Cloud Computing - Mini Project Report

Deploying-flask-and-mongodb-in-k8s

April 2023

Submitted By:
Vishal Police patil | PES2UG20CS482
Mohammed Ryaan | PES2UG20CS521
Deepu Darshan S | PES2UG20CS508
Nachiketha CS | PES2UG20CS524

VI Semester Section H
PES University

Description

In this project, Kubernetes will be used to deploy a Flask web application and a MongoDB database. A container orchestration platform called Kubernetes aids in managing services and applications that are containerized.

A Python web framework called Flask makes it simple and quick for programmers to create web apps. A document-oriented NoSQL database called MongoDB keeps data in a flexible JSON-like format.

In this project, a Kubernetes cluster is created, and the Flask application and MongoDB database are deployed as distinct services inside the cluster. To store and retrieve data, the Flask application will communicate with the MongoDB database.

The Flask application and MongoDB database will be produced as Docker images and submitted to a Docker registry in order to deploy the application. The relevant pods, services, and deployments for managing the application and database will subsequently be created using Kubernetes.

A web browser may be used to visit the Flask application after it has been deployed, and the MongoDB database can be used to store and retrieve data. The Kubernetes platform will make sure the application is operating without a hitch and will deal with any scalability or failover problems that may occur.

Scope

A typical requirement in contemporary web development, the project seeks to show how to deploy a containerized web application and database in a Kubernetes cluster.

The project requires performing a number of tasks, including setting up the Kubernetes cluster, producing Docker images of the Flask application and MongoDB database, publishing those images to a Docker registry, and deploying the application and database using Kubernetes.

The project also entails setting up Kubernetes services, deployments, and pods as well as putting failover, scaling, and load balancing techniques into action to make sure the application functions well and can manage a lot of traffic.

The project may additionally involve extra responsibilities like establishing pipelines for continuous integration and deployment, putting in place security measures, monitoring, and logging.

A Flask web application and a MongoDB database can be deployed in a Kubernetes cluster using the project's overall goal of offering a complete end-to-end solution. The project will go over each step and configuration needed to build a reliable and expandable application architecture.

Methodology

Create a ConfigMap:

ConfigMap is an object that can be used to store configuration data in key-value pairs. in this project it used to store mongodb url.

creating deployments :

Creating a Kubernetes Deployment for MongoDB: this deployment is created using official mongo image.

Creating a Kubernetes Deployment for the Flask App: to create the deployment of flask application first, the image of the application is created using Dockerfile. and this image is used to create deployment.

Creating a Kubernetes Deployment for the mongo express:this deployment is created using official mongo-express image.

creating services :

Creating a Kubernetes Service for MongoDB: this service is created using and exposed using clusterIP type.

Creating a Kubernetes Service for Flask App: this service is created using and exposed using LoadBalancer type.

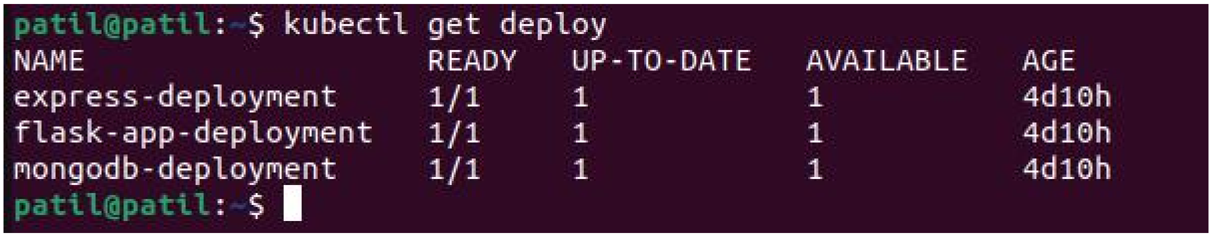
Creating a Kubernetes Service for mongo express: this service is created using and exposed using LoadBalancer type.

Test the Deployment:

Test the Flask app by visiting the Service endpoint in a web browser or using curl commands.

