Django based hosting

Support Link 1: https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create-deploy-python-django.html#python-django-setup-venv

Support Link 2: https://realpython.com/django-setup/

Commands in Terminal (Linux Based)

```
sudo yum update

sudo yum install python3

pip3 list

pip3 install virtualenv

pip3 install awsebcli

virtualenv virt

source ~/virt/bin/activate
```

// for deactivation – go to the folder where virtual file is created and then give “deactivate”

// to remove a directory “rm -r <directory_name>”

```
pip3 install django==2.1
```

//to verify

```
pip freeze

django-admin startproject proj1
```

//the above command will create the directory as below,

```
~/proj1
  |-- proj1
  |   |-- __init__.py
  |   |-- settings.py
  |   |-- urls.py
  |   `-- wsgi.py
  `-- manage.py
```

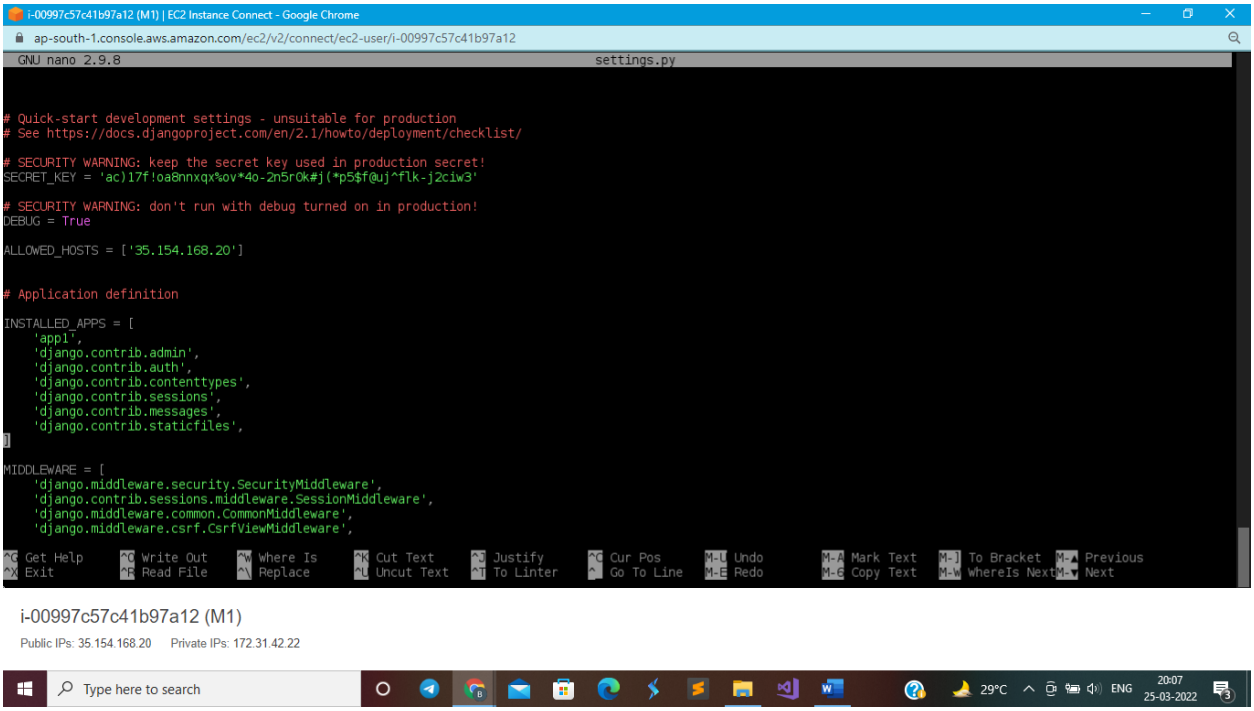
```
cd proj1

python manage.py startapp app1
```

it will create a following directory structure

```
proj1/
|
├── app1/
|   |
|   ├── migrations/
|   |   └── __init__.py
|   |
|   ├── __init__.py
|   ├── admin.py
|   ├── apps.py
|   ├── models.py
|   ├── tests.py
|   └── views.py
|
├── proj1/
|   ├── __init__.py
|   ├── asgi.py
|   ├── settings.py
|   ├── urls.py
|   └── wsgi.py
|
└── manage.py
```

go to settings.py in second “proj1” and add “app1” in the installed apps

go to settings.py in second “proj1” and add “ALLOWED_HOSTS=['35.154.168.20']” this is a public ip of the instance



