB
platix@Platix:~$ docker create --name wmap_nginx_certbot --network wmap_network --network-alias wmap-nginx-certbot -p 80:80 -p 443:443
-t -v wmap_web_data:/usr/share/nginx/html -v $HOME/wmap_nginx_certbot/conf:/etc/nginx/conf.d -v /etc/letsencrypt:/etc/letsencrypt -v /
ar/www/certbot -v html_data:/usr/share/nginx/html/static  wmap_nginx_certbot
Unable to find image 'wmap_nginx_certbot:latest' locally
Error response from daemon: pull access denied for wmap_nginx_certbot, repository does not exist or may require 'docker login': denied
 requested access to the resource is denied
platix@Platix:~$ docker create --name wmap_nginx_certbot --network wmap_network --network-alias wmap-nginx-certbot -p 80:80 -p 443:443
-t -v wmap_web_data:/usr/share/nginx/html -v $HOME/wmap_nginx_certbot/conf:/etc/nginx/conf.d -v /etc/letsencrypt:/etc/letsencrypt -v /
ar/www/certbot -v html_data:/usr/share/nginx/html/static  wmap-nginx-certbot
0d20f058c7ff6cae3560b8a6eacde1c5f64f0e104aad4f3c8e600eff84feb679
platix@Platix:~$
```

It Worked!

5.3

```
platix@Platix:~$ docker create --name wmap_pgadmin4 --network wmap_network --network-alias wmap-pgadmin4 -t -v wmap_pgadmin_data:/var/
ib/pgadmin -e 'PGADMIN_DEFAULT_EMAIL=mark.foley@tudublin.ie' -e 'PGADMIN_DEFAULT_PASSWORD=mypassword' dpage/pgadmin4
Unable to find image 'dpage/pgadmin4:latest' locally
latest: Pulling from dpage/pgadmin4
213ec9aee27d: Pull complete
f5bd6e132847: Pull complete
78611d880b87: Pull complete
b8da2d85edeb: Pull complete
58388dc05662: Pull complete
ed06f0771df9: Pull complete
5f212cce9bde: Pull complete
f236ae664799: Pull complete
a2befbeabb07: Pull complete
f3ffb41e8353: Pull complete
355b3ee7a209: Pull complete
44432767775c: Pull complete
561510a9f033: Pull complete
d3639fbc6f6f: Pull complete
Digest: sha256:a220fea89298275e088be066b0bb6726c363ebb583343c89cd8a8d547b9728cf
Status: Downloaded newer image for dpage/pgadmin4:latest
33e76b984275ae0faec34e69bc0215e670b4f2652e58f3e35d22e32a07050b58
```

5.4

```
platix@Platix:~$ docker create --name wmap_postgis --network wmap_network --network-alias wmap-postgis -t -v wmap_postgis_data:/var/lib
/postgresql -e 'POSTGRES_USER=docker' -e 'POSTGRES_PASS=docker' kartoza/postgis
Unable to find image 'kartoza/postgis:latest' locally
latest: Pulling from kartoza/postgis
31d64ed241e5: Pull complete
8e7518324b44: Pull complete
6f3466703624: Pull complete
4c223880f0cd: Pull complete
7e64650f788f: Pull complete
5ae754b0fec9: Pull complete
71125d38f568: Pull complete
c34f17e93f82: Pull complete
7c98c0f0c136: Pull complete
15c67f62d676: Pull complete
3cfb53097a17: Pull complete
083b9d9d6f79: Pull complete
80c7487b4a68: Pull complete
bb886aeca663: Pull complete
f367039cbc9f: Pull complete
80d7d77f75ba: Pull complete
5b7141fa6574: Pull complete
6eea56700fc5: Pull complete
d29d60289d66: Pull complete
3a88eaa954c8: Pull complete
b9a6a59df913: Pull complete
be29a684e2b7: Pull complete
372342e00d12: Pull complete
Digest: sha256:1360f5d3dcb2de05e8f6ced767e26a5f29344876a9ef119f1a3a69c9583eceda
Status: Downloaded newer image for kartoza/postgis:latest
826e18488776b02b71644eb420138bd2f1a79c629932550001fa72e97b0f8025
platix@Platix:~$
```

6.settings.py secret_key

