**AWS 101 EBS : Python Django Application Deployment using Elastic Beanstalk**

*Posted On September 11, 2019*

**Prerequisites**

To start building and executing Django app on AWS, you should have following packages installed:

Python 3.6 Pip3 Virtualenv awsebcli AWS Account on <https://aws.amazon.com/>

**Setup GIT, Virtual Environment & Django**

**Setup Virtual Environment**

1. Install virtual environment using >$ **pip install virtualenv** , where $ On Windows this represent C:\Users\Username\> 2. Navigate to Project Folder and run command, >$**virtualenv env** This will create env is alias of virtual environment created inside project folder. 3. To Activate/Start virtual environment , $ %ProjectFolder% > **env\Scripts\Activate** 4. env would be prefixed to current path, (env) C:\>

**Install Django** & Create a Django Project

1. Use Pip to install Django, (env) $ **pip install django==2.1.1** To verify that Django has been installed, type (env) $ **pip freeze** **To generate a Django application** 1. Activate your virtual environment

2. Use the django-admin startproject command to create a new Django project named , for example, “DjangoApp”

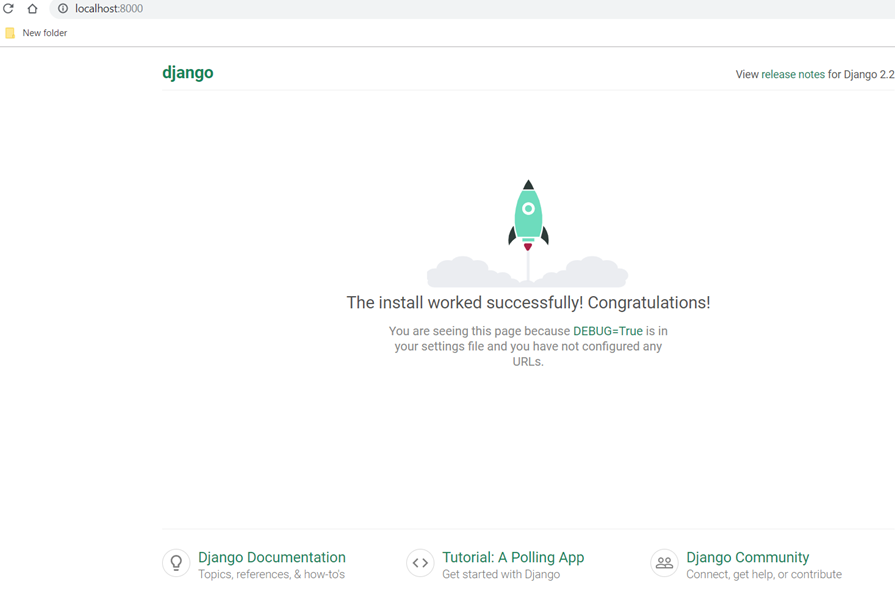
3. (env) C:\Users\Username>**django-admin startproject DjangoApp**

4. Run Project Locally :-> (env) C:\Users\Username>cd DjangoApp

5. You can verify its installation by typing located inside the  virtual environment:- (env) C:\Users\Username\DjangoApp>django-admin –version 2.2.5

6. Run $**python manage.py migrate**

7. Run $**python manage.py runserver**



**Initialize Git**

1. If required to use Git, initialize git from virtual environment **(env) git init** 2. Create gitignore(on Windows) 3. Open Text Editor, Move to File> New File> Save > Save as .gitignore 4. Include following components to .gitignore file bin/ include/ lib/ \*.py[cod] \*.sqlite3 pip-selfcheck.json

**Install AWS Command Line Interface: EBS CLI**

1. Ensure virtualenv is activated

2. pip install awsebcli — Command to install AWS CLI 3. To check installation version:- (env) C:\Users\Username\DjangoApp>**eb –version** EB CLI 3.15.3 (Python 3.7.3)