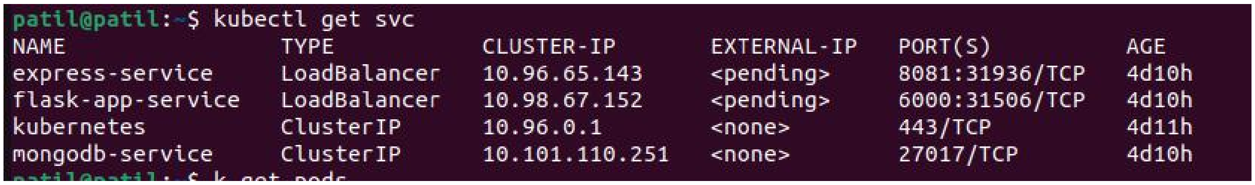
Testing

deployments are created



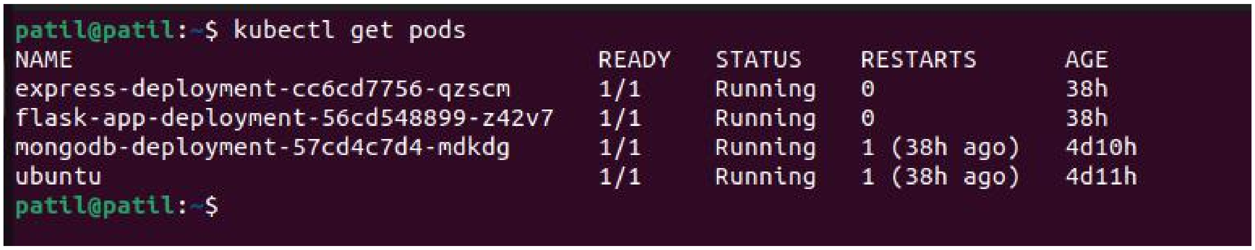
```
patil@patil:~$ kubectl get deploy
NAME                   READY   UP-TO-DATE   AVAILABLE   AGE
express-deployment     1/1     1            1           4d10h
flask-app-deployment   1/1     1            1           4d10h
mongodb-deployment     1/1     1            1           4d10h
patil@patil:~$ 
```

services are created



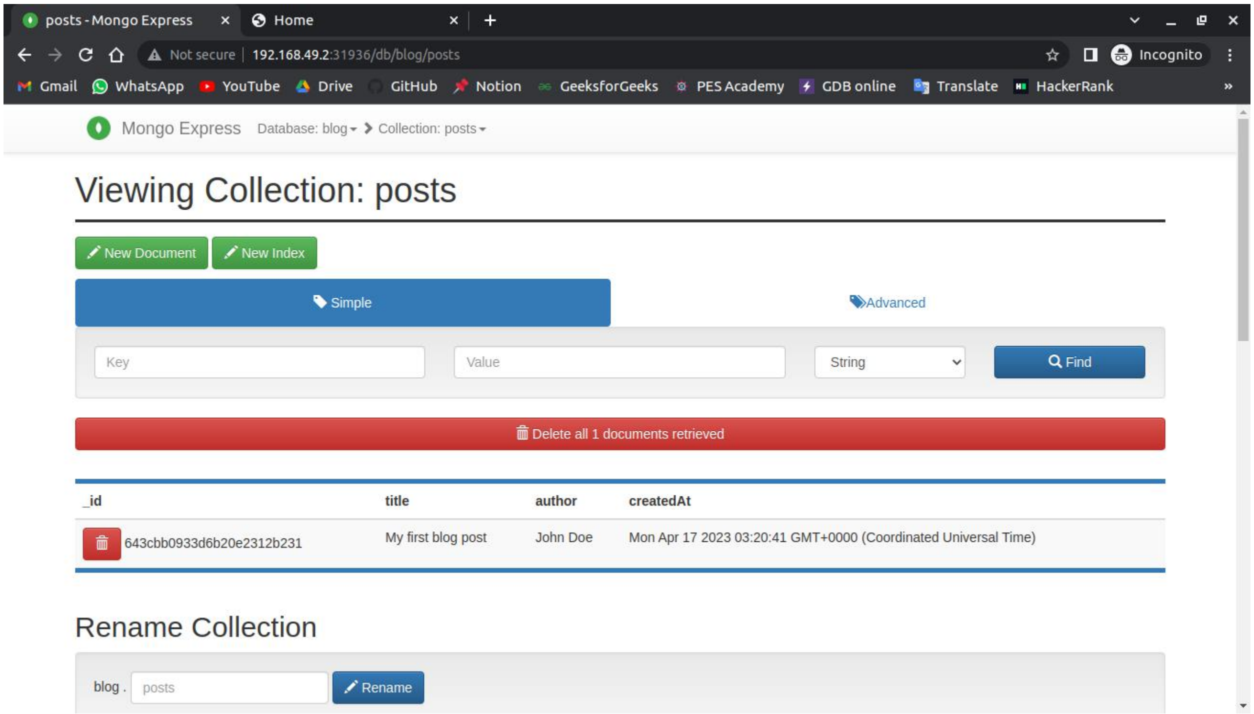
```
patil@patil:~$ kubectl get svc
NAME                TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
express-service     LoadBalancer   10.96.65.143     <pending>     8081:31936/TCP   4d10h
flask-app-service   LoadBalancer   10.98.67.152     <pending>     6000:31506/TCP   4d10h
kubernetes          ClusterIP      10.96.0.1        <none>        443/TCP          4d11h
mongodb-service     ClusterIP      10.101.110.251   <none>        27017/TCP        4d10h
patil@patil:~$ k get pods
```

