12/15/2023

CS-4067
DevOps Project

FARQULEET FARHAT GONDAL
20I-0621
CS-B
https://github.com/fafnirLore/DevOps-Project

Table of Contents

Phase 1

Appointments-service Dockerfile

```
# Use the official Python image as the base image
FROM python:3.9-slim

# Set the working directory in the container
WORKDIR /app

# Install Flask and pymongo
RUN pip install --no-cache-dir Flask

# Copy the Flask application code to the working directory
COPY . .

# Expose the port that the Flask application will be running on
EXPOSE 7070

# Set the environment variable for the MongoDB URL
# ENV APPOINTMENTS_DB_URL="localhost:27017"

# Run the Flask application
CMD ["python", "app.py"]
```

Image Link: https://hub.docker.com/repository/docker/farquleet/appointments-service/general

Doctors-service Dockerfile

```
# Use the official Python image as the base image
FROM python:3.9-slim


# Install MongoDB
# RUN apt-get update && apt-get install -y gnupg2 && \
#     wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | apt-key add
- && \
#     echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu
focal/mongodb-org/5.0 multiverse" | tee /etc/apt/sources.list.d/mongodb-org-
5.0.list && \
#     apt-get update && apt-get install -y mongodb-org && \
#     rm -rf /var/lib/apt/lists/*

# Set the working directory in the container
WORKDIR /app/doctors
```

```
# Install Flask and pymongo
RUN pip install --no-cache-dir Flask

# Copy the Flask application code to the working directory
COPY . .

# Expose the port that the Flask application will be running on
EXPOSE 9090

# Set the environment variable for the MongoDB URL
ENV APPOINTMENTS_DB_URL="localhost:27017"

# Run the Flask application
CMD ["python", "app.py"]
```

Image Link: https://hub.docker.com/repository/docker/farquleet/doctors-service/general

Frontend-service Dockerfile

```
# Use a base image with Node.js pre-installed
FROM node:14

# Set the working directory inside the container
WORKDIR /app

# Copy package.json and package-lock.json to the working directory
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the entire application to the working directory
COPY . .

# Expose the port on which your Node.js application listens
EXPOSE 3000

# Define the command to start your Node.js application
CMD [ "npm", "start" ]
```

Image Link: https://hub.docker.com/repository/docker/farquleet/frontend-service/general

Docker-compse file

```yaml
version: "3"
services:
  appointments-service:
    image: farquleet/appointments-service
    # build:
    #   context: ./appointments
    #   dockerfile: Dockerfile
    ports:
      - 7070:7070
    # depends_on:
    #   - mongo

  doctors-service:
    image: farquleet/doctors-service
    # build:
    #   context: ./doctors
    #   dockerfile: Dockerfile
    ports:
      - 9090:9090
    # depends_on:
    #   - mongo

  frontend:
    image: farquleet/frontend-service
    # build:
    #   context: ./frontend
    #   dockerfile: Dockerfile
    ports:
      - 3000:3000
    depends_on:
      - appointments-service
      - doctors-service
    environment:
      - APPOINTMENTS_SERVICE_URL=172.20.0.3:7070
      - DOCTORS_SERVICE_URL=172.20.0.2:9090

  # mongo:
  #   image: mongo
  #   ports:
  #     - 27017:27017
```

Phase 2

Appointments-service Workflow

```
name: Appointments Service CI/CD PipeLine

on:
  pull_request:
    paths:
      - 'appointments/**'
  push:
    branches:
      - main
    paths:
      - 'appointments/**'

jobs:
  build-and-push:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v3

      - name: Build and push Docker image
        env:
          DOCKER_USERNAME: ${{ secrets.DOCKER_USERNAME }}
          DOCKER_PASSWORD: ${{ secrets.DOCKER_PASSWORD }}
        run: |
          docker build -t farquleet/appointments-service:${{ github.sha }}
./appointments
          docker login -u $DOCKER_USERNAME -p $DOCKER_PASSWORD
          docker push farquleet/appointments-service:${{ github.sha }}
```

Doctors-service Workflow

```
name: Doctors Service CI/CD PipeLine

on:
  pull_request:
    paths:
      - 'doctors/**'
  push:
```

