Name : Jagadeesh jyothula
Id : 2000032181
Sec : 15

Cloud and serverless computing project

Deploying ML models using flask , Docker and Kubernetes

INTRODUCTION:

To Build a ML Web application, Dockerized and Deploy on Kubernetes

FLOW OF THE PROJECT DEPLOYMENT:

STEPS:

1 Build your Flask app

2 Dockerize the Flask app by creating a Docker file.

3 Build a Docker image of your Flask app.

4 Push the Docker image to a Docker registry (e.g., Docker Hub) 5 Create a Kubernetes cluster on AWS

6 Deploy a Kubernetes deployment object to run your Docker image as a container in the cluster

7 Create a Kubernetes service object to expose your Flask app to the internet

Review 1:

Create an nueral network model using python framework Tensorflow

Review 2:

Create an Front end for the model using Web app Frameworks like Flask or Django

Review 3:

Create a Docker File By installing all the dependencies required by the webapp and the model create a Docker image of the working app

Review 4:

Push the Docker image to a Docker registry (e.g., Docker Hub)

Create a Kubernetes cluster on AWS

Deploy a Kubernetes deployment object to run your Docker image as a container in the cluster

REFERENCE LINKS:

https://aws.amazon.com/blogs/opensource/automate-python-flask-deployment-to- the-aws-cloud/

implementation:

-> implementing the meachine learning code into the flask

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

app.py Model_Train.ipynb

Users > macbook > Downloads > Image-Classification-Webapp > app.py

```python
from flask import Flask, render_template, request
from keras.models import load_model
from keras.preprocessing import image

app = Flask(__name__)

dic = {0 : 'Cat', 1 : 'Dog'}

model = load_model('model.h5')

model.make_predict_function()

def predict_label(img_path):
    i = image.load_img(img_path, target_size=(100,100))
    i = image.img_to_array(i)/255.0
    i = i.reshape(1, 100,100,3)
    p = model.predict_classes(i)
    return dic[p[0]]


# routes
@app.route("/", methods=['GET', 'POST'])
def main():
    return render_template("index.html")

@app.route("/about")
def about_page():
    return "Please subscribe  Artificial Intelligence Hub..!!!"
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 1, Col 1 Tab Size: 4 UTF-8 CRLF Python

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

app.py | Model_Train.ipynb

Users > macbook > Downloads > Image-Classification-Webapp > app.py

```python
# routes
@app.route("/", methods=['GET', 'POST'])
def main():
    return render_template("index.html")

@app.route("/about")
def about_page():
    return "Please subscribe  Artificial Intelligence Hub..!!!"

@app.route("/submit", methods = ['GET', 'POST'])
def get_output():
    if request.method == 'POST':
        img = request.files['my_image']

        img_path = "static/" + img.filename
        img.save(img_path)

        p = predict_label(img_path)

    return render_template("index.html", prediction = p, img_path = img_path)


