Project #1 Multi-Container App Deployment

Gabriel de O. F. Gonçalves (uc2023238703) João Pereira (uc2023243538)

Faculdade de Ciências e Tecnologia da Universidade de Coimbra

IACD - MECD

1 Docker

Assignment #1

1. Pull the NODE base image:
   docker pull node
2. Create a Dockerfile with configurations for Backend and Frontend.
3. Build image for Backend:
   docker build -t backend_image .
4. Build image for Frontend:
   docker build -t frontend_image .
5. Pull MongoDB image:
   docker pull mongo
6. Run MongoDB container:
   docker run -d –name mongodb -p 27017:27017 mongo
7. Run Frontend container:
   docker run -d –name frontend -p 3000:3000 frontend_image
8. Run Backend container:
   docker run -d –name backend -p 80:80 backend_image

Assignment #2

Run MongoDB container

1. Create docker network:
   docker network create my_app
2. Check if the network was created:
   docker network ls
3. Run mongoDB container connected with the network and volumes options :
   docker run -d –name mongodb –rm –network my_app -v my_volume:/data/db mongo

Run Multi-container App

1. Change the localhost address for mongoDB in the backend app:
   From: mongodb://localhost:27017/course-goals
   To: mongodb://mongodb:27017/course-goals
2. Build backend image with the change in backend code:
   docker build -t my_backend_image .

3. Run the backend container with network and volumes options:
   docker run –name backend –rm –network my_app -p 80:80 -v my_volume:/app/backend my_backend_image

4. Run the frontend container with network options:
   docker run –name frontend –rm –network my_app -p 3000:3000 my_frontend_image

2 Docker-Compose

Assignment #3

1. Consider the documents created and changes in source from Assignment #1 and Assignment #2.

2. Backend and Frontend images are stored locally.

3. Create a **docker-compose.yaml** file.

4. Deploy the multi-container app using the Docker Compose:
   sudo docker-compose up -d

3 Kubernetes

Assignment #4

1. Docker login.

2. Build backend image:
   docker build -t jmbp75/iacd_pl4_backend .

3. Push backend image to DockerHub:
   docker push jmbp75/iacd_pl4_backend

4. Build frontend image:
   docker build -t jmbp75/iacd_pl4 .

5. Push frontend image to DockerHub:
   docker push jmbp75/iacd_pl4

Imperative Approach

1. Start minikube to use kubernetes locally:
   minikube start

2. Create object deployments
   kubectl create deployment mongodb-deployment –replicas=1 –image=mongo
   kubectl create deployment backend-deployment –replicas=2 –image=jmbp75/iacd_pl4_backend
   kubectl create deployment frontend-deployment –replicas=3 –image=jmbp75/iacd_pl4

3. Check deployments:
   kubectl get deployments

Declarative Approach

1. Create a folder named kubernetes to store all the created files.

2. Create backend deployment file.

3. Deploy backend deployment file:
   kubectl apply -f=backend-deployment.yaml

4. Create backend service file.

5. Deploy backend service file:
   kubectl apply -f=backend-service.yaml

6. Create frontend deployment file.

7. Deploy frontend deployment file:
   kubectl apply -f=frontend-deployment.yaml

8. Create frontend service file.

9. Deploy frontend service file:
   kubectl apply -f=frontend-service.yaml

10. Create mongodb deployment file.

11. Deploy mongodb deployment file:
    kubectl apply -f=mongodb-deployment.yaml

12. Create mongodb service file.

13. Deploy mongodb service file:
    kubectl apply -f=mongodb-service.yaml

Assignment #5

1. Change the address for mongoDB in the backend app:
   From: mongodb://mongodb:27017/course-goals
   To: mongodb://mongo-service:27017/course-goals; where mongo-service is the name defined in the mongodb-service.yaml file.

2. Build backend image with the changes in the backend code:
   docker build -t jmbp75/iacd_pl4_backend .

3. Push backend image with the changes in the backend code:
   Docker push jmbp75/iacd_pl4_backend

4. Apply deployments and services:
   kubectl apply -f=backend-deployment.yaml
   kubectl apply -f=backend-service.yaml
   kubectl apply -f=mongodb-deployment.yaml
   kubectl apply -f=mongodb-service.yaml

5. Create backend file for persistence.

6. Deploy backend file for persistence:
   kubectl apply -f=backend_pv.yaml

7. Create backend file for consistency.

8. Deploy backend file for consistency:
   kubectl apply -f=backend_pvc.yaml

9. Create mongodb file for persistence.

10. Deploy mongodb file for persistence:
    kubectl apply -f=mongo_pv.yaml

11. Create mongodb file for consistency.

12. Deploy mongodb file for consistency:
    kubectl apply -f=mongo_pvc.yaml

13. Find backend url:
    minikube service backend-service –url

14. Copy the ip and change the source code in the frontend app:
    From: http://localhost/goals
    To: http://*Ip found in the previous command*/goals

15. Build frontend image with the changes in the frontend code:
    docker build -t jmbp75/iacd_pl4 .

16. Push frontend image with the changes in the frontend code:
    Docker push jmbp75/iacd_pl4

17. Apply deployments and services for the frontend:
    kubectl apply -f=frontend-deployment.yaml
    kubectl apply -f=frontend-service.yaml

18. Open the App:
    minikube service frontend-service