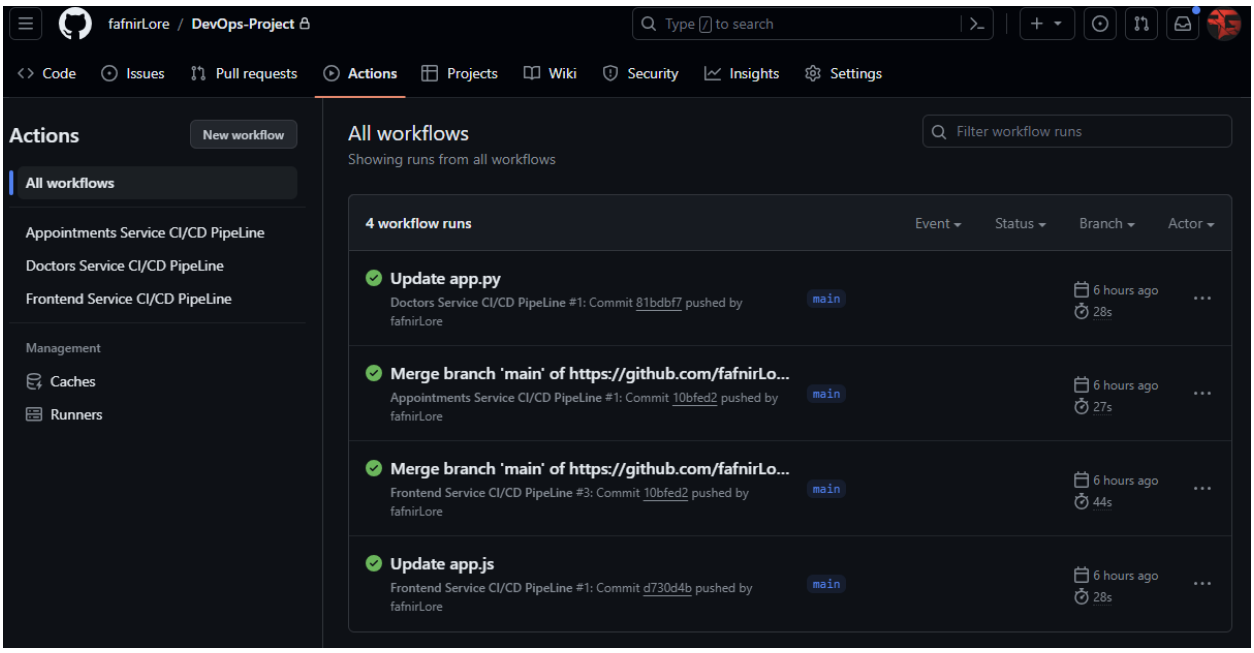
```yaml
    branches:
      - main
    paths:
      - 'doctors/**'


jobs:
  build-and-push:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v3

      - name: Build and push Docker image
        env:
          DOCKER_USERNAME: ${{ secrets.DOCKER_USERNAME }}
          DOCKER_PASSWORD: ${{ secrets.DOCKER_PASSWORD }}
        run: |
          docker build -t farquleet/doctors-service:${{ github.sha }} ./doctors
          docker login -u $DOCKER_USERNAME -p $DOCKER_PASSWORD
          docker push farquleet/doctors-service:${{ github.sha }}
```

Frontend-service Workflow

```yaml
name: Frontend Service CI/CD PipeLine

on:
  pull_request:
    paths:
      - 'frontend/**'
  push:
    branches:
      - main
    paths:
      - 'frontend/**'


jobs:
  build-and-push:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
```

```
      uses: actions/checkout@v3

    - name: Build and push Docker image
      env:
        DOCKER_USERNAME: ${{ secrets.DOCKER_USERNAME }}
        DOCKER_PASSWORD: ${{ secrets.DOCKER_PASSWORD }}
      run: |
        docker build -t farquleet/frontend-service:${{ github.sha }} ./frontend
        docker login -u $DOCKER_USERNAME -p $DOCKER_PASSWORD
        docker push farquleet/frontend-service:${{ github.sha }}
```

