Deployment of ML on AWS

STEPS

1) Create a ML Model

2) Create an API – FastAPI

3) Containerize – Docker

4) Push the Docker image to Docker hub

5) Create an AWS EC2 instance

6) Pull the Docker image from Docker hub to AWS EC2

7) Start the Docker and access the API endpoint

Create a ML Model

1) pip install scikit-learn pandas joblib

```python
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression
import joblib
import numpy as np

# Load data
iris = load_iris()
X, y = iris.data, iris.target

# Train model
model = LogisticRegression(max_iter=200)
model.fit(X, y)

# Test to see if prediction works
X_test = [5.1, 3.5, 1.4, 0.2]
X_test = np.array([X_test])  # reshape for sklearn

prediction = model.predict(X_test)
predicted_class_name = load_iris().target_names[prediction[0]]
print("Predicted class:", predicted_class_name )

# Save model
joblib.dump(model, "iris_model.pkl")
print("Model saved as iris_model.pkl")
```

Create a FastAPI and dockerize it

Create docker file:

```
FROM python:3.12 (last pushed 1 day ago)

# specify Current Working Directory
WORKDIR /code

# copy requirements.txt from local to container
COPY ./requirements.txt /code/requirements.txt

# install all requirements in the container
RUN pip install --no-cache-dir --default-timeout=100 -r /code/requirements.txt

# copy dir app from local to container
COPY ./app /code/app

# Expose port 8000 so other app can communicate with container
EXPOSE 8000

# Run following command to start uvicorn server
CMD ["uvicorn", "app.server_app:app", "--host", "0.0.0.0", "--port", "8000"]
```

Create a FastAPI and dockerize it

1) Create fastAPI server

See the code **server_app.py**

```python
from fastapi import FastAPI
import joblib
import numpy as np
from sklearn.datasets import load_iris

model = joblib.load('app/iris_model.pkl')

app = FastAPI()

@app.get('/')
def read_root():
    return {'message': 'Welcome to Iris model API'}

@app.post('/predict')
def predict(data: dict):
    features = np.array(data['features']).reshape(1, -1)
    prediction = model.predict(features)
    class_name = load_iris().target_names[prediction[0]]
    return {'predicted_class': class_name}
```

2) Create requirements.txt file

See the file **requirements.txt**

```
fastapi==0.116.1
uvicorn==0.35.0
numpy==2.0.2
scikit-learn==1.6.1
```

3) Create Docker file

See the **Docker** file

step a: Start docker engine

Start the docker engine by simply running the **Docker Desktop**: Start -> Docker Desktop

step b: Build docker image

Run following command in **same directory** which contains Dockerfile:

> **docker build -t image_ml_model .**

This would pull the image from docker hub. This could take **4 - 10 minutes**.

step c: Run container

> **docker run --name container_ml_model -p 8000:8000 image_ml_model**

NOTE: If you make a mistake, then delete the container_ml_model

> **docker rm -f <container_id_name>**

and redo above steps

Create a FastAPI and dockerize it

4) Goto URL http://127.0.0.1:8000/docs
Or
http://127.0.0.1:8000/
You should see a simple Welcome message