Come out to first “proj1” folder and type

sudo nano runserver.py

//create the file “runserver.py” in parallel to manage.py and past the code below to change the port from 8000 to 8080

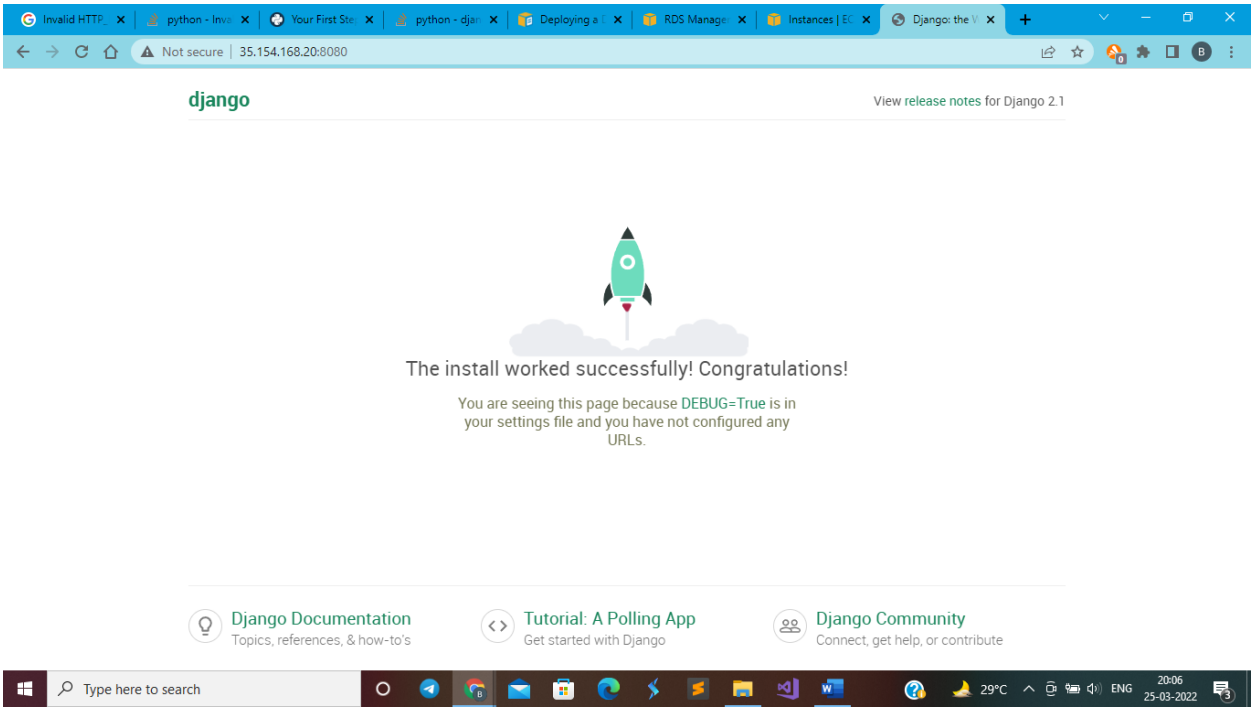
#!/bin/bash
exec ./manage.py runserver 0.0.0.0:8080

sudo chmod +x runserver.py

./runserver.py

// no need of this command “python manage.py runserver” due to above one

put “public_ip_address:8080” in browser



Now break the server and go to urls.py in your second proj1 directory and have these codes

Code

```
from django.contrib import admin

from django.urls import path, include

urlpatterns = [

  path('admin/', admin.site.urls),

  path('', include('app1.urls')),

]
```

Install “filezilla client” and connect your EC2

Now go to “app1” directory and past and replace the docs (urls.py, views.py, test.py, models.py, forms.py, apps.py, admin.py)

Inside app directory, create a folder called “templates” and past the below files (display.html, index.html, page2.html, upload.html)

Now in putty terminal, go to “apps.py” under app1 directory and replace the text “app” to “app1”.

