

Week 10 Homework 1: Project: Machine Learning on Kubernetes

Student ID 19900

Student Name: MD ROKIBUL HASAN

Step 1: Set up a functional Kubernetes cluster:

Open GKE terminal

□□ Now, lets create a kubernetes cluster with three nodes

gcloud container clusters create kubia1 --num-nodes=3 --zone=us-west1-b --machine-type=n1-standard-1

```
kubeconfig entry generated for kubia1.
NAME: kubia1
LOCATION: us-west1-b
MASTER_VERSION: 1.30.5-gke.1443001
MASTER_IP: 34.168.124.78
MACHINE_TYPE: n1-standard-1
NODE_VERSION: 1.30.5-gke.1443001
NUM_NODES: 3
STATUS: RUNNING
mhasan55157@cloudshell:~ (summer-foundry-441517-q7)$
```

□□ Double check if nodes are correctly created

kubectl get nodes

You should see three nodes being created:

```
mhasan55157@cloudshell:~ (summer-foundry-441517-q7)$ kubectl get nodes
NAME                                STATUS    ROLES    AGE     VERSION
gke-kubia1-default-pool-43c66686-8pt7 Ready     <none>    5m19s   v1.30.5-gke.1443001
gke-kubia1-default-pool-43c66686-xrvj Ready     <none>    5m19s   v1.30.5-gke.1443001
gke-kubia1-default-pool-43c66686-zhph Ready     <none>    5m18s   v1.30.5-gke.1443001
mhasan55157@cloudshell:~ (summer-foundry-441517-q7)$
```

Step 2: Machine Learning Deployment using Docker

Implement the following steps in your local machine

□□ Create a local directory

□□ Put the following files into your directory

Source: <https://github.com/HasnaeTalibi/Machine-Learning-Deployment-using-Docker/tree/main>

