**Deploying Django App on AWS**

**Report**

**Made by Muhammad Amjad**

Table of Contents

[Introduction: 3](#_Toc152034127)

[Manually Deployment: 3](#_Toc152034128)

[Step 1: Cloning the Project 3](#_Toc152034129)

[Step 2: Installing Django 3](#_Toc152034130)

[Step 3: Migrating pkgs 3](#_Toc152034131)

[Step 4: Running the Server 3](#_Toc152034132)

[Step 5: Adding Security Rules 3](#_Toc152034133)

[Step 6: Resolving Error 3](#_Toc152034134)

[Automatically Deployment (By using Dockers): 3](#_Toc152034135)

[Step 1: Installing Docker 3](#_Toc152034136)

[Step 2: Creating Docker File 3](#_Toc152034137)

[Step 3: Building Docker file 3](#_Toc152034138)

[Step 4: Running by using Docker 3](#_Toc152034139)

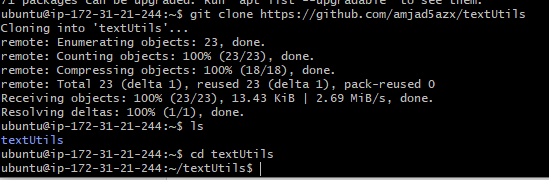
# Introduction:

Introducing TextUtils, a dynamic web application crafted on the Django framework that empowers users to effortlessly manipulate and enhance their text content. This versatile tool offers a range of essential text processing functions, including the removal of punctuations for cleaner prose, uppercase conversion for impactful messaging, new line removal, space removal for concise formatting, and a character count feature to keep tabs on textual length. TextUtils is deployed on AWS seamlessly, utilizing both manual and automated methods through Docker, ensuring a robust and efficient deployment process.

# Manually Deployment:

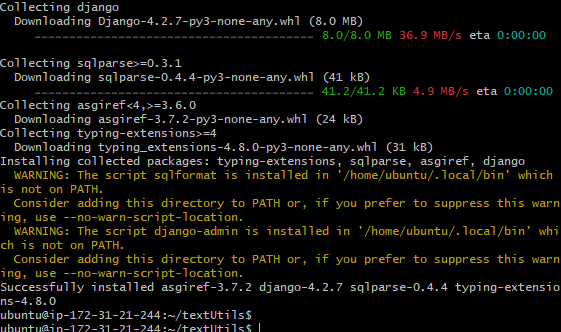
## Step 1: Cloning the Project

Clone the project repository to get started.



## Step 2: Installing Django

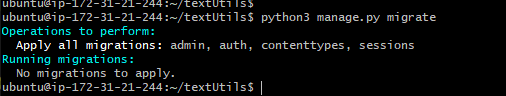
Install Django to environment to proceed with the setup.



## Step 3: Migrating pkgs

Migrate packages for Django to finalize the configuration with following command:

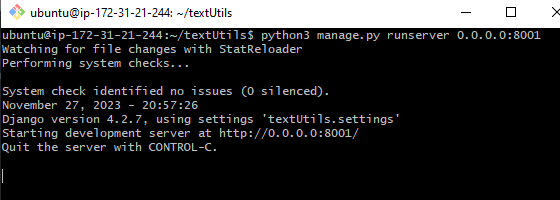
**python3 manage.py migrate**



## Step 4: Running the Server

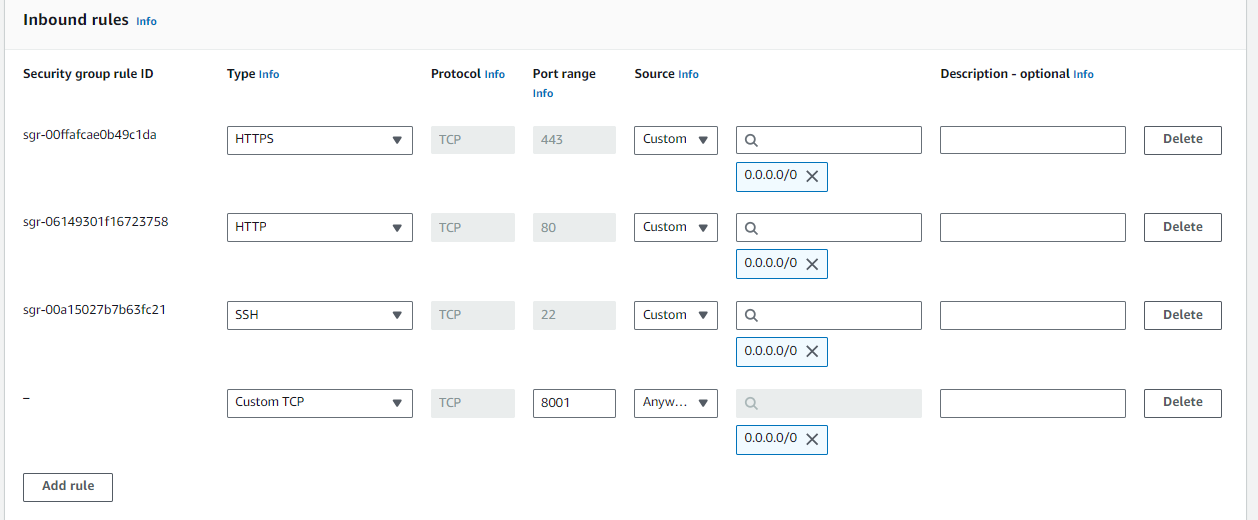
Launch the server with the following command:

**python3 manage.py runserver 0.0.0.0:8001**

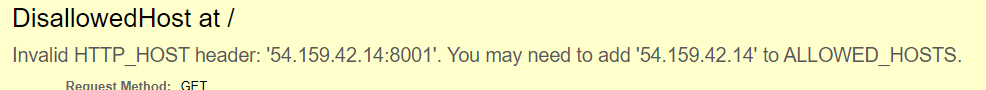


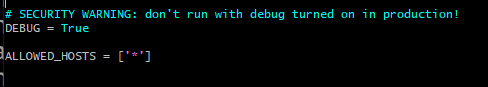
## Step 5: Adding Security Rules

In the Adding security rule menu of ec2 instance, a security rule is introduced by adding port in inbound rule menu.

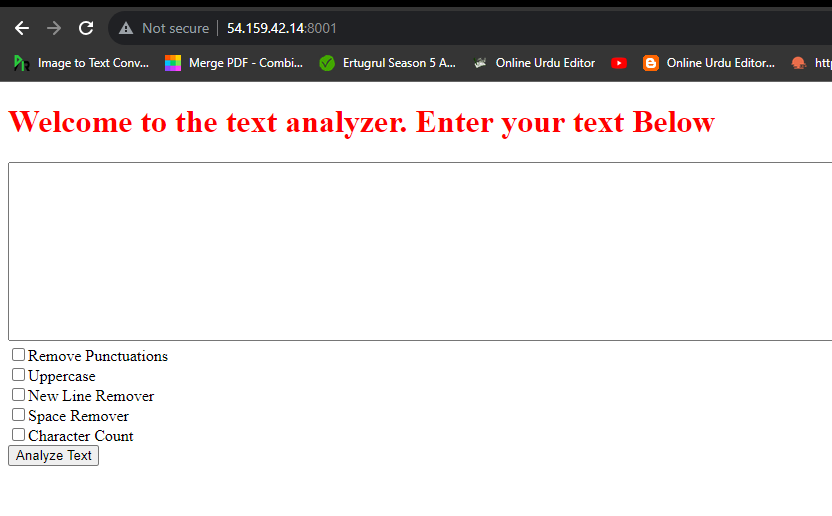


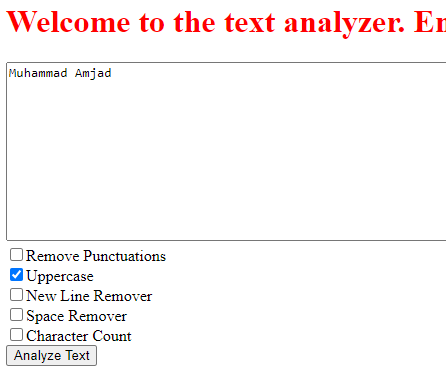
## Step 6: Resolving Error

To resolve a disallowed host error, I permitted all hosts by setting `ALLOWED\_HOSTS = ["\*"]` in the Django project's settings. This configuration allows the application to accept requests from any host.