Now go to “views.py” in app1 directory using terminal and open it.

Change this line “con = sql.connect(host='localhost', user='root', password='root', database='mydb');” with your data of RDS.

Now go to “settings.py” under second proj1 folder and add replace these code,

Code

```
DATABASES = {
   'default': {
      'ENGINE': 'django.db.backends.mysql',
      #'ENGINE': 'django.db.backends.sqlite3',
      #'NAME': BASE_DIR / 'db.sqlite3',
      'HOST': 'localhost',
      'USER': 'root',
      'PASSWORD': 'root',
      'NAME': 'mydb',
   }
}
```

In the above code, provide your RDS details

Then go to and edit the “__init__.py” file in your project origin dir(the same as settings.py)

Add code:

```
import pymysql
pymysql.install_as_MySQLdb()
```

Now, go to your terminal and install the following

```
pip3 install mysql-client
pip3 install mysql-connector-python
pip3 install pymysql
```

```
python3 manage.py makemigrations
python3 manage.py migrate
```

now run application as “./runserver.py”

you can able to communicate with the database now by adding “/upload” and show it by adding “/display” in the url.

Go to views.py and do the following

```
from django.http import JsonResponse

def corsget(request):
        sam_db = Sample.objects.values();
        return JsonResponse(list(sam_db), safe=False);

def postsave(request):
        print('welcome');
        if request.method == 'POST':
                print(request.POST.get('id'));
                id = request.POST.get('id');
                name = request.POST.get('name')
                cur.execute("insert into sample values (%s, %s)", (int(id),name));
                con.commit();
                return JsonResponse({"status": 'Success'})
```

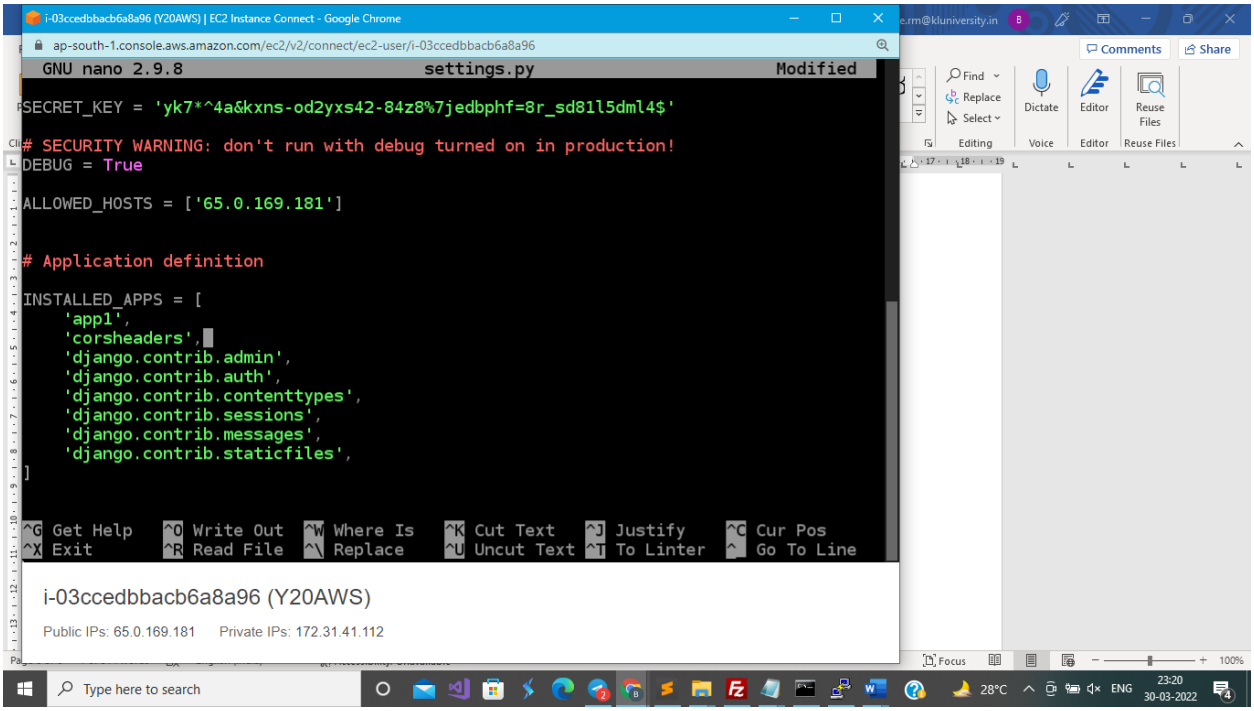
Go to app1’s “urls.py” and add the below routing