□□ Build a docker image

1. Turn on your docker app in your local computer
2. Use the following command to build an image
sudo docker build -t my-flask-app .

```
alien@alien-Virtual-Machine:~/files$ sudo docker build -t my-flask-app .
[+] Building 10.2s (9/9) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 499B                             0.0s
=> [internal] load metadata for docker.io/continuumio/anaconda3:4.4.0 0.7s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                     0.0s
=> [internal] load build context                                0.0s
=> => transferring context: 626B                                   0.0s
=> CACHED [1/4] FROM docker.io/continuumio/anaconda3:4.4.0@sha256:c6bb52 0.0s
=> [2/4] COPY . /usr/ML/app                                     0.0s
=> [3/4] WORKDIR /usr/ML/app                                    0.0s
=> [4/4] RUN pip install --trusted-host pypi.org --trusted-host pypi.pyt 8.8s
=> exporting to image                                           0.4s
=> => exporting layers                                           0.4s
=> => writing image sha256:e41564fc25c9ac6958973e1034ff70de22d82e82e207c 0.0s
=> => naming to docker.io/library/my-flask-app                  0.0s

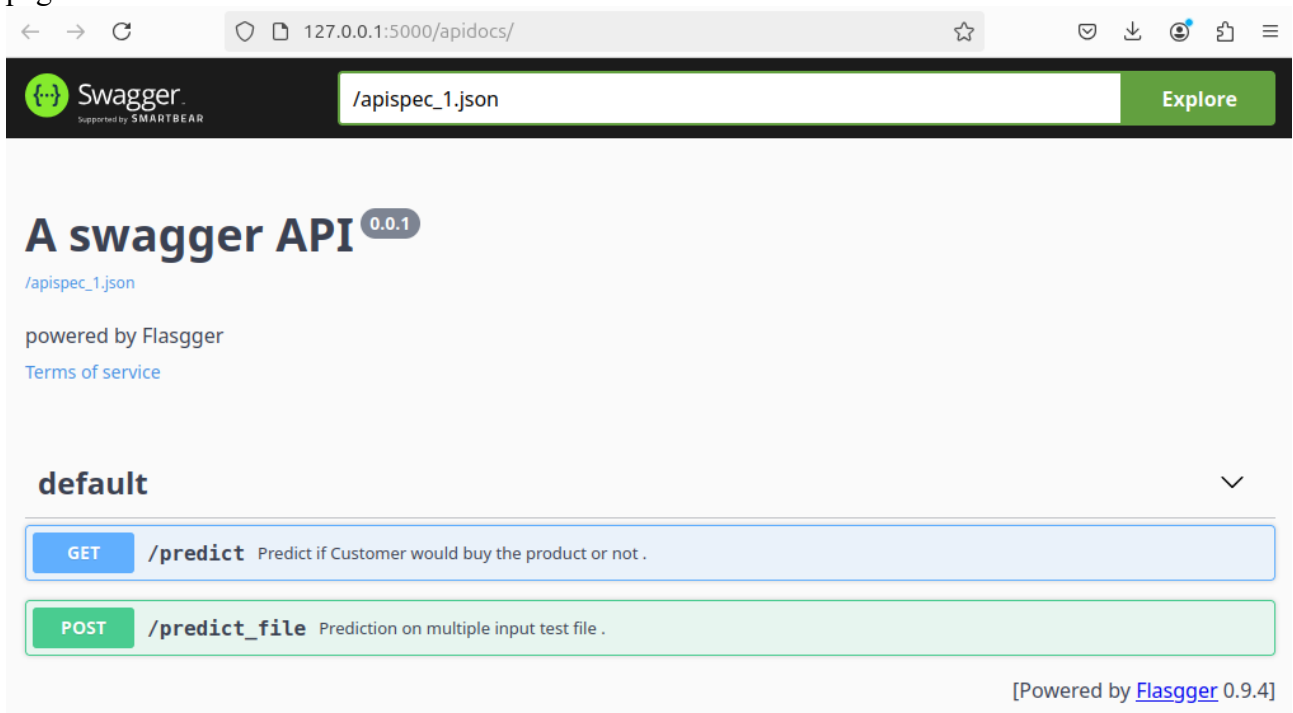
1 warning found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended
behavior related to OS signals (line 17)
alien@alien-Virtual-Machine:~/files$
```

□□ Initiate the container to run our ML app

docker container run -p 5000:5000 my-flask-app

```
root@alien-Virtual-Machine:/myfolder/files# docker container run -p 5000:5000 my-flask-app
* Serving Flask app "flask_api" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
/opt/conda/lib/python3.6/site-packages/sklearn/base.py:318: UserWarning: Trying to unpickle estimator
LogisticRegression from version 0.23.2 when using version 0.22.1. This might lead to breaking code o
r invalid results. Use at your own risk.
UserWarning)
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
* Restarting with stat
/opt/conda/lib/python3.6/site-packages/sklearn/base.py:318: UserWarning: Trying to unpickle estimator
LogisticRegression from version 0.23.2 when using version 0.22.1. This might lead to breaking code o
r invalid results. Use at your own risk.
UserWarning)
* Debugger is active!
```

To access to the app, we simply have to go to <http://127.0.0.1:5000/apidocs> to load the Swagger UI page



□□ Push the image to your docker hub

- Login your docker hub
 - docker login

```
root@alien-Virtual-Machine:/myfolder/files# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
root@alien-Virtual-Machine:/myfolder/files#
```

- Tag the image "my-flask-app" as latest
 - docker tag my-flask-app rokibul2024/myflaskapp

