AWS Elastic Beanstalk Deployment Instructions Personal

- Create AWS account following these instructions: https://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account/
- Install the EB CLI using the instructions found here:
    - https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/eb-cli3-install.html
- Configure Django for deployment using the instructions found here:
    - https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create-deploy-python-django.html
    - This is a general list of steps you'll need to take:
        - Create & Activate your Python virtual environment
        - Install Django related dependencies in that environment
        - Create a requirements.txt file with your dependencies
            - pip freeze > requirements.txt
                - EB will use this file to figure out which packages your application needs
        - Create a .ebextensions directory in your project's root directory
        - Add django.config file in .ebextensions with the following settings

```
option_settings:

  aws:elasticbeanstalk:application:environment:
```

```
    DJANGO_SETTINGS_MODULE: "MY_APP.settings"

    PYTHONPATH: "/var/app/current:$PYTHONPATH"

  aws:elasticbeanstalk:container:python:

    WSGIPath: MY_APP.wsgi:application
```

    - Don't forget to add EBS domain name to the list of ALLOWED_HOSTS in the Django app.

NOTE: the environment variable set up doesn't seem to be included in AWS documentation, you can refer to this page instead: https://testdriven.io/blog/django-elastic-beanstalk/

- Deploy your app to EBS by ("**Deploy your site with EB CLI section**"):
    - Initialize your EB CLI
    - Configure SSH access to EC2 instances
    - Create environment for your project
    - Deploy the app
- If you get an error saying that the nodes are degraded because there's no default VPC, navigate to the AWS VPC panel, and create a new default VPC. AWS will create a default VPC and related infrastructure (subnets, route tables, security groups, etc.).
- As an additional safeguard against unanticipated service charges, you can set up a Cloud Watch, to alert you whenever the charges on your account exceed a particular amount. To do so, please follow instructions on this link:

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/monitor_estimated_charges_with_cloudwatch.html

Miscellaneous notes:

- Please read the documentation. If you get stuck, try Googling the error.
- You need to create an AWS account. One account per team is enough. AWS is using IAM roles to manage account access. The proper way to configure permissions would be to assign a separate role to each team member with whatever permissions that member needs. An easier, less proper approach would be to just share the account password with each other and change it when semester is over.
- Don’t check your keys into your source code. If someone gains access to your root AWS account, he might, for example, try to use the resources to mine Bitcoins, until your account gets cut off with 10 000$ charge on it…