Workflow Runs



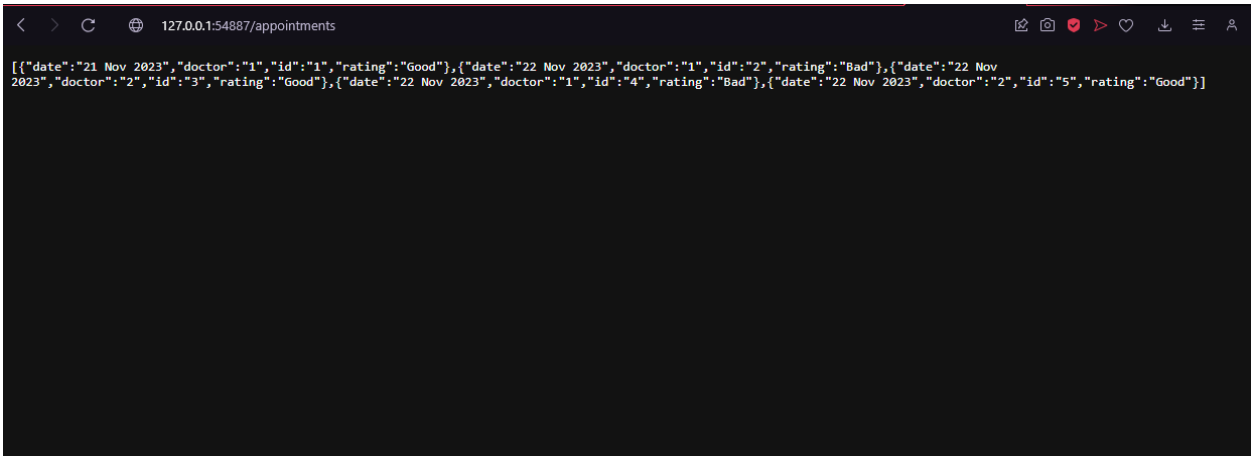
Phase 3

Appointments-service Deployment & Service

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: appointments-deployment
spec:
  replicas: 1
  selector:
```

```yaml
    matchLabels:
      app: appointments
  template:
    metadata:
      labels:
        app: appointments
    spec:
      containers:
        - name: appointments
          image: farquleet/appointments-service:latest
          ports:
            - containerPort: 7070

---
apiVersion: v1
kind: Service
metadata:
  name: appointments-service
spec:
  selector:
    app: appointments
  ports:
    - protocol: TCP
      port: 7070
      targetPort: 7070
```

```
C:\Users\Thinkpad\Desktop\Current\Semester7\devops\doctor-appointment-system>minikube service appointments-service
W1215 16:20:06.521493    9160 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": context no
t found: open C:\Users\Thinkpad\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688f\meta.json: Th
e system cannot find the path specified.
|-----------|----------------------|-------------|--------------|
| NAMESPACE |         NAME         | TARGET PORT |     URL      |
|-----------|----------------------|-------------|--------------|
| default   | appointments-service |             | No node port |
|-----------|----------------------|-------------|--------------|
😿  service default/appointments-service has no node port
🏃  Starting tunnel for service appointments-service.
|-----------|----------------------|-------------|------------------------|
| NAMESPACE |         NAME         | TARGET PORT |          URL           |
|-----------|----------------------|-------------|------------------------|
| default   | appointments-service |             | http://127.0.0.1:54887 |
|-----------|----------------------|-------------|------------------------|
🎉  Opening service default/appointments-service in default browser...
❗  Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```



Doctors-service Deployment & Service

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: doctors-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: doctors
  template:
    metadata:
      labels:
        app: doctors
    spec:
      containers:
        - name: doctors
          image: farquleet/doctors-service:latest
          ports:
            - containerPort: 9090
---
apiVersion: v1
kind: Service
metadata:
  name: doctors-service
spec:
  selector:
    app: doctors
  ports:
    - protocol: TCP
```

```
    port: 9090
    targetPort: 9090
```

```
C:\Users\Thinkpad\Desktop\Current\Semester7\devops\doctor-appointment-system>minikube service doctors-service
W1215 16:21:17.512173   11104 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": context no
t found: open C:\Users\Thinkpad\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688f\meta.json: Th
e system cannot find the path specified.
|-----------|-----------------|-------------|--------------|
| NAMESPACE |      NAME       | TARGET PORT |     URL      |
|-----------|-----------------|-------------|--------------|
| default   | doctors-service |             | No node port |
|-----------|-----------------|-------------|--------------|
😿  service default/doctors-service has no node port
🏃  Starting tunnel for service doctors-service.
|-----------|-----------------|-------------|------------------------|
| NAMESPACE |      NAME       | TARGET PORT |          URL           |
|-----------|-----------------|-------------|------------------------|
| default   | doctors-service |             | http://127.0.0.1:54927 |
|-----------|-----------------|-------------|------------------------|
🎉  Opening service default/doctors-service in default browser...
❗  Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```

```
127.0.0.1:54927/doctors

[{"firstName":"Muhammad Ali","id":"1","lastName":"Kahoot","speciality":"DevOps"},{"firstName":"Good","id":"2","lastName":"Doctor","speciality":"Test"}]
```

