Deploying a Multi-Tier Web Application on Kubernetes

Prerequisites

Before starting, ensure you have the following installed:

- Kubernetes Cluster (Minikube/Kubeadm)
- Docker
- kubectl CLI

Folder Structure & File Usages

k8s-project/

```
├── mysql/              # MySQL Database Configuration
│   ├── mysql-pv.yaml       # Persistent Volume for MySQL Data Storage
│   ├── mysql-secret.yaml    # Stores MySQL Root Password Securely
│   ├── mysql-deployment.yaml # Deploys MySQL Database as a StatefulSet
├── flask/              # Flask Backend Configuration
│   ├── app.py              # Flask API Code to Handle Requests
│   ├── Dockerfile          # Flask App Containerization Instructions
│   ├── requirements.txt     # Dependencies for Flask
│   ├── flask-deployment.yaml # Deploys Flask Application
│   ├── flask-service.yaml    # Exposes Flask App as a Cluster Service
├── nginx/              # Nginx Configuration
│   ├── nginx-configmap.yaml  # Reverse Proxy Configuration for Flask
│   ├── nginx-deployment.yaml # Deploys Nginx
│   ├── nginx-service.yaml    # Exposes Nginx via NodePort
```

Step-by-Step Deployment Guide

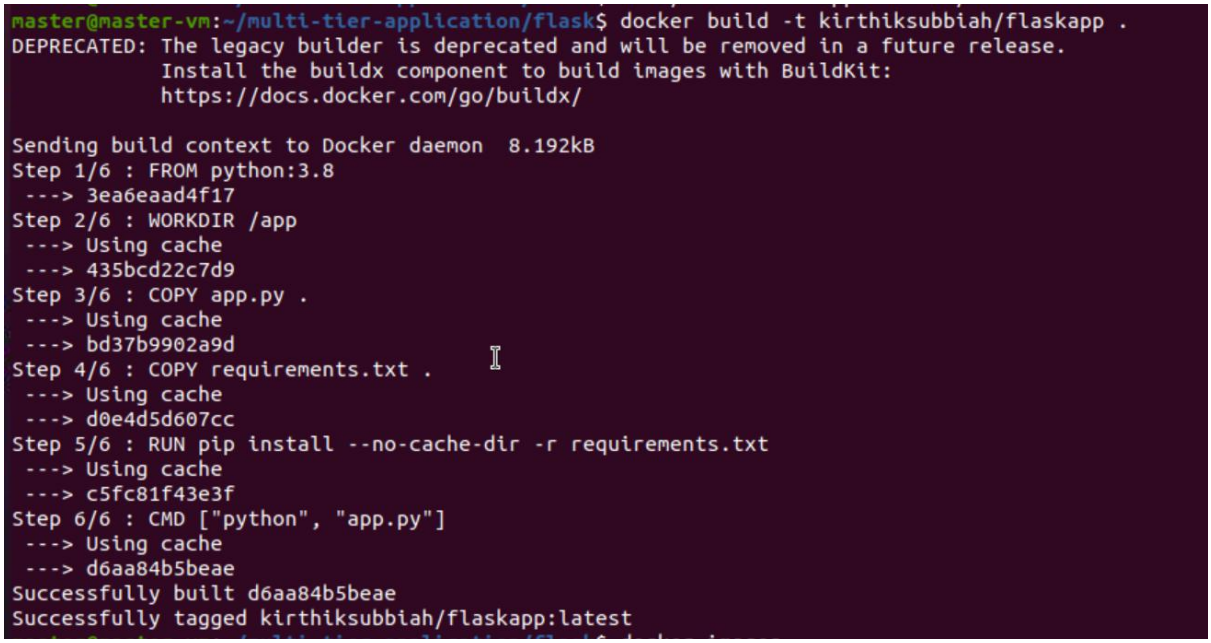
Step 1: Build and Push Docker Image

1.1 Navigate to the Flask directory

cd flask

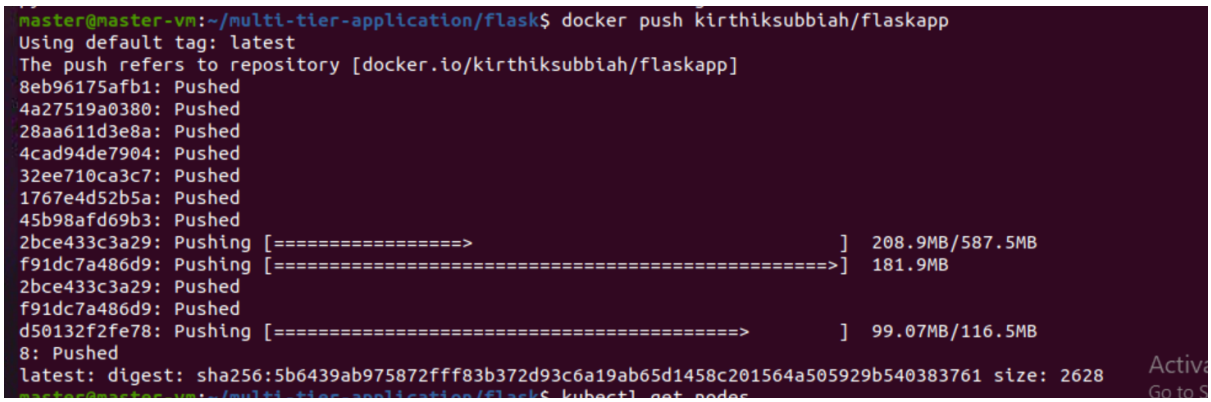
1.2 Build the Docker image

docker build -t dockerhub_username/flaskapp .