After this Application is now running

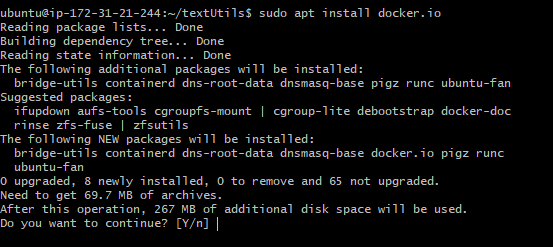


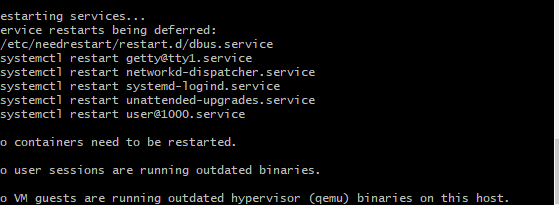


# Automatically Deployment (By using Dockers):

## Step 1: Installing Docker

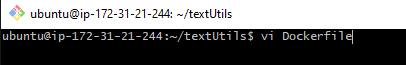
Successfully installed Docker using the command: `sudo apt install docker.io`.

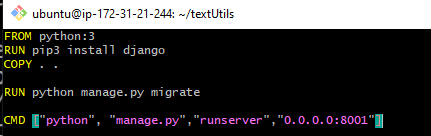




## Step 2: Creating Docker File:

The Dockerfile has been created, as shown below in snapshot.

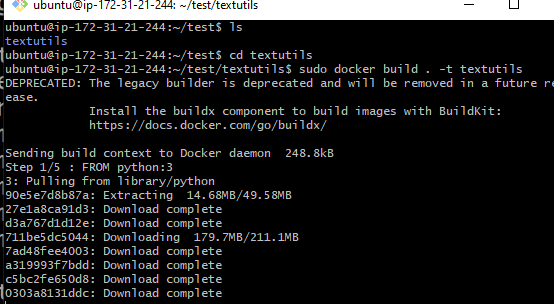




## Step 3: Building Docker file

Docker file is built by following command as shown below:

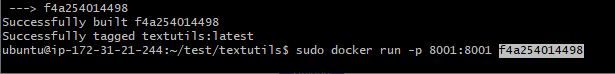
“sudo docker build . -t textutils”

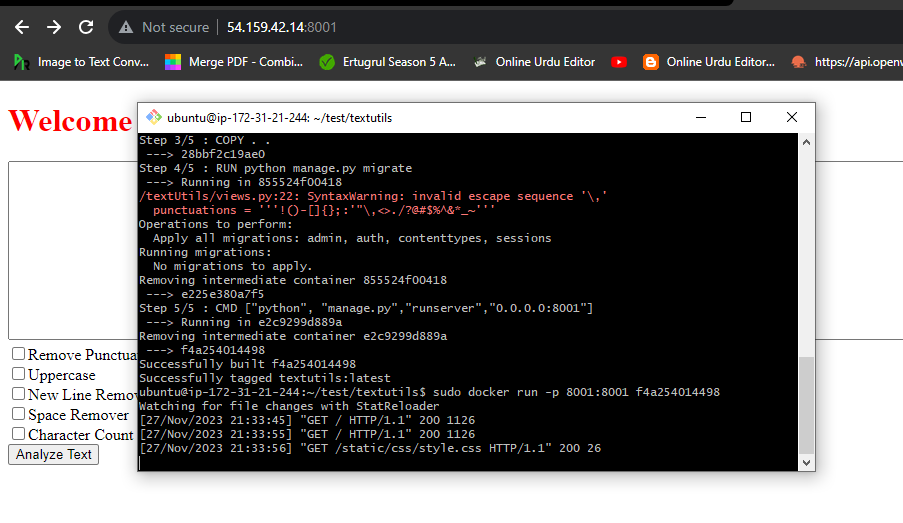


## Step 4: Running by using Docker

Docker file is run by following command as shown below:

“sudo docker run -p 8001:8001 f4a25014498”





For Running in background, you can use following code:

“sudo docker run -d -p 8001:8001 f4a25014498”

