Django Class Notes

Clarusway

Django Deployment to Elastic Beanstalk

Nice to have VSCode Extentions:

- Djaneiro - Django Snippets (Be carefull about other conflicting extentions!)

Needs

- Python 3.7 or later (add the path environment variable while installing)
- pip
- virtualenv
- awsebcli

Summary

- What is Deployment
- What is AWS and Elastic Beanstalk
- Install awsebcli
- Create project
- Deploy your site with the EB CLI
    - Update and Deploy
    - Clean up
- DNS management
- Certificate management

What is Deployment

Info about deployment

What is AWS and Elastic Beanstalk

Sign up to the AWS if you want to join the session interactively.

How to set up AWS account

AWS as a cloud provider, has some advantages:

- Leader
- On-demand
- Pay as you go
- No upfront cost
- Compute, storage, db etc.
- Free tier experimentation, minimal cost
- Regions, az's
- Scalable
- Available
- Fault tolerant

https://aws.amazon.com/elasticbeanstalk/?nc1=h_ls

https://www.youtube.com/watch?time_continue=8&v=uiM1xzOX8Qg

Install awsebcli

Manually install the EB CLI: https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/eb-cli3-install-advanced.html

Create project

- Shorten your powershell terminal prompt:

```
Function Prompt { "MyCode: " }
```

- Create a working directory, name it as you wish, cd to new directory
- Create virtual environment as a best practice:

```
python3 -m venv eb-virt # for Windows or
python -m venv eb-virt # for Windows
virtualenv eb-virt # for Mac/Linux or;
virtualenv eb-virt -p python3 # for Mac/Linux
```

- Activate scripts:

```
.\eb-virt\Scripts\activate  # for Windows
source eb-virt/bin/activate  # for MAC/Linux
```

- See the (eb-virt) sign before your command prompt.
- Install django:

```
# This version is important, may change in the future
# this is about elastic beanstalk configuration
pip install django==2.2
```

- See installed packages:

```
pip freeze

# you will see:
Django==2.2
pytz==2021.1
sqlparse==0.4.1

# If you see lots of things here, that means there is a problem with your virtual
env activation.
# Activate scripts again
```

- Create requirements.txt same level with working directory, send your installed packages to this file, requirements file must be up to date:

```
pip freeze > requirements.txt
```

- Create project:

```
django-admin startproject ebdjango .
# With . it creates a single project folder.
# Avoiding nested folders
```

- Various files has been created!
- Check your project if it's installed correctly:

```
python manage.py runserver
py -m manage.py runserver
```

- Create a directory named .ebextensions.

```
mkdir .ebextensions
```

- In the .ebextensions directory, add a configuration file named "django.config" with the following text.

```
option_settings:
  aws:elasticbeanstalk:container:python:
    WSGIPath: ebdjango.wsgi:application
```

- Don't forget to chage project name on "WSGIPath: .wsgi:application" if you use different one.
- Deactivate your virtual environment with the deactivate command.

```
deactivate
```

Deploy your site with the EB CLI

There are two ways to deploy Django to ElasticBeanstalk,

- Deployment from dashboard using zipped project file
- Deployment using CLI (recomended)

Manual Deployment

- Open the project folder
- Select all files excluding virtual env folder
- Zip that files, do not to zip the project folder itself!
- Open ElasticBeanstalk dashboard
- Create environment, copy hostname, and paste hostname to allowed hosts on settings.py
- Upload and deploy with a new zip file after making changes

CLI Deployment

You've added everything you need to deploy your application on Elastic Beanstalk.

- Initialize your EB CLI repository with the eb init command.

```
# -p PLATFORM, --platform PLATFORM (default Platform)
# -r REGION, --region REGION (use a specific region)
# eb init <application_name> [options ...]
eb init -p python-3.7 django-tutorial -r us-east-1
```

- (Optional) Run eb init again to configure a default key pair so that you can use SSH to connect to the EC2 instance running your application.

```
eb init
```

- Create an environment and deploy your application to it with eb create.

```
# eb create <environment_name> [options ...]
eb create django-env -r us-east-1
```

- When the environment creation process completes, find the domain name of your new environment by running eb status.

```
eb status
```

- Open the settings.py file in the ebdjango directory. Locate the ALLOWED_HOSTS setting, and then add your application's domain name that you found in the previous step to the setting's value. If you can't find this setting in the file, add it to a new line.

```
ALLOWED_HOSTS = ['eb-django-app-dev.elasticbeanstalk.com']
```

- Save the file, and then deploy your application by running eb deploy. When you run eb deploy, the EB CLI bundles up the contents of your project directory and deploys it to your environment.

```
eb deploy
```

- When the environment update process completes, open your website with eb open.

```
eb open
```

Update and Deploy

Every time you update your project, deploy your changes!

- Create app

```
python manage.py startapp app
```

- Go to settings.py and add the app to the INSTALLED_APPS:

```
'app'
```

- Create url of project
- Go to the urls.py on project, add a new path

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('app.urls')),
]
```

- Create urls.py on app

```
from django.urls import path
from .views import home

urlpatterns = [
    path('', home, name='home'),
]
```

- Go to views.py in app
- Create home view by adding:

```
from django.shortcuts import render

def home(request):
    return render(request, "app/home.html")
```

- Create app/templates/app directory and create a home.html file under it:

```
<h1>Welcomee to the real World!</h1>
```

- Deactivate

```
deactivate
```

- Save the file, and then deploy your application by running eb deploy.

```
eb deploy
```

- Refresh the page and see the changes.

Clean up

To save instance hours and other AWS resources between development sessions, terminate your Elastic Beanstalk environment with eb terminate.

```
eb terminate django-env
```

- Check you AWS account if there is any resource left. Be carefull about the region, search for the resources on the region which elastic beanstalk deployed the project.

DNS management

https://aws.amazon.com/getting-started/hands-on/get-a-domain/

Certificate management

https://aws.amazon.com/certificate-manager/ https://www.youtube.com/watch?v=Ge-dkZgqLKg https://www.youtube.com/watch?v=_otcYm8RVHA https://aws.amazon.com/premiumsupport/knowledge-center/elastic-beanstalk-https-configuration/

Sources

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create-deploy-python-django.html https://www.ordinarycoders.com/blog/article/deploying-from-aws-eb-consolev CLI deployment for mac users: https://www.starwindsoftware.com/blog/deploying-django-project-to-aws-elastic-beanstalk

Next Steps

- Implement what you are learned from this lesson to your own personal website
- Set up a CI / CD pipeline, with one or more stages:
    - Lint
    - Test
    - Deploy
    - Scan
    - Release
- Add your project to your github, and resume to show to recruiters
- Good luck Full Stack Developers!!!

☺ **Happy Coding!** ✍