1.3 Push the image to Docker Hub

docker push dockerhub_username/flaskapp



Step 2: Apply Kubernetes Configurations

2.1 Deploy Flask application

kubectl apply -f flask-deployment.yaml

kubectl apply -f flask-service.yaml

```
master@master-vm:~/multi-tier-application/flask$ kubectl get nodes
NAME         STATUS   ROLES           AGE     VERSION
master-vm    Ready    control-plane   2d23h   v1.28.15
worker1-vm   Ready    <none>          2d23h   v1.28.15
worker2-vm   Ready    <none>          2d23h   v1.28.15
master@master-vm:~/multi-tier-application/flask$ kubectl apply -f flask-deployment.yaml
deployment.apps/flask-app created
master@master-vm:~/multi-tier-application/flask$ kubectl apply -f flask-service.yaml
service/flask-service created
master@master-vm:~/multi-tier-application/flask$ cd ../mysql
master@master-vm:~/multi-tier-application/mysql$ kubectl apply -f mysql-deployment.yaml
deployment.apps/mysql created
service/mysql created
master@master-vm:~/multi-tier-application/mysql$ kubectl apply -f mysql-pv.yaml
persistentvolume/mysql-pv created
persistentvolumeclaim/mysql-pvc created
```

2.2 Deploy MySQL database

cd ../mysql

kubectl apply -f mysql-deployment.yaml

kubectl apply -f mysql-pv.yaml

kubectl apply -f mysql-secret.yaml

```
master@master-vm:~/multi-tier-application/mysql$ kubectl apply -f mysql-deployment.yaml
deployment.apps/mysql created
service/mysql created
master@master-vm:~/multi-tier-application/mysql$ kubectl apply -f mysql-secret.yaml
secret/mysql-secret created
master@master-vm:~/multi-tier-application/mysql$ kubectl apply -f mysql-pv.yaml
persistentvolume/mysql-pv unchanged
persistentvolumeclaim/mysql-pvc unchanged
```

2.3 Deploy Nginx

cd ../nginx

kubectl apply -f nginx-configmap.yaml

kubectl apply -f nginx-deployment.yaml

kubectl apply -f nginx-service.yaml

```
master@master-vm:~/multi-tier-application/nginx$ nano nginx-configmap.yaml
master@master-vm:~/multi-tier-application/nginx$ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx unchanged
master@master-vm:~/multi-tier-application/nginx$ kubectl apply -f nginx-service.yaml
service/nginx-service created
master@master-vm:~/multi-tier-application/nginx$ kubectl apply -f nginx-configmap.yaml
configmap/nginx-config created
master@master-vm:~/multi-tier-application/nginx$
master@master-vm:~/multi-tier-application/nginx$ cd ..
```

Step 5: Initialize MySQL Database

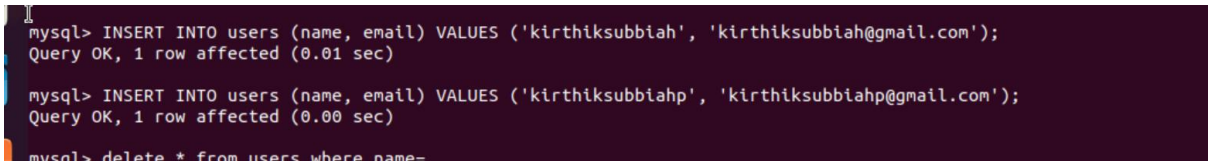
5.1 Access MySQL inside the Pod

kubectl exec -it mysql-0 -- mysql -u root -p

```
master@master-vm:~/multi-tier-application/nginx$ kubectl exec -it mysql-66d468f74c-b4wk9 -- mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.44 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

5.2 Create and populate the database

```
CREATE DATABASE mydb;

USE mydb;

CREATE TABLE users (
   id INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(100),
   email VARCHAR(100)
);

INSERT INTO users (name, email) VALUES ('Alice', 'alice@example.com');

INSERT INTO users (name, email) VALUES ('Bob', 'bob@example.com');

SELECT * FROM users;

GRANT ALL PRIVILEGES ON mydb.* TO 'user'@'%';
```




```
FLUSH PRIVILEGES;
```


Not Secure
192.168.147.129:30007
Hello from Flask with MySQL!


192.168.147.129:30007/users
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON
0:
email: "kirthiksubbiah@gmail.com"
id: 4
name: "kirthiksubbiah"
1:
email: "kirthiksubbiahp@gmail.com"
id: 5
name: "kirthiksubbiahp"