all pods are running :



```
patil@patil:~$ kubectl get pods
NAME                                   READY   STATUS    RESTARTS       AGE
express-deployment-cc6cd7756-qzscm     1/1     Running   0              38h
flask-app-deployment-56cd548899-z42v7  1/1     Running   0              38h
mongodb-deployment-57cd4c7d4-mdkdg     1/1     Running   1 (38h ago)    4d10h
ubuntu                                 1/1     Running   1 (38h ago)    4d11h
patil@patil:~$
```

mongo-express running

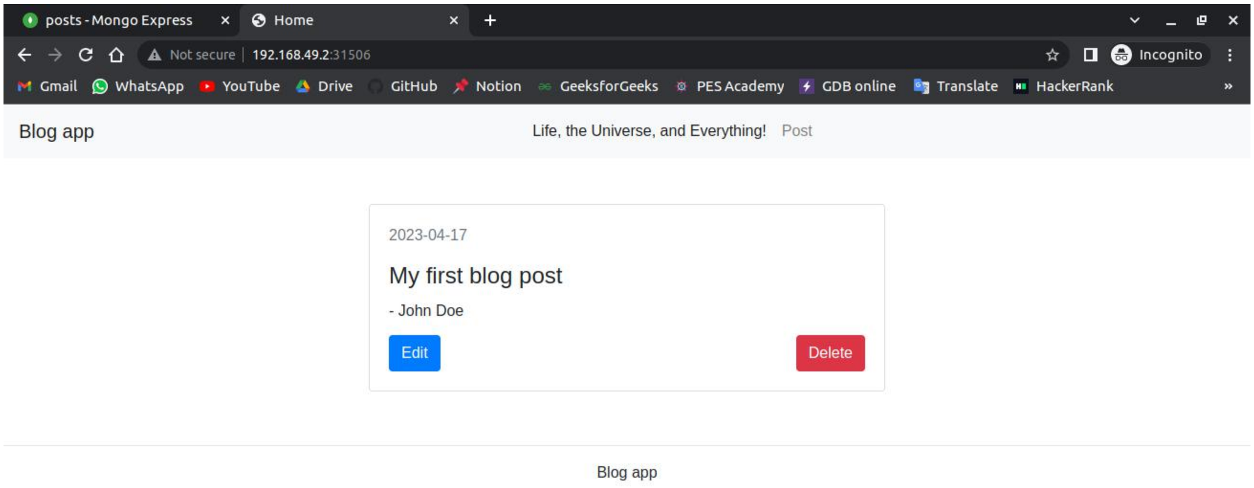


Viewing Collection: posts

_id	title	author	createdAt
643cbb0933d6b20e2312b231	My first blog post	John Doe	Mon Apr 17 2023 03:20:41 GMT+0000 (Coordinated Universal Time)

Rename Collection

flask app running running :



2023-04-17

My first blog post

- John Doe

Results and Conclusions :

As the result of this project we get the flask app and mongodb database running seperatly and communication inside the kubernetes cluster.
by this we can conslude that Kubernetes enables us to deploy and manage containerized applications across a cluster of servers or cloud instances. Kubernetes automates container deployment, scaling, and management, making it easier to manage large-scale container deployments.