Hosting a Django Application on AWS
(Using EC2 and EFS)

Launching an Amazon EC2 Instance

1. Using the Amazon console, choose to the EC2 service.
2. Click on Launch Instance and choose the desired instance type. (Amazon Linux 2 AMI, t2.micro)
3. In the security group section, allow SSH requests so that the instance can be accessed via SSH over the internet and also allow HTTPS requests so that the hosted website can be accessed from the URL.

Connecting to an EC2 Instance using SSH

1. A user must have the private SSH key file (.pem) for that instance stored on their system.
2. Make a note of the public IP address of the instance and the username (default: ec2-user)
3. In your terminal (or SSH client terminal), type in the following command with the respective values
   ```
   ssh -i /path/key-pair.pem instance-user-name@instance-public-dns-name
   ```
4. You should now be connected to your EC2 instance.

Creating an EFS Volume

1. Using the Amazon console, choose to the EFS service.
2. Click on Create File System and configure the same as per your needs.
3. Make sure that the EFS is provisioned in the same region and subnet as your EC2 instance.
4. Set up a security group for your EFS such that it allows to your EC2 instance.

Mounting EFS

1. Make sure that the mount target state of your EFS is “Available” in the same subnet as your EC2 instance.
2. Run the following command to install the amazon-efs-utils package
   ```
   sudo yum install -y amazon-efs-utils
   ```
3. Create a directory onto which you want to attach your EFS.
   ```
   mkdir AppEFS
   ```
4. Click on the Attach button of your EFS and use the EFS mount helper command
   ```
   sudo mount -t efs -o tls efs-id:/ AppEFS/
   ```
5. Your EFS should be now mounted onto the ‘AppEFS’ directory.

Hosting an Application

1. Enter your EFS directory
   ```
   cd AppEFS
   ```
2. We will need to clone the Django Application repository which is to be hosted.
   ```
   sudo yum install git
   git clone repository-link
   ```
3. Enter the project directory and install the required dependencies (generally given in the requirements.txt file)
   ```
   pip install -r requirements.txt
   ```
4. Open the setting.py file of the project and add the EC2 public IP and DNS to the Allowed Hosts.
   ```
   sudo vim project-folder/settings.py
   >> ALLOWED_HOSTS=['public-ip','public-DNS',]
   ```
5. Run the Django Application Server on the 8000 port
   ```
   python3 manage.py runserver 0.0.0.0:8000
   ```

NOTE: To have the server running even after console is closed, use nohup (no hang up)

```
nohup python3 manage.py runserver 0.0.0.0:8000
```

Testing

1. To test the application, open a browser and append :8000 to the public IP of the EC2 instance.
   ```
   http://public-ip:8000
   ```
2. Django built in server only supports HTTP requests. So make sure you make a HTTP request and not a HTTPS request.

NOTE: You might have to give read-write permissions to your user for the project directory

```
chown user-name project-directory
```

Demonstration Video:

https://youtu.be/BULcbf7dAZw

Made By
Chaitanya Malani