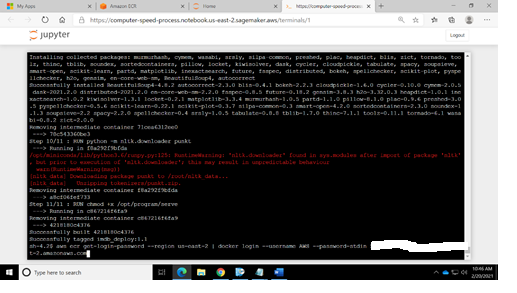


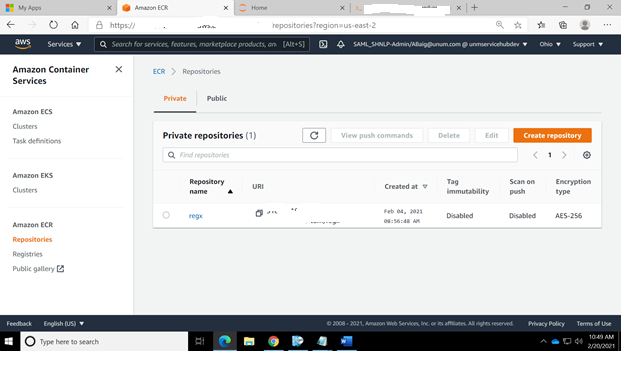
Navigate onto the folder where the dockerfile is there and build the docker using the above command; docker build -t <image\_name> -f .

The docker file contains instruction about preparing the image; this image contains our deployment scripts.

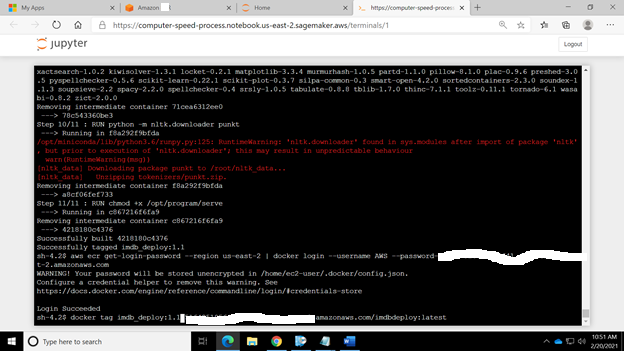
The Next step is to login to the Amazon ECR; once you create the repository in the ECR you would find all the commands for push; in once such command you would find the details for login.



We need to create a ECR repository



Now with the already built docker ; we need to tag that to this ECR Repository

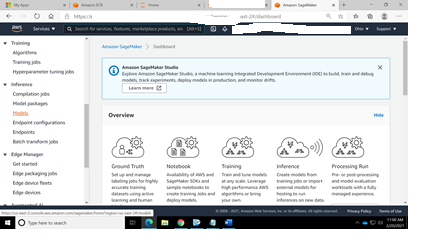


And then simply push the repository to the remote. ( you will find the commands for the same in the ECR Repository; just keep in mind the names used)

This is going to take some time.

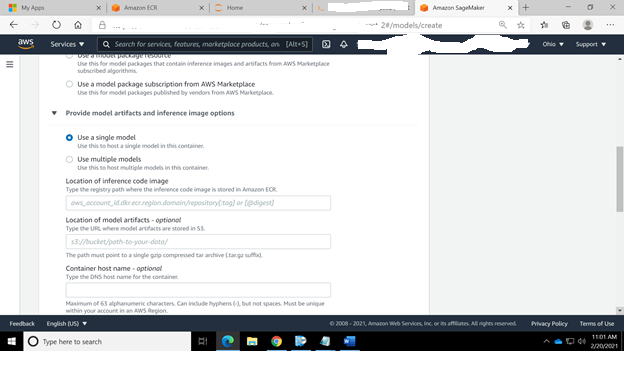
When the docker is pushed to remote ECR Repository you will find that there in the repository, this image would be generated, which would be useful as it would be deployed on the machine of our choice where the model would be hosted. A few good practices are to make this image as light as possible in size. Keep a note of the image uri, as it would be used in the next steps.

As we are deploying the model on the sagemaker; we need to create an endpoint on the sagemaker; navigate on to the sagemaker and click on Models.



Under Models; click on create Model and fill out a name for the model;

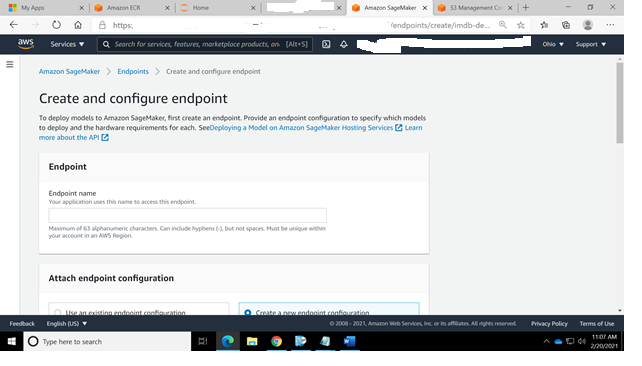
Leave everything as is and just fill out two fields.



those two are the location of the inference code image and location of model artifacts; inference code image is nothing but the image uri which you pushed.

And the model artifacts could be your pickle file but in tar.gz format; this needs to be uploaded to an S3 container and from there you could get the uri of that. This is done so that in real time with every image deployed on the machine a model artifacts is also fetched and kept at a specific location on that image.

Once that is done click on create model.



Now for that Model we need to create an Endpoint. This will create that container; now we could call and end-point as a service

import boto3

import sagemaker

from sagemaker import get\_execution\_role

role=get\_execution\_role()

import time

sagemaker = boto3.client('runtime.sagemaker')

test\_sample='{"input": {"text": "I love the movie"}}'

strt=time.time()

ENDPOINT\_NAME=’name of the deployed end point'

response=sagemaker.invoke\_endpoint(EndpointName=ENDPOINT\_NAME,ContentType='application/json',Body=test\_sample)

print("time taken for WS response is ",time.time()-strt)

print(response['Body'].read().decode())