```python
import socket
import dj_database_url
from decouple import config
import os

SECRET_KEY = config('SECRET_KEY', default=None)
DEPLOY_SECURE = config('DEPLOY_SECURE', default=False, cast=bool)

if DEPLOY_SECURE:
    DEBUG = False
    CSRF_COOKIE_SECURE = True
    SESSION_COOKIE_SECURE = True
    ALLOWED_HOSTS = config('ALLOWED_HOSTS', cast=lambda v: [s.strip() for s in v.split(',')])
    CSRF_TRUSTED_ORIGINS = config('CSRF_TRUSTED_ORIGINS', cast=lambda v: [s.strip() for s in v.split(',')])
else:
    DEBUG = True
    CSRF_COOKIE_SECURE = False
    SESSION_COOKIE_SECURE = False
    ALLOWED_HOSTS = []

STATICFILES_STORAGE = 'whitenoise.storage.CompressedManifestStaticFilesStorage'
STATIC_ROOT = os.path.join(BASE_DIR, "static")
STATIC_URL = "/static/"

if socket.gethostname() =="platix":
```

```
if socket.gethostname() =="platix":
    DATABASES["default"]["HOST"] = "localhost"
    DATABASES["default"]["PORT"] = docker_config.POSTGIS_PORT
else:
    DATABASES["default"]["HOST"] = f"{docker_config.Geodjango9}-postgis"
    DATABASES["default"]["PORT"] = 5432

# Set DEPLOY_SECURE to True only for LIVE deployment
if docker_config.DEPLOY_SECURE:
    DEBUG = False
    TEMPLATES[0]["OPTIONS"]["debug"] = False
    # ALLOWED_HOSTS = ['.retiredblindman.online', 'localhost',]
    CSRF_COOKIE_SECURE = True
    SESSION_COOKIE_SECURE = True
else:
    DEBUG = True
    TEMPLATES[0]["OPTIONS"]["debug"] = True
    ALLOWED_HOSTS = ['*', ]
    CSRF_COOKIE_SECURE = False
    SESSION_COOKIE_SECURE = False
```

7.Django docker image and container

Docker Hub

```
docker push platix/geodjango9:tagname
```

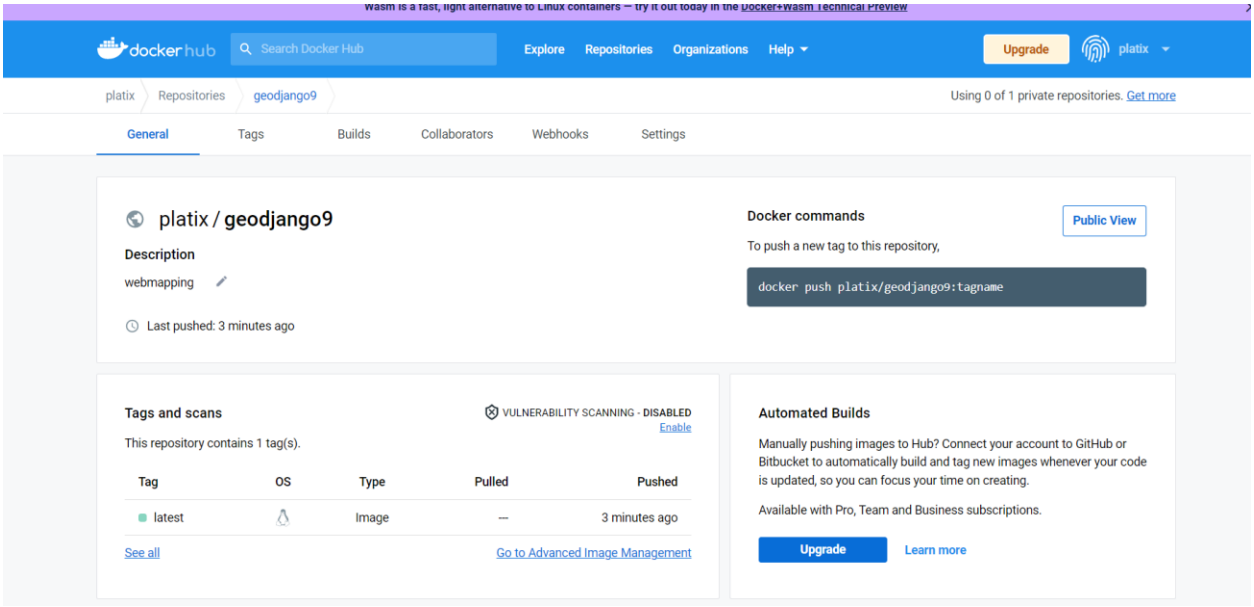
```
C:\Users\macgo>docker push platix/geodjango9:tagname
The push refers to repository [docker.io/platix/geodjango9]
An image does not exist locally with the tag: platix/geodjango9

