Hosting React on AWS Amplify:

[Youtube Video](https://www.youtube.com/watch?v=4hb4qLhJGoQ&t=622s)

Creating EC2 instance:

**create a key pair and save it**

Key Pair Type : RSA

File format : .pem

**Network settings**

Allow SSH traffic from Change Anywhere(0.0.0.0/0) to My IP

Allow HTTPS traffic from the internet

Allow HTTP traffic from the internet

[Digital Ocean Documentation](https://www.digitalocean.com/community/tutorials/how-to-set-up-an-asgi-django-app-with-postgres-nginx-and-uvicorn-on-ubuntu-20-04)

In the documentation in Step 4 — Creating and Configuring a New Django Project

### Don’t create a new project ([Creating the Django Project](https://www.digitalocean.com/community/tutorials/how-to-set-up-an-asgi-django-app-with-postgres-nginx-and-uvicorn-on-ubuntu-20-04#creating-the-django-project)

) clone your existing project from git,

### Change database if you want. And continue from [Completing Initial Project Setup](https://www.digitalocean.com/community/tutorials/how-to-set-up-an-asgi-django-app-with-postgres-nginx-and-uvicorn-on-ubuntu-20-04#completing-initial-project-setup)

To add DNS:

1. Purchase a domain.
2. Go to the site where you purchased the domain, go to page DNS / Name Servers (Edit or Add) and Add Record.  
   type: A

Name: @

Points to: EC2 Public IPv4 address (e.g.: 17.160.253.180)

TTL (Time to Live): Add time (e.g.: 14400 seconds)

{Time to live (TTL) refers to the amount of time or “hops” that a packet is set to exist inside a network before being discarded by a router. TTL is also used in other contexts including CDN caching and DNS caching.}

1. sudo nano /etc/nginx/sites-available/myproject

server {

listen 80;

server\_name yourdomainname.com www.yourdomainname.com;

location = /favicon.ico { access\_log off; log\_not\_found off; }

location /static/ {

root /home/sammy/myprojectdir;

}

location / {

include proxy\_params;

proxy\_pass http://unix:/run/gunicorn.sock;

}

}

1. To make DNS secure (http to https).

sudo add-apt-repository ppa:certbot/certbot

sudo apt-get update

sudo apt-get install python3-certbot-nginx

sudo certbot --nginx -d yourdomain.com -d [www.yourdomain.com](http://www.yourdomain.com)

sudo certbot --nginx -d insyncbackend.arunkrishna.online -d www.insyncbackend.arunkrishna.online

1. To Connect With WS(Web Socket) Server.

server {

server\_name yourdomain.com www.yourdomain.com;

location = /favicon.ico { access\_log off; log\_not\_found off; }

location /static/ {

root /home/ubuntu/insyncbackend1;

}

location / {

include proxy\_params;

proxy\_pass http://unix:/run/gunicorn.sock;

}

location /ws/ {

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

proxy\_redirect off;

proxy\_pass http://127.0.0.1:8001;

}

listen 443 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/yourdomain.com/fullchain.pem; # managed by Certbot

ssl\_certificate\_key /etc/letsencrypt/live/yourdomain.com/privkey.pem; # managed by Certbot

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

server {

if ($host = www.yourdomain.com) {

return 301 https://$host$request\_uri;

} # managed by Certbot

if ($host = yourdomain.com) {

return 301 https://$host$request\_uri;

} # managed by Certbot

listen 80;

server\_name yourdomain.com www.yourdomain.com;

return 404; # managed by Certbot

}

1. Open settings.py and add

CHANNEL\_LAYERS = {

'default': {

'BACKEND': 'channels\_redis.core.RedisChannelLayer',

'CONFIG': {

"hosts": [('127.0.0.1', 6379)],

},

},

}

1. Open EC2 inbound rule add a new rule  
   Type: Custom TCP

Port range: 6379

Source: Custom 0.0.0.0/0

1. sudo systemctl restart nginx
2. Run Daphne in the Foreground:

daphne backend.asgi:application --port 8001

Observe the output for any error messages or issues.

pip install channels\_redis

Check Redis Server Status

sudo systemctl status redis

sudo systemctl start redis

Test Redis Connection

redis-cli ping

Install Redis if it does not exist

sudo apt-get update

sudo apt-get install redis-server

If Redis is not running, you can start it with

redis-server

Check Redis Process:

ps aux | grep redis-server

run daphne again

daphne backend.asgi:application --port 8001

CREATING SERVICE FOR DAPHNE

sudo nano /etc/systemd/system/daphne.service

[Unit]

Description=Daphne ASGI Server

After=network.target

[Service]

User=ubuntu

Group=ubuntu

WorkingDirectory=/home/ubuntu/insyncbackend1

ExecStart=/home/ubuntu/venv/bin/daphne insyncbackend.asgi:application --port 8001

Restart=always

RestartSec=5

[Install]

WantedBy=multi-user.target

sudo systemctl daemon-reload

sudo systemctl enable daphne.service

sudo systemctl start daphne.service