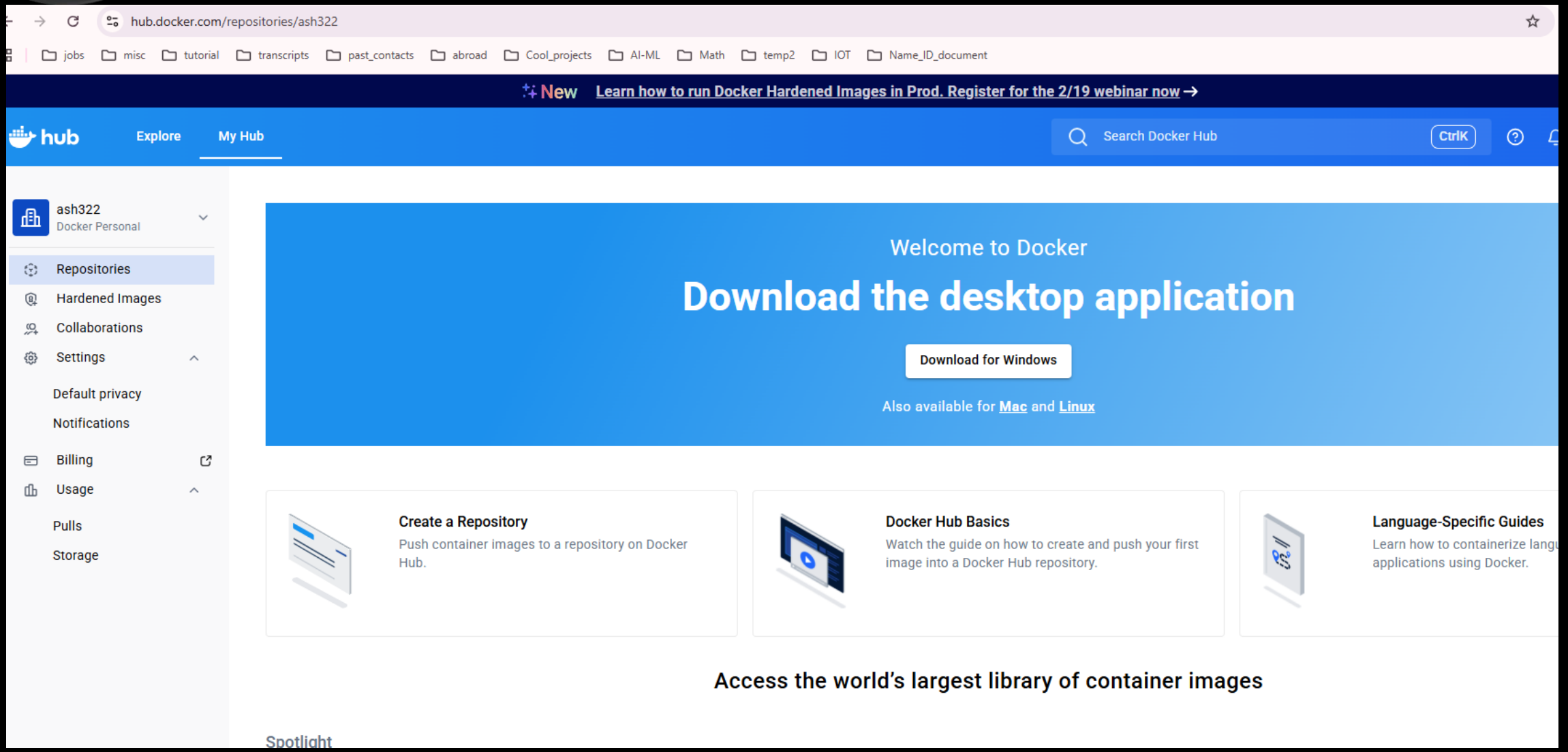
5) Access the API endpoint via python program
Run the file **client_to_access_API_endpoint.py** in a dedicated terminal
Or
Run the jupyter notebook **client_to_access_API_endpoint.ipynb**

