So Deployment here will be very similar to previous deployment we have done before using GitHub action and self-hosted run with a combination of AWS.

1. So let check the prerequisite. You will be using Docker so make sure docker is install in your system

* Docker –version



1. Now that we have verified that we have a docker. We can now build our docker file

A screen shot of a computer

Description automatically generated

1. Now that we have created our dockerfile. We can test it by building our docker image. Let call it stt\_test

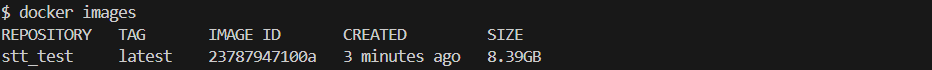
- docker build -t stt\_test .

NOTE: If you are confused, please go to C:\Users\midof\OneDrive\Desktop\INeuron\Dockers\Guide. This is where you can learn everything about docker and how to build a docker image and more. Matter of fact the instructions here are copied and pasted here.

1. Now if we do

* Docker images

We will be able to see the image we just built

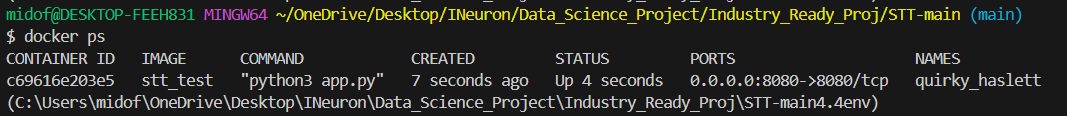
* 

1. We can now run the docker images which we just build in cmd line

- docker run -d -e AWS\_ACCESS\_KEY\_ID=AKIA2UC274VORSAISXES -e AWS\_SECRET\_ACCESS\_KEY=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 -p 8080:8080 stt\_test

6. With the below cmd, you can now see our image running in the background

- docker ps



7.Anoter way to check if our docker is running or not is to check the logs. This is particularly useful if the docker got an error meaning when we do docker ps, we can see it running

- docker logs c69 (WHERE C69 is the first 3 characters of our docker container ID)

A black screen with white text

Description automatically generated

8. Now if we run our application. We would be able to see it because we are running it locally

- <http://localhost:8080/>

9. You can shut it down now that we have verified it is working. **Now imagine we want the project to be accessible to everyone. This is why we will deploy it to AWS**

10. Now let work our magic in AWS. Because we have a user, and we have some policy attached. We will skip all the into stuff since this is not our first time using AWS

A screenshot of a computer

Description automatically generated

11. As you can see. Only 1 policy we don’t have, and we need that since we will be dealing with docker and that is ECR. So let get it.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

12. We can now attach that policy we created to myself (the user)

A screenshot of a computer

Description automatically generated

13. KABOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOM

A screenshot of a computer

Description automatically generated

14. Let do our EC2 and self-hosted runner setup. Again, all these shenanigans you are familiar with. So, if you don’t remember. I’ll advise you check out C:\Users\midof\OneDrive\Desktop\INeuron\Data\_Science\_Project\Industry\_Ready\_Proj\Sensor\_Detection\_Projec\Deployment\_Steps

OR

C:\Users\midof\OneDrive\Desktop\INeuron\Data\_Science\_Project\Industry\_Ready\_Proj\CNN\Search\_Engine\search-engine-data-collection\AWS-MongoDB-Setup

Especially AWS-MongoDB-Setup because it has the security group problem solved on there.

15. After everything has been set up. We need to set up our container registry since we are doing deployment in other for the tool to be accessible to everyone.

Again. You can see other examples on how to create a repository which is pretty easy: C:\Users\midof\OneDrive\Desktop\INeuron\Data\_Science\_Project\Industry\_Ready\_Proj\CNN\Search\_Engine\search-engine-data-collection\AWS-MongoDB-Setup

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

16. Now we can see our push cmds to build and push our dockers just like I was doing in my local machine but in this case, we will be doing it in the cloud so it can be accessible to everyone.

**NOTE**: This will be done programmatically in our GitHub workflow so no need to manually do that.

A screenshot of a computer

Description automatically generated

17. Finally make sure you have your keys in your repo secret in github

A close-up of a white box

Description automatically generated

18. Now Git add and PUSHHHHHHHHHHHHHHHHHHHHHHHHHH