Deployment 1 Notes by Kingman Tam

Steps taken:

1- Create EC2 with security groups:

#While creating the EC2, specific ports (22, 80, 8080) needed to be opened so that there is limited remote #access into the EC2. Having all ports open may lead to vulnerabilities or at least difficulty tracking #traffic.

2- Update and install JRE (Java Runtime Environment) on EC2:

#Since Jenkins is a Java application, the EC2 needs to have JRE installed so that it has the ability to run #the program.

3- wget jenkins install file and decrypt with gpg:

#The Jenkins program itself needs to be downloaded onto the EC2. gpg is used for security and #encryption. File is likely encrypted and 'gpg –dearmor' command is used to decrypt it

4- Install and start Jenkins

After the installation file is downloaded, Jenkins can be installed onto the EC2 and then started with the # "systemctl start jenkins" command. This is verified with the "systemctl status jenkins" command.

5- Configure Jenkins

- Jenkins needs to be unlocked with a password that was generated in a "secret" file called "initialAdminPassword"
- Plugins are installed so that Jenkins can utilize other programs within it
- First time users need to create a username and password to access Jenkins
- Amazon EC2 plugin is installed so that Jenkins can be integrated with Amazon

6- Configure a (EC2) cloud

Now that Jenkins is running and the plugin for Amazon EC2 is installed, the EC2 instance must be set up to be used as Jenkins agents.
AWS credentials including access key, secret access key, region, username, and the private key from the .pem file needed to be input
A lot of trial and error occured while trying to complete this step. Different options in the drop down menus were tried before realizing which one was the proper one that worked. Each time I tried to "Test Connection" with an incorrect input, it would return "ERROR".

7- Install pip and venv in EC2

The EC2 was configured to run Ubuntu which doesn't come with python virtual environments. Python 3-pip is installed for its ability to install python packages.
Python 3.10-venv (NOT python3-10-venv as written in the instructions!) is used to create an environment with all the proper dependencies that "application.py" can run in.

8- Connect GitHub repo to Jenkins

Jenkins needs to be connected to the GitHub repository of the application so that it can read/build/test the files stored on the repo.

To connect Jenkins to the GitHub Repo, a unique access token must be created on GitHub and then added to a “multibranch pipeline” item. This step also requires the GitHub username and Repository URL.

9- Jenkins “builds” program and tests it

After the application’s GitHub Repo is added to Jenkins, it “builds” the application as instructed in the “Jenkinsfile” in the repo.

Jenkins also uses the instructions in the “test_app.py” to run a test on the “application.py” file.

The test file in this deployment was written to always pass. Future deployments will be different.

10- GitHub Repo is compressed into zip file

If the files in the repo pass the tests run by Jenkins, it is ready to be moved into the next (production?) environment.

All of the files INCLUDING the hidden files in the repository need to be compressed into a zip file so that it can be uploaded to AWS Elastic Beanstalk

11- AWS Elastic Beanstalk creates web application with zip file and generates URL

The zip file is uploaded in a “Web Server environment” and AWS Elastic beanstalk configures and compiles the application.

When the action is completed, a URL is supplied that will bring the user to the website where the application is hosted/live.

What could I improve?

This entire process was a lot of steps that were not initially clear to me. Listing and commenting out these steps here really helped me understand the importance of each action. With practice and repetition I think the entire process will become more familiar and completed with ease. Some parts of the process could also be automated in a script so that one could just run the script and it would go through the steps automatically. These steps would include ones such as downloading and installing Jenkins and the python files on to the EC2.