- Push the image to Docker Hub
 - `docker push rokibul2024/myflaskapp:latest`

```
root@alien-Virtual-Machine:/myfolder/files# docker push rokibul2024/myflaskapp:latest
The push refers to repository [docker.io/rokibul2024/myflaskapp]
2babf274907f: Layer already exists
5f70bf18a086: Layer already exists
4d7debba76e7: Layer already exists
ca173dc10e31: Layer already exists
54e10c08a841: Layer already exists
1f09b1beaa90: Layer already exists
9e63c5bce458: Layer already exists
latest: digest: sha256:1affa4bd6da70de6826b0a0e428a17d4e036ea7f64bce0e22ae8d7fb36d82209 size: 1791
root@alien-Virtual-Machine:/myfolder/files#
```

rokibul2024

Search by repository name

All Content

Create repository

rokibul2024 / myflaskapp

☆ 0

↓ 20

Public

Scout inactive

Contains: Image

Last pushed: 5 minutes ago

□□ Stop the container

The last step left after running the application is to stop the running container. This can be done using the `docker stop` or `kill` command on the running container. We can see the list of running containers using the `docker ps` command and can select the running container ID to stop it.

```
root@alien-Virtual-Machine:/myfolder/files# docker ps
```

CONTAINER ID	IMAGE	COMMAND NAMES	CREATED	STATUS	PORTS
4a118a0c503d	my-flask-app	"/usr/bin/tini -- /b..."	5 minutes ago	Up 5 minutes	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp

```
root@alien-Virtual-Machine:/myfolder/files#
```

`docker kill <Container_ID>`

```
root@alien-Virtual-Machine:/myfolder/files# docker kill 4a118a0c503d
4a118a0c503d
root@alien-Virtual-Machine:/myfolder/files#
```

Step 3: Deploy your ML app to GKE

Use the GKE we have created in Step 1

□□ Create a `appdeployment.yaml` with the following contents.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ml-app-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ml-app
  template:
    metadata:
      labels:
        app: ml-app
    spec:
      containers:
```

- name: my-flask-app
image: rokibul2024/myflaskapp
ports:
- containerPort: 5000

□□ after creating apply the deployment file

kubectl apply -f appdeployment.yaml

```
mhasan55157@cloudshell:~ (summer-foundry-441517-q7)$ vi appdeployment.yaml
mhasan55157@cloudshell:~ (summer-foundry-441517-q7)$ kubectl apply -f appdeployment.yaml
deployment.apps/ml-app-deployment created
```

□□ Wait for couple minutes and list all the pods created

kubectl get pods

```
mhasan55157@cloudshell:~ (summer-foundry-441517-q7)$ vi appdeployment.yaml
mhasan55157@cloudshell:~ (summer-foundry-441517-q7)$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
ml-app-deployment-5d8bc5876d-j4q2w  1/1     Running   0           80s
mhasan55157@cloudshell:~ (summer-foundry-441517-q7)$
```

□□ Create a service.yaml

apiVersion: v1
kind: Service
metadata:
name: ml-app-service
spec:
selector:
app: ml-app
ports:
- protocol: TCP
port: 80
targetPort: 5000
type: LoadBalancer

□ after creating apply the services

kubectl apply -f service.yaml

```
mhasan55157@cloudshell:~ (summer-foundry-441517-q7)$ kubectl apply -f services.yaml
service/ml-app-service created
mhasan55157@cloudshell:~ (summer-foundry-441517-q7)$
```

□□ Get service external ip

kubectl get svc

```
mhasan55157@cloudshell:~ (summer-foundry-441517-q7)$ kubectl get svc
NAME            TYPE           CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes      ClusterIP      34.118.224.1    <none>           443/TCP          36m
ml-app-service  LoadBalancer  34.118.239.59   35.185.243.85   80:32596/TCP     34s
```

□□ Access using browser:

external-ip/apidocs

Swagger
Supported by SMARTBEAR

/apispec_1.json

Explore

A swagger API 0.0.1

/apispec_1.json

powered by [Flasgger](#)

[Terms of service](#)

default

GET /predict Predict if Customer would buy the product or not .

POST /predict_file Prediction on multiple input test file .

[Powered by [Flasgger](#) 0.9.4]

THE END