# \d\DREAM\bodhi-learn\backend\services\identity-management\user-service\.github\workflows\ci.yml

name: CI

on:

  push:

    branches: [ main, develop ]

  pull\_request:

    branches: [ main, develop ]

jobs:

  test:

    runs-on: ubuntu-latest

    services:

      postgres:

        image: postgres:13

        env:

          POSTGRES\_USER: test

          POSTGRES\_PASSWORD: test

          POSTGRES\_DB: test\_db

        ports:

          - 5432:5432

        options: >-

          --health-cmd pg\_isready

          --health-interval 10s

          --health-timeout 5s

          --health-retries 5

      redis:

        image: redis:6

        ports:

          - 6379:6379

        options: >-

          --health-cmd "redis-cli ping"

          --health-interval 10s

          --health-timeout 5s

          --health-retries 5

      mongodb:

        image: mongo:5

        ports:

          - 27017:27017

      rabbitmq:

        image: rabbitmq:3.8-management

        ports:

          - 5672:5672

          - 15672:15672

    steps:

    - uses: actions/checkout@v3

    - name: Use Node.js

      uses: actions/setup-node@v3

      with:

        node-version: '16.x'

        cache: 'npm'

    - name: Install dependencies

      run: npm ci

    - name: Run linter

      run: npm run lint

    - name: Run tests

      run: npm test

      env:

        NODE\_ENV: test

        DATABASE\_URL: postgresql://test:test@localhost:5432/test\_db

        MONGODB\_URL: mongodb://localhost:27017/test

        REDIS\_URL: redis://localhost:6379

        RABBITMQ\_URL: amqp://localhost:5672

        JWT\_SECRET: test\_secret

    - name: Upload coverage

      uses: codecov/codecov-action@v3

      with:

        token: ${{ secrets.CODECOV\_TOKEN }}

  build:

    needs: test

    runs-on: ubuntu-latest

    if: github.event\_name == 'push' && (github.ref == 'refs/heads/main' || github.ref == 'refs/heads/develop')

    steps:

    - uses: actions/checkout@v3

    - name: Set up Docker Buildx

      uses: docker/setup-buildx-action@v2

    - name: Login to Docker Hub

      uses: docker/login-action@v2

      with:

        username: ${{ secrets.DOCKERHUB\_USERNAME }}

        password: ${{ secrets.DOCKERHUB\_TOKEN }}

    - name: Build and push

      uses: docker/build-push-action@v4

      with:

        context: .

        push: true

        tags: |

          bodhilearn/user-service:latest

          bodhilearn/user-service:${{ github.sha }}

        cache-from: type=registry,ref=bodhilearn/user-service:buildcache

        cache-to: type=registry,ref=bodhilearn/user-service:buildcache,mode=max

  deploy-staging:

    needs: build

    runs-on: ubuntu-latest

    if: github.ref == 'refs/heads/develop'

    steps:

    - uses: actions/checkout@v3

    - name: Install doctl

      uses: digitalocean/action-doctl@v2

      with:

        token: ${{ secrets.DIGITALOCEAN\_TOKEN }}

    - name: Save DigitalOcean kubeconfig

      run: doctl kubernetes cluster kubeconfig save staging

    - name: Deploy to staging

      run: |

        kubectl set image deployment/user-service user-service=bodhilearn/user-service:${{ github.sha }} -n bodhi-learn

        kubectl rollout status deployment/user-service -n bodhi-learn

  deploy-production:

    needs: build

    runs-on: ubuntu-latest

    if: github.ref == 'refs/heads/main'

    environment: production

    steps:

    - uses: actions/checkout@v3

    - name: Install doctl

      uses: digitalocean/action-doctl@v2

      with:

        token: ${{ secrets.DIGITALOCEAN\_TOKEN }}

    - name: Save DigitalOcean kubeconfig

      run: doctl kubernetes cluster kubeconfig save production

    - name: Deploy to production

      run: |

        kubectl set image deployment/user-service user-service=bodhilearn/user-service:${{ github.sha }} -n bodhi-learn

        kubectl rollout status deployment/user-service -n bodhi-learn

    - name: Create GitHub Release

      uses: actions/create-release@v1

      env:

        GITHUB\_TOKEN: ${{ secrets.GITHUB\_TOKEN }}

      with:

        tag\_name: v${{ github.run\_number }}

        release\_name: Release v${{ github.run\_number }}

        body: |

          Release of user-service version v${{ github.run\_number }}

          Changes in this release:

          ${{ github.event.head\_commit.message }}