**Configure your site for Elastic Beanstalk**

1. From active virtual environment run the command , Run pip freeze and save the output to a file named requirements.txt (env) C:\Users\Username\DjangoApp> pip freeze > requirements.txt

2. Create a new folder called .ebextensions in the root of virtual environment.

3. Add a configuration file django.config with the following text: option\_settings: aws:elasticbeanstalk:container:python: WSGIPath: DjangoApp/wsgi.py This setting, WSGIPath, specifies the location of the WSGI script that elastic Beanstalk uses to start application

4. Within the .ebextenstions directory, add **Git Commit** 1. git add . 2. git commit -m “Comment”

**Create AWS User Credentials**

1. Navigate to [IAM Users](https://console.aws.amazon.com/iam/home?) at console.aws.amazon.com

2. Select Users > Add User

3. Enter awsuser as a username (or whatever you prefer)

4. Download credentials and keep them safe.

5. Credentials.csv file that was just downloaded/created, contains the Access Key Id and Secret Access Key as they are needed for a future step. These will be referred to as <your\_access\_key\_id>  and <your\_secret\_access\_key>

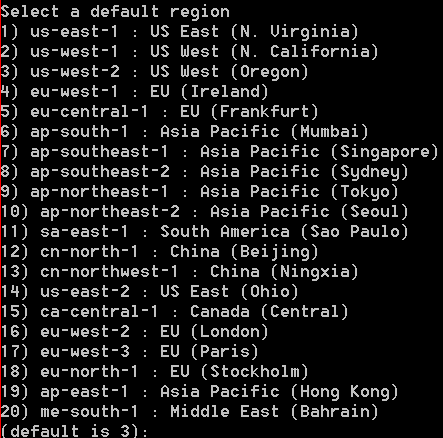
**Create Elastic Beanstalk Application via Command Line**

The first thing we want to do is create a Beanstalk environment to host the application on.

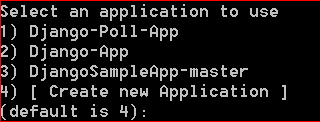
1. Run this from the project directory. **$eb init**

2. This will prompt you with a number of questions to help you configure your environment like,

Default region



Application Name:



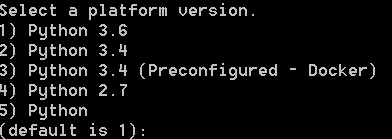
SSH:

https://ticksteps.com/insight/wp-content/uploads/2019/09/SSH.png

RSA Keypair:



Python Version:



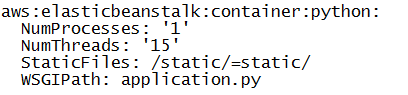
3. Create an environment and deploy you application to it with  $**eb create** **django-env** Once the environment is created, the environment details can be found by running $**eb status**

4. Edit the settings.py file in the DjangoApp directory, locate the ALLOWED\_HOSTS setting ALLOWED\_HOSTS = [‘*eb-django-enc-dev.elasticbeanstalk.com*‘] In case of production version of settings.py , add AWS\_ACCESS\_KEY\_ID = "<your\_access\_key\_id>" AWS\_SECRET\_ACCESS\_KEY = "<your\_secret\_access\_key>"

5. If you are using Git with your project , Commit to Git all the changes done. $ git add . $ git commit -m “Updates Settings” We have to commit code changes as the deployment eb deploy command will work on latest file only when it is committed.

6. eb deploy — This the command that will deploy code to AWS EBS

**Configuring our Python environment**

AWS EC2 instance on EBS will find our application by its WSGIPATH that we have set. WSGI file is assumed as application.py by default. This has to be corrected. Run command >$**eb config** Opens deployed app configuration file as shown below:-  Instead of application.py, it has to be changed to DjangoApp/DjangoApp.py aws:elasticbeanstalk:container:python: NumProcesses: ‘1’ NumThreads: ’15’ StaticFiles: /static/=static/ WSGIPath: DjangoApp/DjangoApp.py This step will help correct Error like:- ERROR Your WSGIPath refers to a file that does not exist