This docker is running locally

Push the Docker Image to Docker Hub

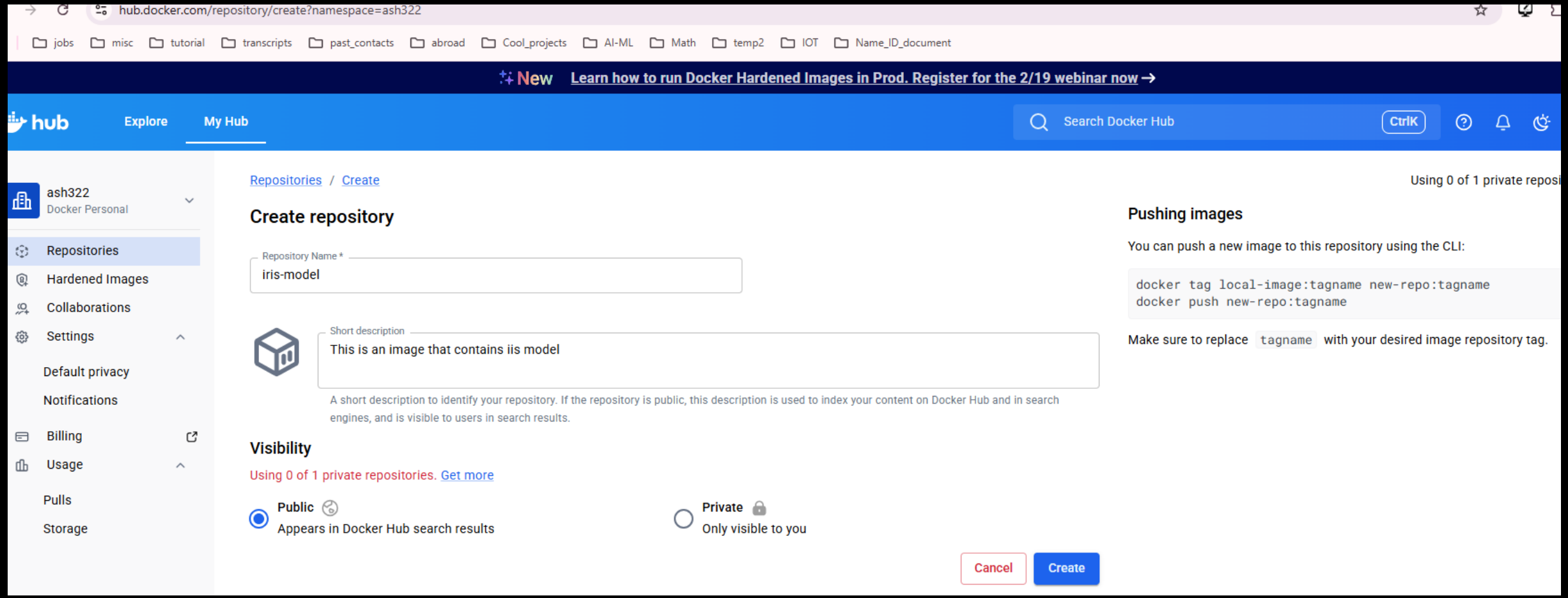
Push the Image to Docker Hub

Goto docker hub and create an account. Also note down user name



Push the Image to Docker Hub

Create a repository: **image_ml_model**



Push the Image to Docker Hub

1) Now let's look at the docker images on local machine:

> **docker image ls**

2) Now we tag the image:
> **docker tag image_ml_model ash322/image_ml_model**

Here **ash322** is my username

3) Now login:
> **docker login**

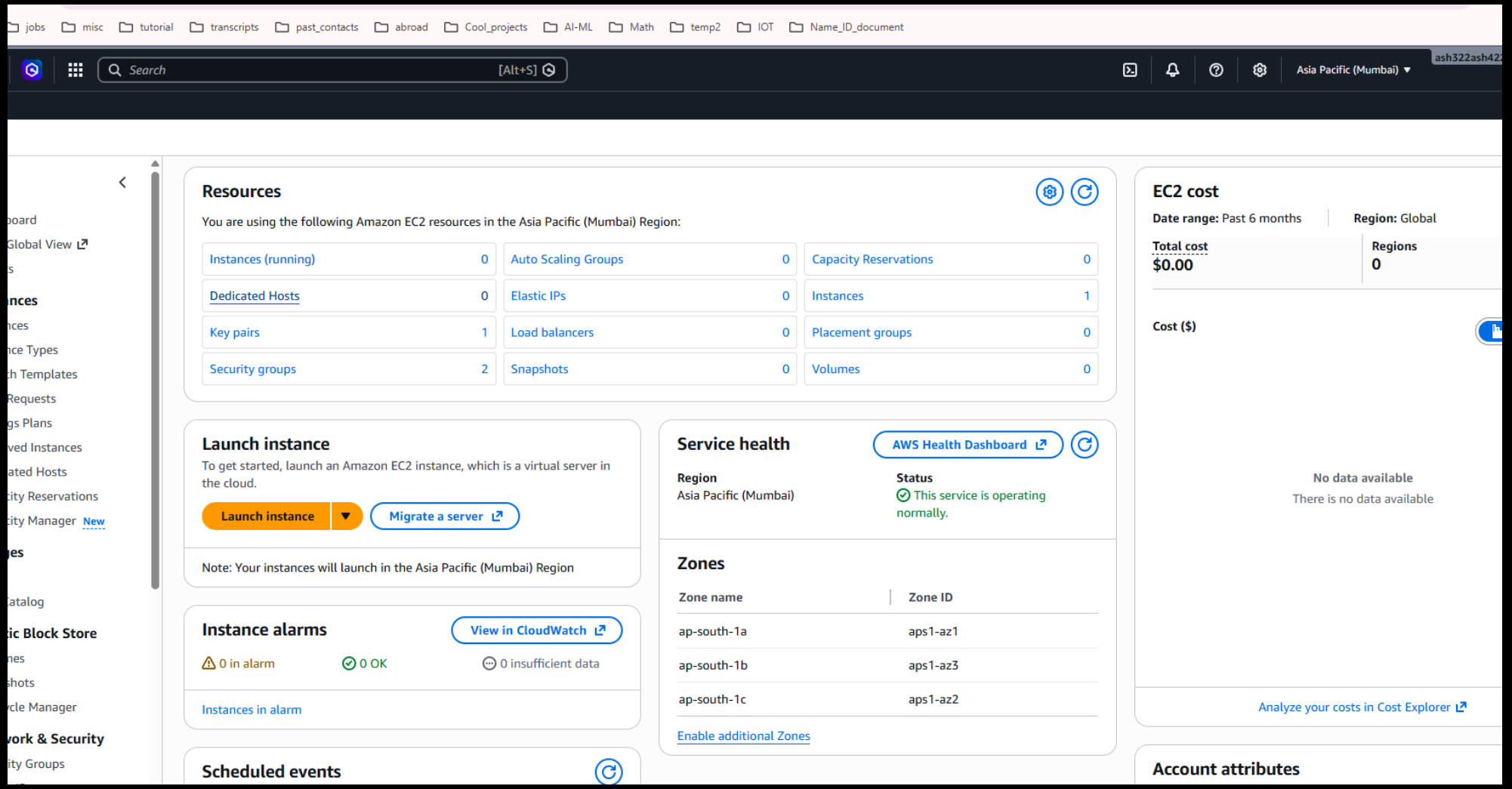
(Follow instructions)

3) Now push the image to docker hub (This could take few minutes):
> **docker push ash322/image_ml_model**