if __name__ =='__main__':
    #app.debug = True
    app.run(debug = True)
```

Restricted Mode ⊗ 0 ⚠ 0 Ln 1, Col 1 Tab Size: 4 UTF-8 CRLF Python

->summary

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

app.py | Model_Train.ipynb

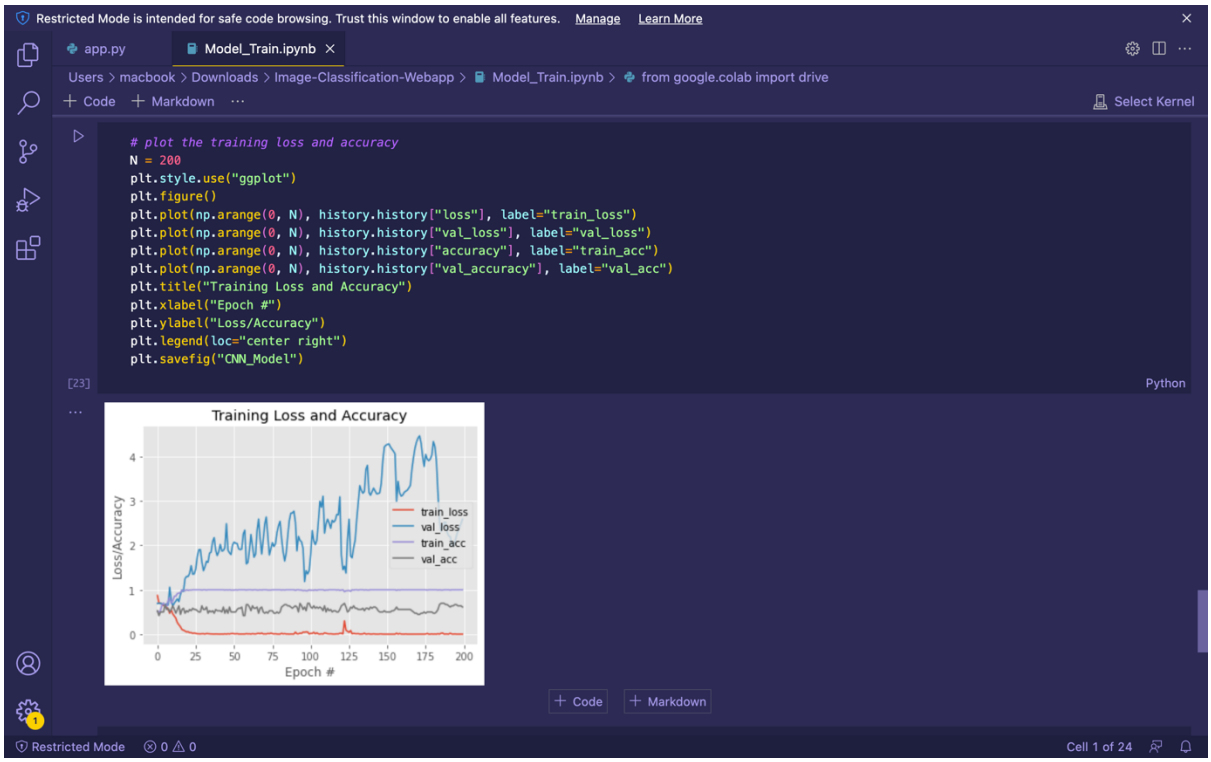
Users > macbook > Downloads > Image-Classification-Webapp > Model_Train.ipynb > from google.colab import drive

+ Code + Markdown ··· Select Kernel

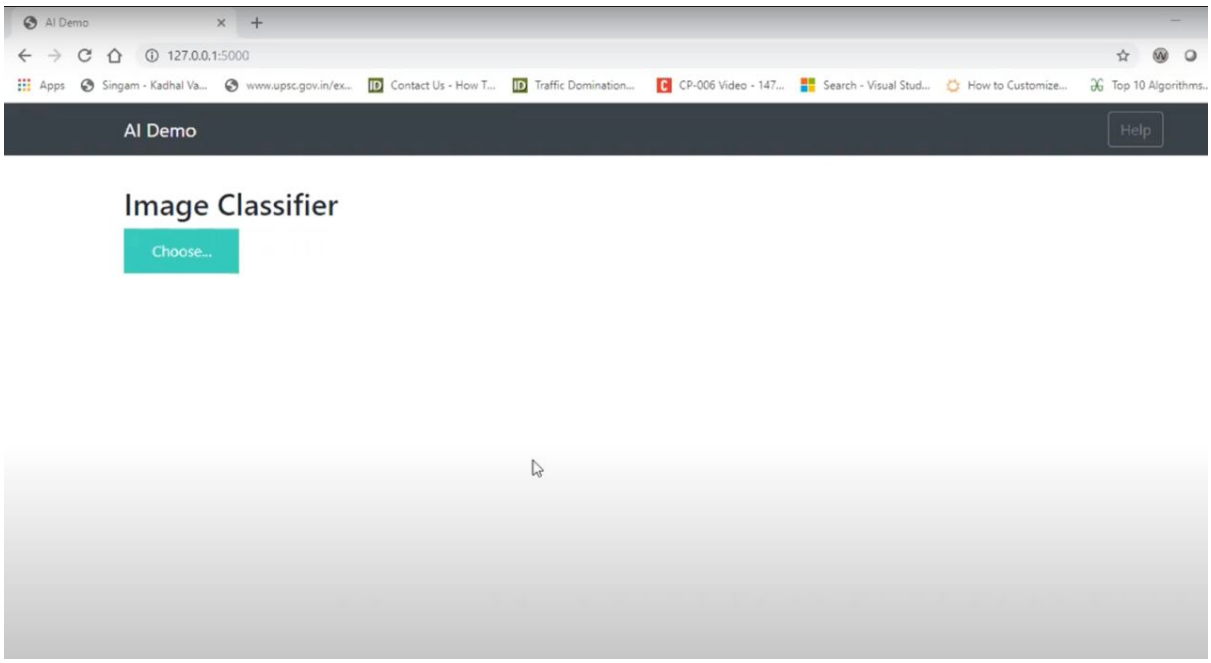
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 98, 98, 200)       5600
_________________________________________________________________
activation_1 (Activation)    (None, 98, 98, 200)       0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 49, 49, 200)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 47, 47, 100)       180100
_________________________________________________________________
activation_2 (Activation)    (None, 47, 47, 100)       0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 23, 23, 100)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 52900)             0
_________________________________________________________________
dropout_1 (Dropout)          (None, 52900)             0
_________________________________________________________________
dense_1 (Dense)              (None, 50)                2645050
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 102
=================================================================
Total params: 2,830,852
Trainable params: 2,830,852
Non-trainable params: 0
```