Go to terminal and install it

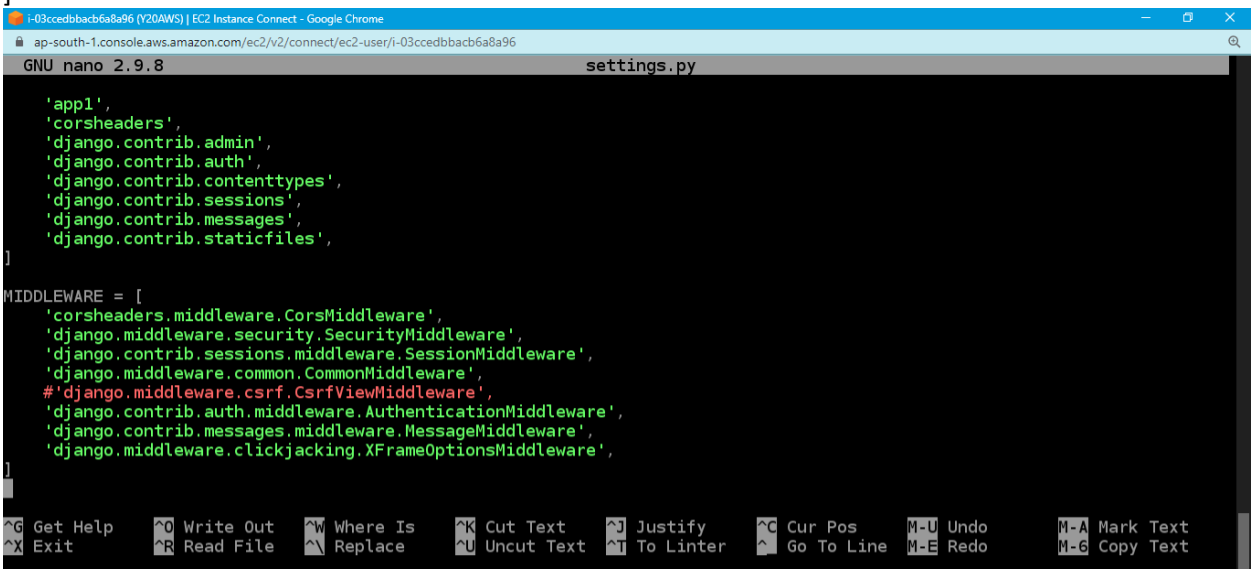
```
pip install django-cors-headers
```

Go to “settings.py” and do the following

```
INSTALLED_APPS = [
  ...
  'corsheaders',
  ...
]
```




```
MIDDLEWARE = [
    ...
   'corsheaders.middleware.CorsMiddleware',
   'django.middleware.common.CommonMiddleware',
   #'django.middleware.csrf.CsrfViewMiddleware',

]
```



If you want to allow access for all domains, set the following variable to TRUE in settings.py file: (before installed app, specify the line)

```
CORS_ORIGIN_ALLOW_ALL = True
```

Alternatively, you can specify which domains you want to give access to by doing the following in settings.py file:

```
CORS_ORIGIN_ALLOW_ALL = False
CORS_ORIGIN_WHITELIST = (
  'http://localhost:8000',
)
```