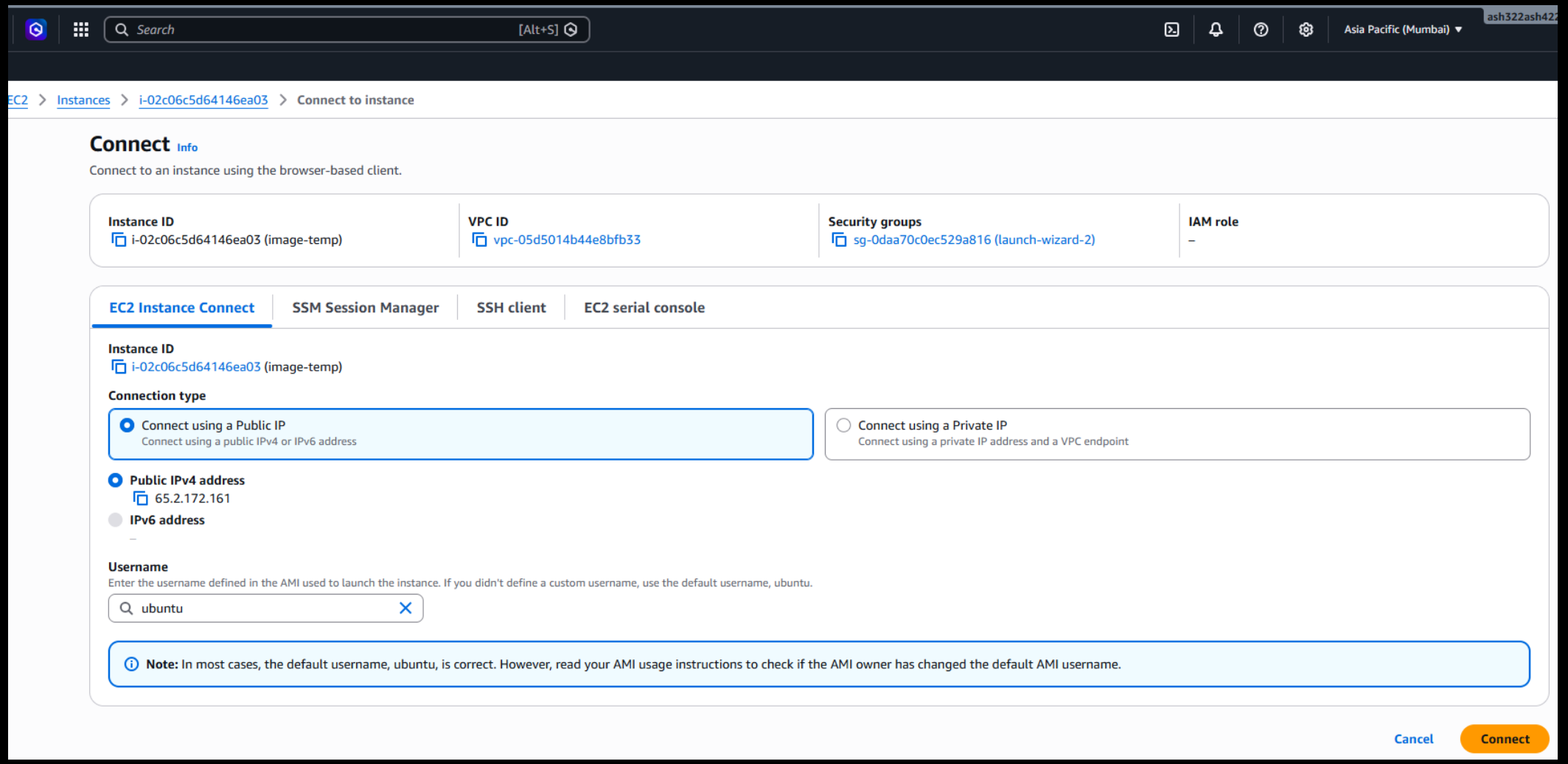
NOW CREATE AN AWS EC2 INSTANCE AND PULL THE DOCKER IMAGE

Here we **launch the EC2 instance**.