Frontend-service Deployment & Service

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

```yaml
          image: farquleet/frontend-service:latest
          ports:
            - containerPort: 3000
          env:
            - name: APPOINTMENTS_SERVICE_URL
              value: "http://appointments-service:7070"
            - name: DOCTORS_SERVICE_URL
              value: "http://doctors-service:9090"
---
apiVersion: v1
kind: Service
metadata:
  name: frontend-service
spec:
  selector:
    app: frontend
  ports:
    - protocol: TCP
      port: 3000
      targetPort: 3000
```

Kubernetes Dashboard

Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
frontend-deployment-787bbf655-6mvpc	farquleet/frontend-service:latest	app: frontend pod-template-hash: 787bbf655	minikube	Running	0	-	-	17 minutes ago
doctors-deployment-56f7c9dbf7-z5f4b	farquleet/doctors-service:latest	app: doctors pod-template-hash: 56f7c9dbf7	minikube	Running	0	-	-	17 minutes ago
appointments-deployment-5bf6b8777c-87z5s	farquleet/appointments-service:latest	app: appointments pod-template-hash: 5bf6b8777c	minikube	Running	0	-	-	17 minutes ago

Services

Name	Labels	Type	Cluster IP	Internal Endpoints	External Endpoints	Created ↑
frontend-service	-	ClusterIP	10.104.223.59	frontend-service:3000 TCP frontend-service:0 TCP	-	6 minutes ago
doctors-service	-	ClusterIP	10.110.112.132	doctors-service:9090 TCP doctors-service:0 TCP	-	6 minutes ago
appointments-service	-	ClusterIP	10.104.175.112	appointments-service:7070 TCP appointments-service:0 TCP	-	6 minutes ago
kubernetes	component: apiserver provider: kubernetes	ClusterIP	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	8 minutes ago

Phase 4

Appointments-service Deployment & Service

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: appointments-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: appointments
  template:
    metadata:
      labels:
        app: appointments
    spec:
      containers:
        - name: appointments
          image: farquleet/appointments-service:latest
          ports:
            - containerPort: 7070
          resources:
            limits:
              cpu: 500m
              memory: 512Mi
            requests:
              cpu: 200m
              memory: 256Mi
          readinessProbe:
            httpGet:
              path: /health
              port: 80
            initialDelaySeconds: 10
            periodSeconds: 5
          livenessProbe:
            httpGet:
              path: /health
              port: 80
            initialDelaySeconds: 15
            periodSeconds: 10

---
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: appointments-service
spec:
  selector:
    app: appointments
  ports:
    - protocol: TCP
      port: 7070
      targetPort: 7070
```

Doctors-service Deployment & Service

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: doctors-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: doctors
  template:
    metadata:
      labels:
        app: doctors
    spec:
      containers:
        - name: doctors
          image: farquleet/doctors-service:latest
          ports:
            - containerPort: 9090
          resources:
            limits:
              cpu: 500m
              memory: 512Mi
            requests:
              cpu: 200m
              memory: 256Mi
          readinessProbe:
            httpGet:
              path: /health
```

```
              port: 80
            initialDelaySeconds: 10
            periodSeconds: 5
          livenessProbe:
            httpGet:
              path: /health
              port: 80
            initialDelaySeconds: 15
            periodSeconds: 10
---
apiVersion: v1
kind: Service
metadata:
  name: doctors-service
spec:
  selector:
    app: doctors
  ports:
    - protocol: TCP
      port: 9090
      targetPort: 9090
```

Frontend-service Deployment & Service

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: frontend
  template:
    metadata:
      labels:
        app: frontend
    spec:
      containers:
        - name: frontend
          image: farquleet/frontend-service:latest
          ports:
            - containerPort: 3000
```

```yaml
          env:
            - name: APPOINTMENTS_SERVICE_URL
              value: "http://appointments-service:7070"
            - name: DOCTORS_SERVICE_URL
              value: "http://doctors-service:9090"
          resources:
            limits:
              cpu: 500m
              memory: 512Mi
            requests:
              cpu: 200m
              memory: 256Mi
          readinessProbe:
            httpGet:
              path: /health
              port: 80
            initialDelaySeconds: 10
            periodSeconds: 5
          livenessProbe:
            httpGet:
              path: /health
              port: 80
            initialDelaySeconds: 15
            periodSeconds: 10
---
apiVersion: v1
kind: Service
metadata:
  name: frontend-service
spec:
  selector:
    app: frontend
  ports:
    - protocol: TCP
      port: 3000
      targetPort: 3000
```