

Континуирана интеграција и испорака

Докеризација и испорака на Django CRM апликација



Универзитет „Св. Кирил и Методиј“

Факултет за информатички науки и компјутерско инженерство

Автор: Филип Тодоров

Ментори: проф. Д-р Милош Јовановиќ

проф. Д-р Панче Рибарски

12.02.2025

Содржина

Континуирана интеграција и испорака.....	1
Докеризација и испорака на Django CRM апликација.....	1
.....	1
Краток опис за апликацијата:.....	3
Поставување на јавен Git репозиториум:.....	3
Докеризација на апликацијата:.....	3
Оркестрирање на апликацијата и базата со Docker Compose:.....	3
Континуирана интеграција:.....	4
Service за апликацијата:.....	4
Ingress за апликацијата:.....	4
StatefulSet со потребните ConfigMaps/Secrets:.....	5
Namespace за кластерот.....	5

- **Краток опис за апликацијата:**

Апликацијата е изработена со Django и служи за CRM, односно управување со корисници и нивните информации. Може да се извршуваат CRUD операции на записите на корисниците. Корисниците можат да се регистрираат и логираат на апликацијата со свој профил. Апликацијата користи MySQL база на податоци за складирање на информациите за корисниците.

- **Поставување на јавен Git репозиториум:**

Апликацијата е поставена на GitHub на следниот линк:

<https://github.com/FilipTodorov1/cicdproject>

- **Докеризација на апликацијата:**

За докеризација на апликацијата, искористив **Dockerfile**, со кој се стартува python image, се инсталираат потребните dependencies за MySQL, и тие што се наведени во **requirements.txt**. Потоа проектот се копира во контејнер и се пушта да работи на порта 8000.

За градење на сликата ја искористив командата **docker build -t django-app .** Docker контејнерот го стартував со командата **docker run -d -p 8000:8000 --name django-app django-app**

Со оваа команда се мапира локалната порта на контејнерот во овој случај 8000:8000, а со тоа апликацијата е пристаплива на <http://localhost:8000>

[cicdproject/Dockerfile at master · FilipTodorov1/cicdproject](#)

- **Оркестрирање на апликацијата и базата со Docker Compose:**

За оркестрирање на апликацијата и на базата креирав **docker-compose.yml** фајл во кој се дефинирани два сервиси – **web** за Django апликацијата и **mysql-db** за базата на податоци. Ги додадов конфигурациите за апликацијата да се поврзе со базата и ја мапирав локалната порта 8000 до портата 8000 во контејнерот за да може да се пристапи апликацијата на <http://localhost:8000>

Со командата **docker-compose up -d** ги стартував двата контејнери односно Django апликацијата и MySQL базата. Потоа ја извршив командата **docker-compose exec web python manage.py migrate** за да се постави базата.

За да се осигурам дека апликацијата е поврзана со базата на податоци и дека се создадени табелите ја извршив командата **docker-compose exec mysql-db mysql -u root -p***** djangoappdb -e "SHOW TABLES;"**

[cicdproject/docker-compose.yml at master · FilipTodorov1/cicdproject](#)

- **Континуирана интеграција:**

За овој дел користам **GitHub Actions** за автоматизирано градење и поставување на Docker слика од мојата апликација на Docker Hub. Првин креирав workflow фајл **docker-ci.yml** кој се извршува на секоја измена на master гранката. Workflow-от прави check-out на кодот се логира на Docker Hub користејќи GitHub Secrets, ја гради Docker сликата, го стартува контејнерот и проверува дали тој е активен.

За CD делот пробав да направам deploy на апликацијата користејќи SSH, ја отворив портата 22 на мојот рутер за да овозможи SSH пристап до серверот но имав проблеми со SSH портата кои не успеав да ги разрешам и затоа не го направив овој дел.

[cicdproject/.github/workflows/docker-ci.yml at master · FilipTodorov1/cicdproject](#)

- **Kubernetes deployment со ConfigMaps/Secrets:**

За deployment на Django апликацијата, креирав **deployment.yaml** и **configmap.yaml** file, во configmap има конфигурација за поврзување со базата, како и кои домени може да ги опслужува апликацијата.

[cicdproject/k8s/deployment.yaml at master · FilipTodorov1/cicdproject](#)

- **Service за апликацијата:**

Креирав сервис кој ќе ја изложи апликацијата на мрежата. За ова користам **NodePort** кој ја изложува апликацијата на специфична порта на node во кластерот. Сервисот со селектор **django-app** го насочува сообраќајот кон подовите, ја мапира портата 8000 на кластерот кон 8000 порта во контејнерот, а ова е портата на која работи Django апликацијата во подот. Со ова можам да ја пристапам апликацијата надвор од кластерот на **localhost:30978**

[cicdproject/k8s/service.yaml at master · FilipTodorov1/cicdproject](#)

- **Ingress за апликацијата:**

За овој дел креирав Ingress ресурс **django-ingress** во **django-app** кој е конфигуриран да слуша за барања на доменот **django-app.local**. Тука ги дефинирам сите барања на доменот **django-app.local** да се пренасочуваат до сервисот **django-service** на порта 8000. Со ова апликацијата може да се пристапи на <http://django-app.local>. За процесирање на барањата во Ingress користам Nginx Ingress Controller. Исто така за да работи овој дел направив промени во /etc/hosts.

[cicdproject/k8s/ingress.yaml at master · FilipTodorov1/cicdproject](#)

- **StatefulSet со потребните ConfigMaps/Secrets:**

Во **mysql.yaml**, **mysql-db** претставува StatefulSet за управување со базата на податоци, креира сервис за да ја експонира MySQL базата на податоци во кластерот и и овозможува на Django апликацијата да се поврза со базата преку порта 3306. **mysql-pvc** е PVC кој обезбедува постојано складирање на податоците во базата, во **mysql-config** се конфигурирани податоци за базата, а во **mysql-secret** се чува лозинката на root корисникот на MySQL

[cicdproject/k8s/mysql.yaml at master · FilipTodorov1/cicdproject](#)
[cicdproject/k8s/configmap.yaml at master · FilipTodorov1/cicdproject](#)

- **Namespace за кластерот**

Во Kubernetes кластерот креирав посебен namespace **django-app** за да ги изолирам и организирам сите ресурси поврзани со апликацијата. Овој namespace ги содржи ресурсите за Deployment, Service, Ingress, StatefulSet, ConfigMap и Secret ресурсите.

[cicdproject/k8s/namespace.yaml at master · FilipTodorov1/cicdproject](#)