Restricted Mode ⊗ 0 ⚠ 0 Cell 1 of 24

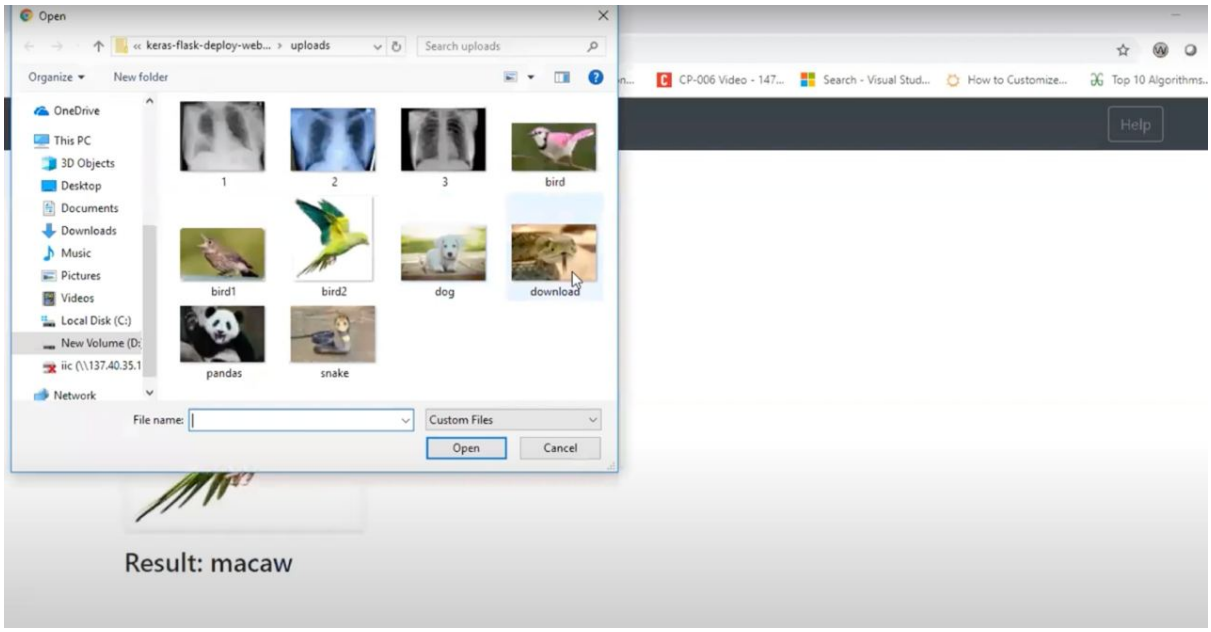
->Performance:



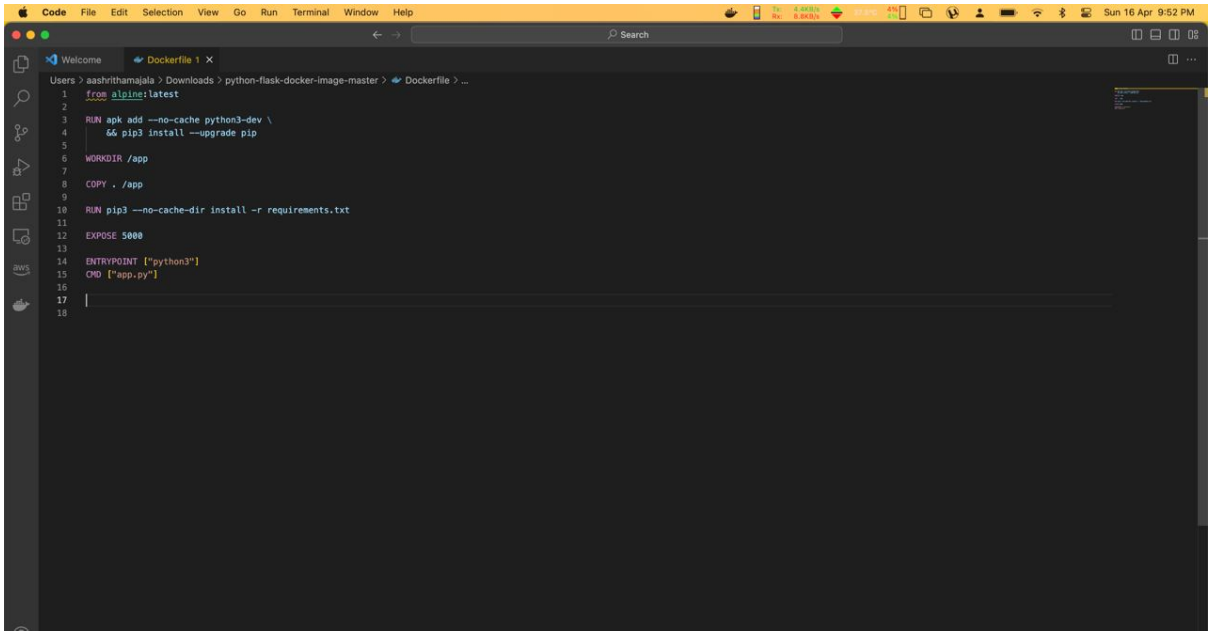
->Flask App Running on the Local Server:



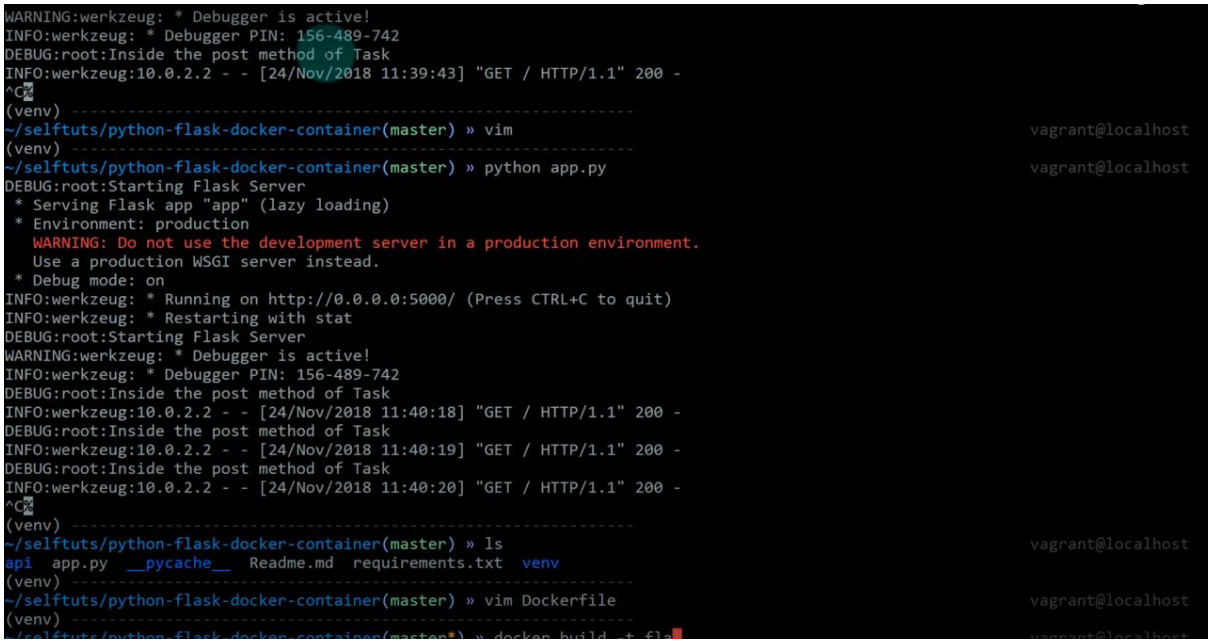

AI Demo
Help
Image Classifier
Choose...
Result: macaw


Open
« keras-flask-deploy-web... › uploads
Search uploads
Organize
New folder
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
New Volume (D:
iic (\\137.40.35.1
Network
1
2
3
bird
bird1
bird2
dog
download
pandas
snake
File name:
Custom Files
Open
Cancel
CP-006 Video - 147...
Search - Visual Stud...
How to Customize...
Top 10 Algorithms..
Help
Result: macaw

->Creating the Docker File:



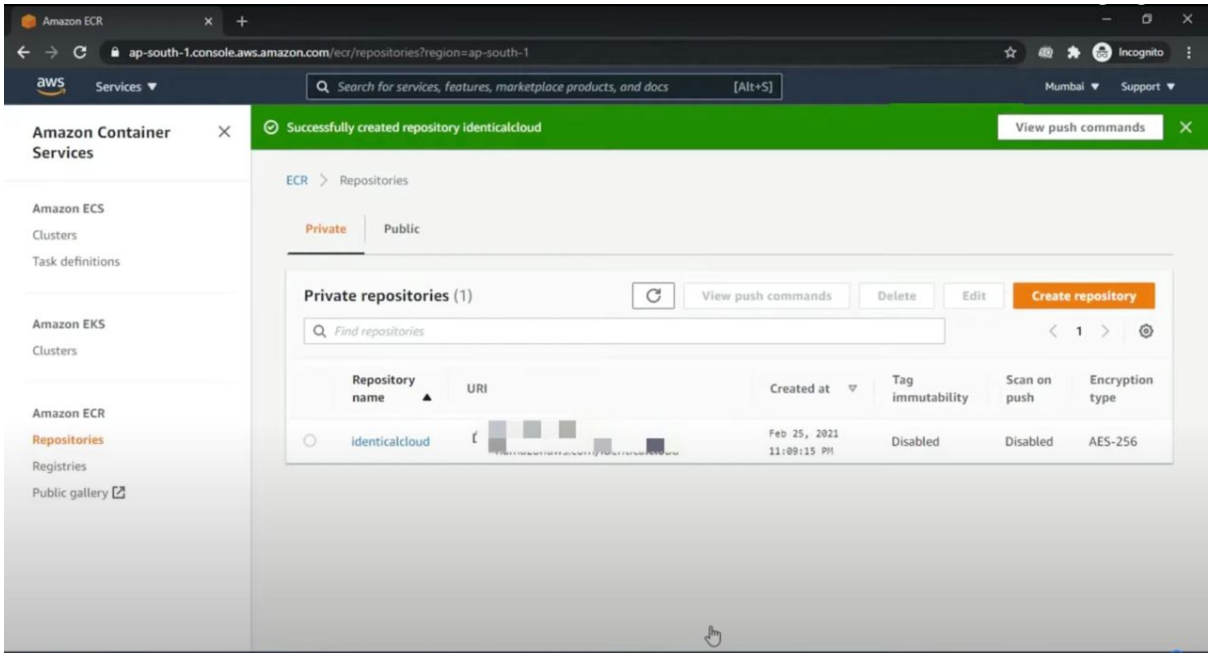
->Container Building:



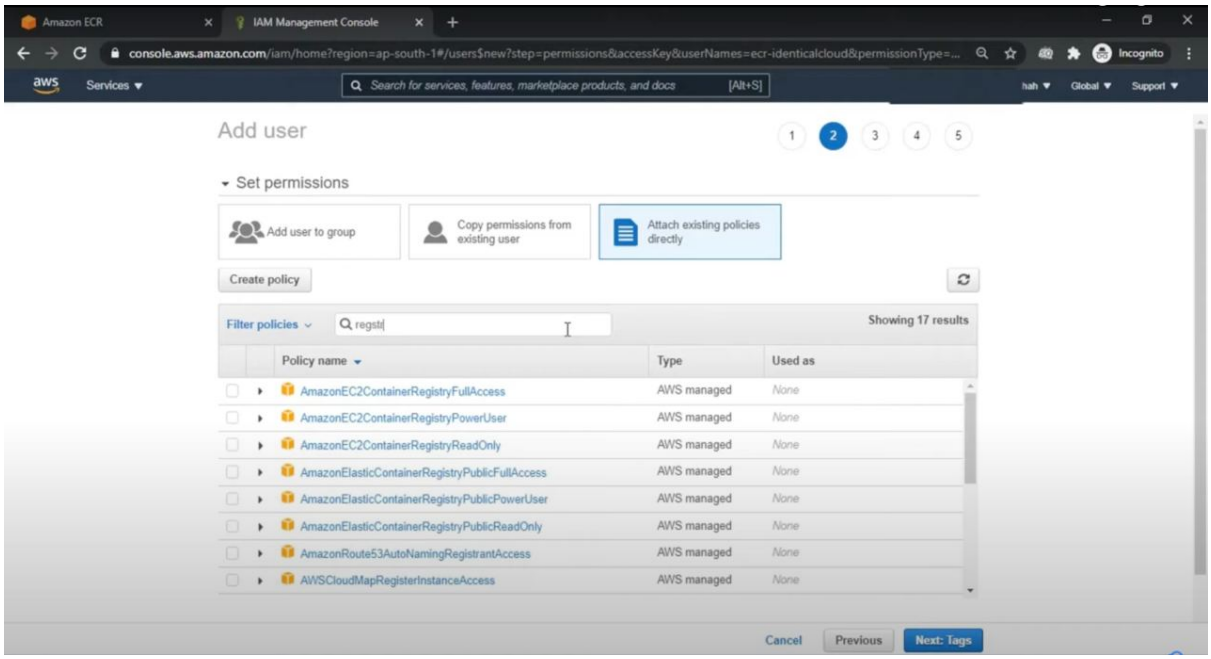
->Container Running:

```
~/selftuts/python-flask-docker-container(master*) » docker ps                                   vagrant@localhost
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
fa4bc775a1f3        flaskapp            "/bin/sh"           14 seconds ago      Up 13 seconds                           cranky_
snyder
~/selftuts/python-flask-docker-container(master*) »                                              vagrant@localhost
```