C:\Users\macgo>docker images
REPOSITORY        TAG       IMAGE ID       CREATED         SIZE
geodjango9        latest    206fa74f8a73   6 hours ago     3.67GB
<none>            <none>    f233beac2598   6 hours ago     3.67GB
<none>            <none>    b037317a1da3   6 hours ago     3.67GB
<none>            <none>    3f3b03d3137a   7 hours ago     3.67GB
<none>            <none>    ae75ab540ddc   7 hours ago     3.67GB
<none>            <none>    f19c652cc82a   7 hours ago     3.67GB
<none>            <none>    eaa43829af85   7 hours ago     3.67GB
<none>            <none>    5f6c3377a1e9   7 hours ago     3.67GB
<none>            <none>    f2652a2ad962   8 hours ago     4.29GB
<none>            <none>    372e872235eb   2 days ago      3.65GB
postgres          latest    027eba2e8939   2 weeks ago     377MB
dpage/pgadmin4    latest    173beb2d4f13   3 weeks ago     370MB
hello-world       latest    feb5d9fea6a5   13 months ago   13.3kB
kartoza/postgis   latest    acb761fa7225   16 months ago   1.65GB

C:\Users\macgo>docker tag geodjango9 platix/geodjango

C:\Users\macgo>docker push platix/geodjango9
Using default tag: latest
The push refers to repository [docker.io/platix/geodjango9]
An image does not exist locally with the tag: platix/geodjango9

C:\Users\macgo>docker push platix/geodjango
Using default tag: latest
The push refers to repository [docker.io/platix/geodjango]
e63cefc9db0f: Preparing
ec6278262fdf: Preparing
6978db1913c3: Preparing
148e8b8707ee: Preparing
33b1096bd017: Preparing
133bd5842405: Waiting
56be2f0c1fe4: Waiting
d0bd627a6b09: Waiting
2bd01b0018c8: Waiting
```

Kept getting errors had to login

```
5f70bf18a086: Waiting
288fbcfdfa99: Waiting
041cea4279db: Waiting
8717fc7b73d9: Waiting
ab2731ec3f53: Waiting
6fa1f4185aa2: Waiting
ad6562704f37: Waiting
denied: requested access to the resource is denied

C:\Users\macgo>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 htt
ub.docker.com to create one.
Username: Platix
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password

C:\Users\macgo>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 htt
ub.docker.com to create one.
Username: platix
Password:
Login Succeeded
```

Success

```
C:\Users\macgo>docker push platix/geodjango9
Using default tag: latest
The push refers to repository [docker.io/platix/geodjango9]
e63cefc9db0f: Pushed
ec6278262fdf: Pushed
6978db1913c3: Pushed
148e8b8707ee: Pushed
33b1096bd017: Pushed
133bd5842405: Pushed
56be2f0c1fe4: Pushed
d0bd627a6b09: Pushed
2bd01b0018c8: Pushed
5f70bf18a086: Pushed
288fbcfdfa99: Pushed
041cea4279db: Pushed
8717fc7b73d9: Pushed
ab2731ec3f53: Mounted from continuumio/miniconda3
6fa1f4185aa2: Mounted from continuumio/miniconda3
ad6562704f37: Mounted from continuumio/miniconda3
latest: digest: sha256:4c8ea116656b6ceb239bf598428dffe68dd8d6f68c4c8be62925bb4918850de4 size: 3682
```



Pull into ubuntu

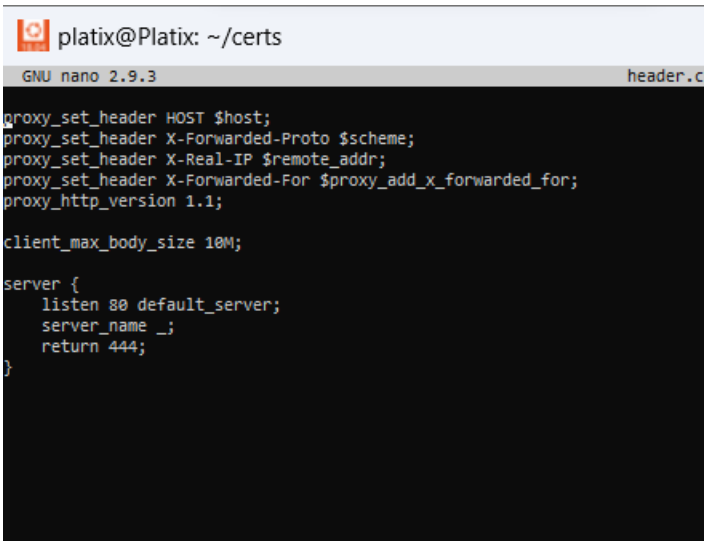
```
platix@Platix:~/certs$ docker pull platix/geodjango9
Using default tag: latest
latest: Pulling from platix/geodjango9
42c077c10790: Pull complete
1a23c9d790a3: Pull complete
22a6fc63b9b5: Pull complete
7a3af7c1e7f9: Pull complete
c42c7956ca14: Pull complete
969f3ff07df1: Pull complete
4f4fb700ef54: Pull complete
2138f132d946: Pull complete
63bd5449c4c5: Pull complete
ec190d10dd47: Pull complete
4033ade0c911: Pull complete
d27c9d6b0f82: Pull complete
baa604b6ddb7: Pull complete
9e25c2d5263c: Pull complete
b199482540c8: Pull complete
22404adbf53a: Pull complete
Digest: sha256:4c8ea116656b6ceb239bf598428dffe68dd8d6f68c4c8be62925bb4918850de4
Status: Downloaded newer image for platix/geodjango9:latest
docker.io/platix/geodjango9:latest
```

Create container with image

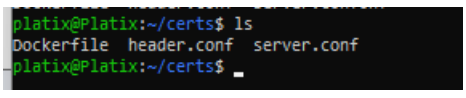
```
platix@Platix:~$ docker create --name geodjango9 --network wmap_network --network-alias wmap_network -t -v html_data:/usr/src/app/sta
c platix/geodjango9
```

8.

Headers.conf


platix@Platix: ~/certs
GNU nano 2.9.3
header.c
proxy_set_header HOST $host;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_http_version 1.1;
client_max_body_size 10M;
server {
listen 80 default_server;
server_name _;
return 444;
}

Serv.conf



9.

POST GIS: Successful

```
platix@Platix:~$ docker start wmap_postgis
wmap_postgis
```

PgAdmin4: Successful

```
platix@Platix:~$ docker start wmap_pgadmin4
wmap_pgadmin4
```

Django: Successful

```
platix@Platix:~$ docker start geodjango9
geodjango9
```

Nginx: Successful

```
platix@Platix:~$ docker start wmap_nginx_certbot
wmap_nginx_certbot
```