0. Pre-requisite:

0.1. SSH to EC2 from your device (Windows/Mac)

0.2. Enable two-factor authentication with RobinHood & Google Authenticator mobile App:

- Check this Tutorial

1. Working with AWS EC2

1.1. Set up EC2 instance

```
sudo apt-get update
sudo apt-get install build-essential python3-venv unzip sqlite3 -y
sudo apt-get install mysql-server mysql-client -y
sudo apt-get install python3-mysqldb libmysqlclient-dev python3-dev -y
```

1.2. Set up Python virtual environment

```
python3 -m venv f-venv
source f-venv/bin/activate
pip install wheel
```

- Note: `pip install -r requirement.txt` is moved to the next step

1.3. Get Django project

```
wget https://github.com/qz-fordham/algo-trading-microservice/archive/main.zip
unzip main.zip
cd algo-trading-microservice-main/algo_trading/
pip install -r requirement.txt
```

2. Working with AWS RDS (MySQL)

2.1. Remote access RDS to create Database on MySQL server

```
mysql -u admin -h MYSQL-HOST-NAME-OF-YOURS -p
CREATE DATABASE algo_trading_db;
```

- Note: Make sure you use correct RDS instance hostname to replace `MYSQL-HOST-NAME-OF-YOURS`

3. Configure Django Project

3.1. Manually login once with your robinhood account (RobinHood will cache your credential)

```
python
import robin_stocks as rbh
rbh.login(username, password, mfa_code=xxxxxx)
```

3.2. Update database setting in settings.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': 'algo_trading_db',
        'USER': 'admin',
        'PASSWORD': 'password123',
        'HOST': '',
        'PORT': '3306',
```

```
    }
}
```

- Note: Make sure you add EC2 public DNS to the `HOST` field

3.3. Update allowed_host in settings.py

- Add ec2 hostname to `ALLOWED_HOSTS` (use the ame hostname as previous step)

3.4. Start broker

```
python manage.py migrate
python manage.py makemigrations
python manage.py runserver 0.0.0.0:8000
```

4. Update EC2 instance security setting so that site can be visited from anywhere

4.1. Update EC2 instance security group inbound, e.g.

- Add Custom TCP - 8000 (source 0.0.0.0/0). If that's the port Django is running on.

5. Now you can access the website from your browser