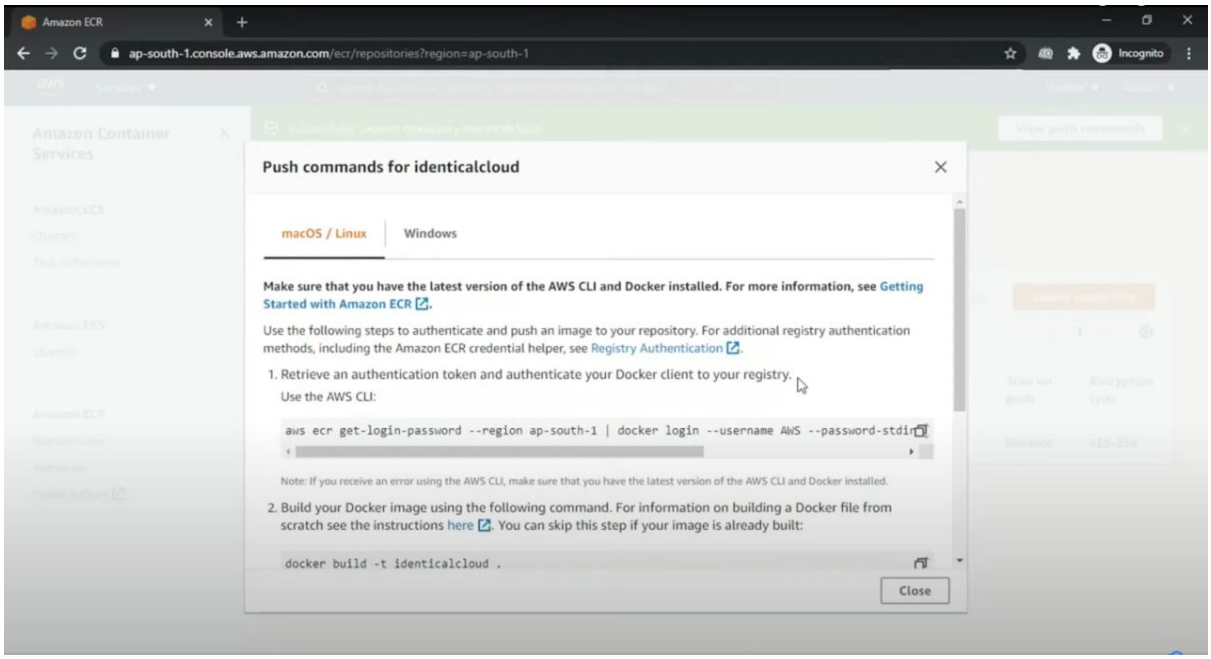
-> Creating an Elastic container registry repository



Creating iam role for pushing to repository:



Push commands for identical cloud



Pushing into the ECR repository:

```
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@ip-172-31-9-60:~# docker images
REPOSITORY              TAG         IMAGE ID       CREATED        SIZE
identicalcloud/nodejs   2           6f7124b1d12c   7 days ago     86MB
ic-nodejs               2           6f7124b1d12c   7 days ago     86MB
identicalcloud/nodejs   1           50a63ebd1c63   7 days ago     86MB
node                    10-alpine   cd8095d6b851   12 days [redacted]
root@ip-172-31-9-60:~#
```

```
2344f49dfbe3: Pushed
d179931639eb: Pushed
37cc999115d4: Pushed
9d8759356f0a: Pushed
4992425f60b2: Pushed
729a2badad91: Pushing [==================================================>]  8.371MB
ec7c69516687: Pushing [=============>                                     ]  18.35MB/68.78MB
0fcbbeeeb0d7: Pushing [==================================================>  ]  5.396MB/5.615MB
```

After pushing into the ecr repository: