Deploy app using kubenetes and apache

High level workflow:

1.) Create app

2.) Create Dockefile

3.) Create deployment.yaml file

4.) Install minikube, apache and docker

5.) configure minikube with required resources

6.) Apply deployment.yaml files

7.) Get url of service and configure with apache server

Commands

1. Dockerfile :- $nano Dockerfile ()

In the file mention the:

* Platform version
* Update system packages(get update and shit)
* Install independent libraries (torch detectron etc)
* Install required tools(pytesseratc, popplar etc)
* Application setup in container(create dir(separate for files if large files cannot be uploaded to Hub), add all files to directory, and set it as working dir)
* Install req.txt
* If large files on GDrive, RUN the command to download it
* RUN flaask/django app

1. Deployment.yaml :- $nano deployment.yaml

One for service and one for app. Both can be in one yaml file too. Here in the app part mentioin the docker image name.

The port mentioned in the app part is the flask/django port.

In the selector part of both service and app, the “app:” shjoul;d be in “metadata:” of app part, the “app:” should be same

The “target port:” service part should be same as flask port. The port of service can be set by us.

1. Install Docker: (<https://docs.docker.com/engine/install/ubuntu/)>

sudo apt-get update   
sudo apt-get install ca-certificates

curl gnupg lsb-release

sudo mkdir -p /etc/apt/keyrings  
 curl -fsSL <https://download.docker.com/linux/ubuntu/gpg> | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

echo \  
 "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] <https://download.docker.com/linux/ubuntu> \  
 $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

Index of linux/ubuntu/

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin

#minikube installation (<https://phoenixnap.com/kb/install-minikube-on-ubuntu)>

sudo apt-get update -y

sudo apt-get install curl

sudo apt-get install apt-transport-https

wget <https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64>

sudo cp minikube-linux-amd64 /usr/local/bin/minikube

sudo chmod 755 /usr/local/bin/minikube

minikube version

#install kubectl

curl -LO [https://storage.googleapis.com/kubernetes-release/release/`curl](https://storage.googleapis.com/kubernetes-release/release/%60curl) -s [https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl](https://storage.googleapis.com/kubernetes-release/release/stable.txt%60/bin/linux/amd64/kubectl)

chmod +x ./kubectl  
  
sudo mv ./kubectl /usr/local/bin/kubectl  
  
kubectl version -o json

install apache (<https://www.syncfusion.com/blogs/post/hosting-multiple-asp-net-core-apps-in-ubuntu-linux-server-using-apache.aspx)>

* Let’s install the Apache server in the Linux machine using the following
* sudo apt install apache2

* After installing the Apache Server, install the following required modules

* sudo a2enmod rewrite
* sudo a2enmod proxy
* sudo a2enmod proxy\_http
* sudo a2enmod headers
* sudo a2enmod ssl

* Now, restart the Apache server using the following command
* sudo service apache2 restart

#setup minikube

minikube start –cpus=4 –memory=12000 –driver=docker  
nano deployment.yaml  
minikube image build -t ocr-app .  
minikube image ls  
nano deployment.yaml  
kubectl apply -f deployment.yaml  
kubectl get po  
kubectl get svc  
minikube service ocr-api-service –url

Setup Apache Virtual Host :

* Go to below path from terminal

cd /etc/apache2/sites-available

* Create new conf file

sudo nano ocr-api.conf

* After Open in new file then type below command (remember to remove <>)

<VirtualHost \*:6996> #(You can give any port but that port need to add in port.conf file)

ProxyPreserveHost On

ProxyPass / <http://127.0.0.1:5000/> (Once minikube service is set up then copy the url and paste it here)

ProxyPassReverse / [http://127.0.0.1:5000](http://127.0.0.1:5000/)/ (Once minikube service is set up then copy the url and paste it here)

ErrorLog /var/log/apache2/ocr-api-error.log

CustomLog /var/log/apache2/ocr-api-access.log common

</VirtualHost>

Test the configuration. If everything passes, the result should be OK.

sudo apachectl configtest

(Open terminal and go to below path )

cd /etc/apache2/

Open ports.conf file

sudo nano ports.conf

Add new port in below file

Listen 80

Listen 6996

Listen 6767 etc...

#then do this

sudo a2ensite stability.conf -> for site enable

sudo apachectl configtest

systemctl reload apache2 -> for restart