**Hosting a Django App with Docker, Git/GitHub, AWS Elastic Beanstalk, and AWS**

By Kenyou Teoh/Gamer0813

In this guide, we will walk you through the steps to deploy a Django web application using Docker, Git/GitHub, AWS Elastic Beanstalk (EB), and Amazon Web Services (AWS).

Note: This is the very basic way to setting it up, if you want more control over it such as using the ECR, RDS, Route 53 stuff like that. Go figure it out yourself. This is for the beginner that just want to post their Django project online.

We also created a video list if you want steps by steps:

[Host Django with Dockerfile and AWS](https://www.youtube.com/playlist?list=PL63MaEJ40D8IwVY3kqoGu0QTutvfITZT6)

Also, Gamer0813 is Kenyou Teoh and Kenyou Teoh is Gamer0813.

**Prerequisites**

* **An AWS account**: If you don't have one, sign up for an AWS account at <https://aws.amazon.com/>
* **Django Application**: Have your Django application code ready for deployment.
* **Git/GitHub**: You got this.

[Step 1: Create and connect to EC2 instance.](#_Step_1:_Create)

[Step 2: Use Git to get Django Project](#_Step_2:_Use)

[Step 3: Dockerize your Django Project](#_Step_3:_Dockerize)

[Step 4: Install Elastic Beanstalk CLI](#_Step_4:_Install)

[Step 5: Use EB with IAM to host Django Project](#_Step_5:_Use)

# Step 1: Create and connect to EC2 instance.

Go to <https://us-east-2.console.aws.amazon.com/ec2/>

Go to launch instance.

图形用户界面, 应用程序

描述已自动生成

Type what you want it to be called, for example:

图表

中度可信度描述已自动生成

Use Ubuntu and use the default one that is free tier eligible.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

Select your instance and click on connect.

图形用户界面, 文本, 应用程序, Word

描述已自动生成

You want to be lazy just use:

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

I use SSH client.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

Copy underlined text. Go to the key you downloaded.

图形用户界面, 文本, 应用程序

描述已自动生成

Right click and open terminal.

图形用户界面, 应用程序, Word

描述已自动生成

文本

描述已自动生成

Paste the command, if question ask, answer yes.

If you see 文本

描述已自动生成

You made it.

# Step 2: Use Git to get Django Project

In the terminal type the following one line at a time:

mkdir Django

cd Django/

git init

git clone <https://github.com/Iforgotaboutitin2seconds/Sprint.git>

(replace the github with yours)

Then you are done.

You can do it your way if you know what this step means.

# Step 3: Dockerize your Django Project

Create Dockerfile at the root of your Django project.

Dockerfile:

# Use an official Python runtime as a parent image

FROM python:3.8-slim

# Set environment variables for Python (unbuffered mode) and Django

ENV PYTHONDONTWRITEBYTECODE 1

ENV PYTHONUNBUFFERED 1

# Set the working directory in the container

WORKDIR /app

# Copy the requirements file into the container at /app

COPY requirements.txt /app/

# Install any needed packages specified in requirements.txt

RUN pip install --no-cache-dir -r requirements.txt

# Copy the current directory contents into the container at /app

COPY . /app/

# Expose port 80 for the Django development server

EXPOSE 80

# Run the Django development server

CMD ["python", "manage.py", "runserver", "0.0.0.0:80"]

Execute the following commands to install and enable docker and change permission in the EC2.

sudo apt-get update

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install -y docker-ce docker-ce-cli containerd.io

sudo usermod -aG docker ubuntu

sudo systemctl enable docker

sudo systemctl start docker

Restart your connection, exit and connect again.

The following commends is to build your docker image and check that your Django project is successfully Dockerize.

docker build -t music .

docker run music

# Step 4: Install Pip and Elastic Beanstalk CLI

Follow the commands to install pip and ebcli:

sudo apt install python3-pip

pip install awsebcli

vim ~/.bashrc

add the following line to the end of the file:

export PATH="/home/ubuntu/.local/bin:$PATH"

exit and reconnect to the EC2.

Type eb to confirm it is installed.

# Step 5: Use EB with IAM to host Django Project

You need a IAM user to setup the EB init

Go to the IAM <https://us-east-1.console.aws.amazon.com/iam>

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成 图形用户界面, 应用程序

描述已自动生成 图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

Copy down your key or download it.

图形用户界面, 文本, 应用程序

描述已自动生成

#Get awscli

sudo apt install awscli

#Configure things this to enter your access keys and default region

aws configure

#Make sure you are at the root of your Django Project

#Just like github

eb init -p docker music-env

#This will auto start deploy

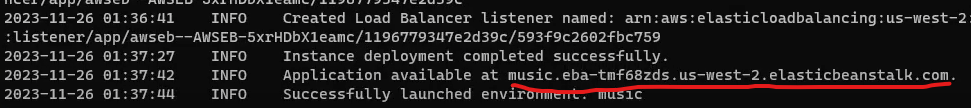
eb create music-dev

#Use this when you updated your code and want to apply the changes to the website that is already deployed

eb deploy

#Use this to see the url to your app

eb status



Copy the url and paste it in your browser.

And you done!