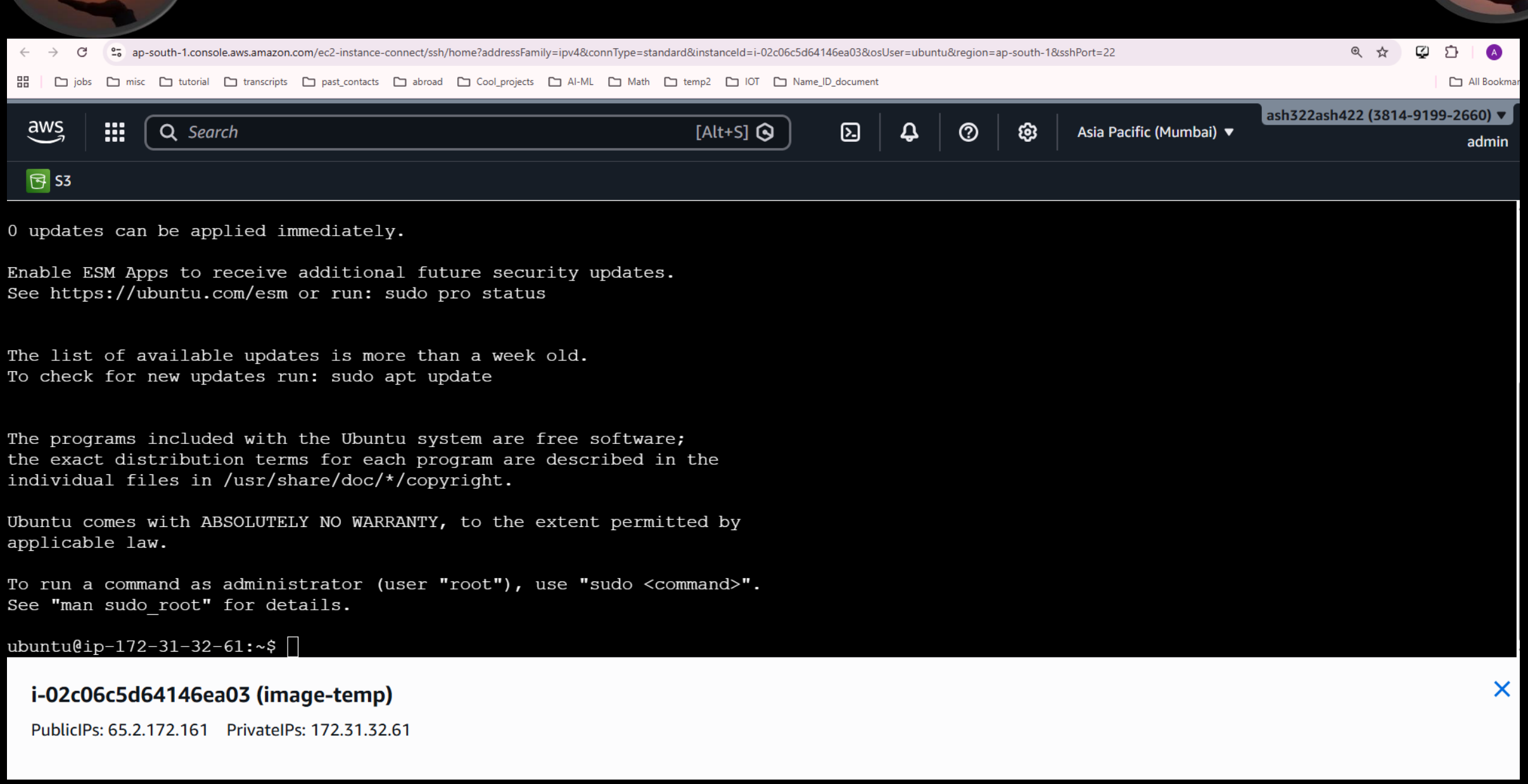
- Name = iris-model-instance
- OS image=Ubuntu
- Do not need to create key-pair. Proceed without key-pair

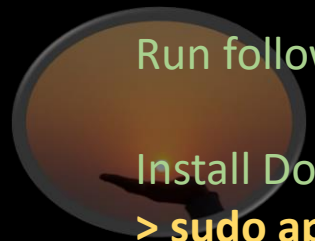


Now we connect to our EC2 instance: **Connect using a Public IP**



You should see following EC2 console





Run following commands on **EC2 console:**

Install Docker on EC2

> sudo apt update

> sudo apt install docker.io -y

> sudo systemctl start docker

Check if docker is running:

> ps –aux | grep dock

Login to docker:

> docker login

Go to url specified and paste the code

(Now we give permission)

> sudo usermod -aG docker $USER

> newgrp docker

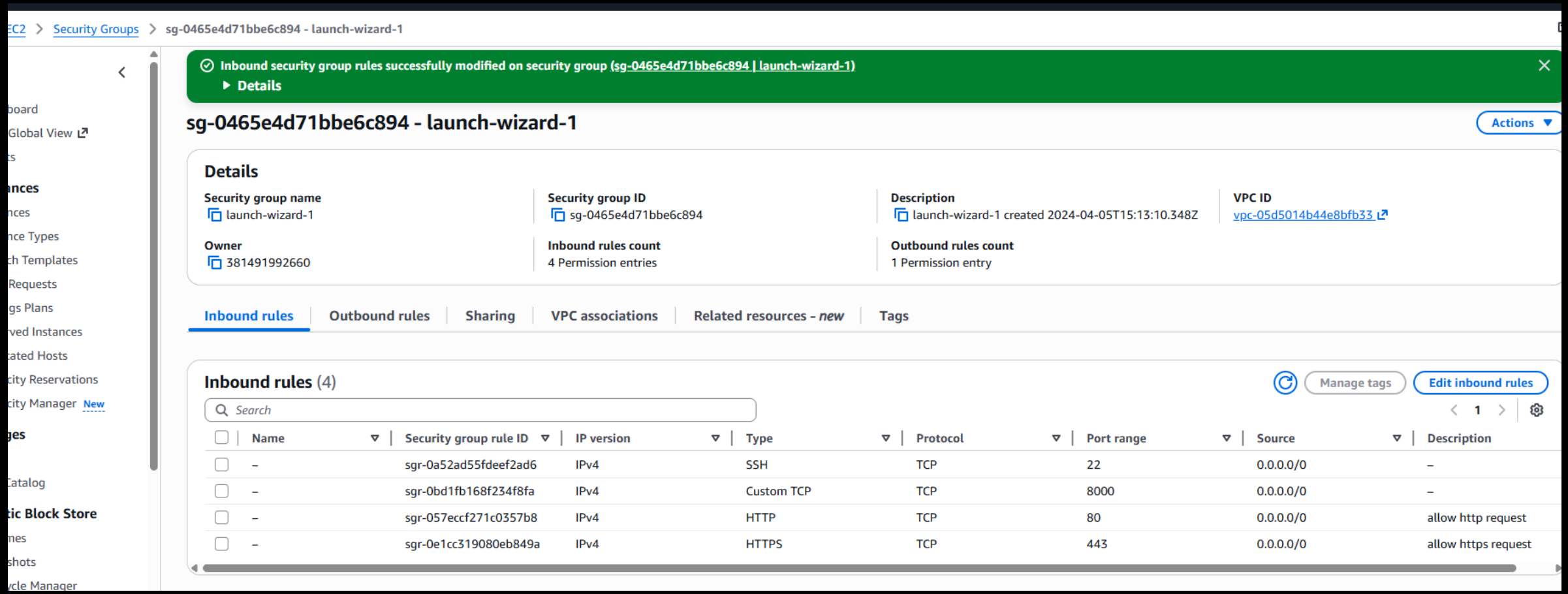
Pull Image on Ubuntu OS:

> docker pull ash322/image_ml_model

Run Container

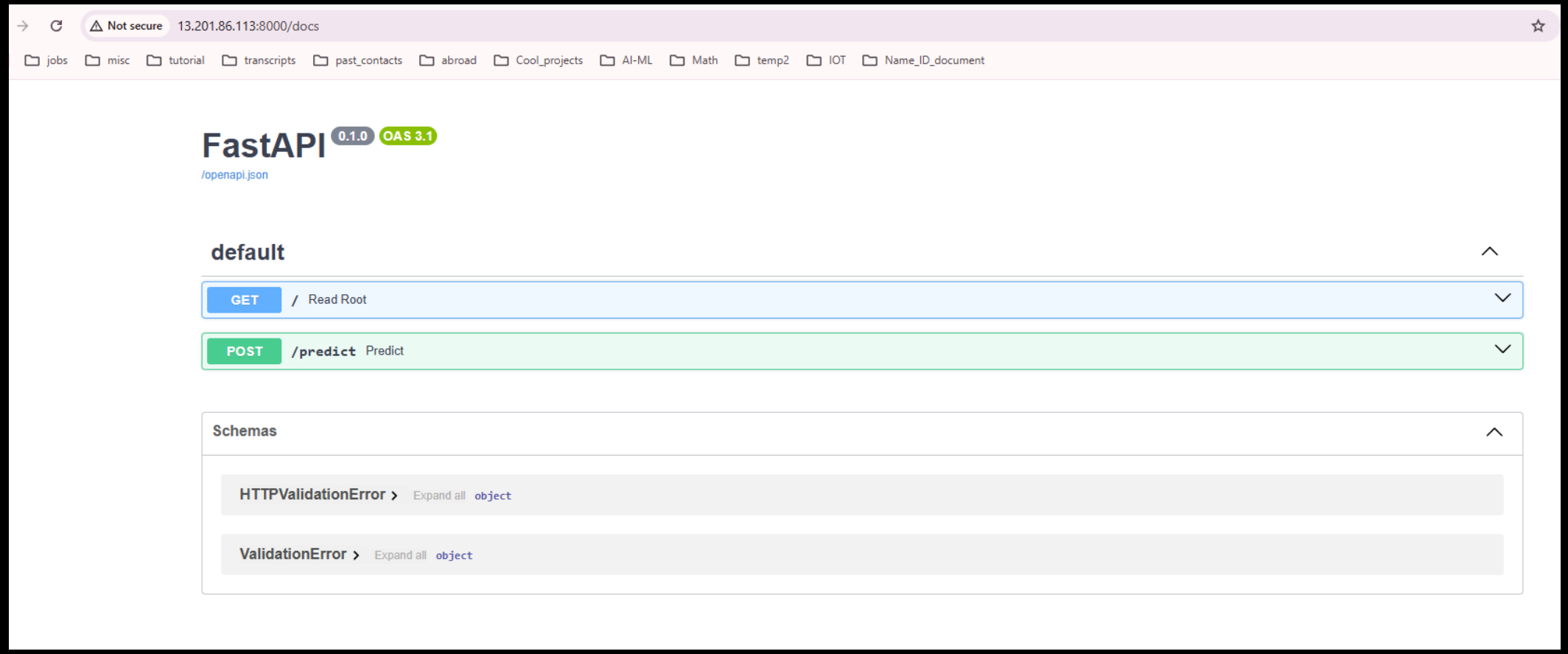
> docker run -d -p 8000:8000 ash322/image_ml_model

Open Security Group:
Make sure you have following
Custom TCP → Port 8000 → Anywhere



Access Model:

http://EC2-PUBLIC-IP:8000/docs



Now run following:

```
import json
import requests
```

```
data = [
    [5.4, 3.4, 1.7, 0.2],
    [5.1, 3.7, 1.5, 0.4],
    [5, 2.3, 3.3, 1],
    [6.7, 3.3, 5.7, 2.1],
]

# url = 'http://127.0.0.1:8000/predict/'
url = 'http://13.201.86.113:8000/predict/' # from AWS EC2


predictions = []
for record in data:
    payload = {'features': record}
    payload = json.dumps(payload)
    response = requests.post(url, data=payload)
    predictions.append(response.json()['predicted_class'])

print("The prediction received from server:\n", predictions)
```

```
The prediction received from server:
 ['setosa', 'setosa', 'versicolor', 'virginica']
```