**namespace**

Креирав посебен namespace за да ги изолирам ресурсите на проектот и да ја подобрам организацијата и управувањето во кластерот.

**deployments**

Имам две Deployment-и – еден за backend и еден за frontend. Backend-от е Flask API, а frontend-от е React апликација. Користам Docker images кои ги имам објавено на Docker Hub. Конфигурацијата е надворешно управувана преку ConfigMap и Secret.

**services**

Креирав Kubernetes Services за да ги направам компонентите достапни една на друга преку стабилни DNS имиња. MongoDB сервисот е headless, што овозможува директна комуникација меѓу подови ако се потребни StatefulSet-и.

**secrets**

Осетливите податоци како URI за базата и тајниот клуч се чуваат во Kubernetes Secret. Ова овозможува побезбедно управување со креденцијали

**configmap**

„Со ConfigMap ги чувам јавните конфигурации, на пример URL до backend API-то што frontend-от го користи

**Docker compose**  
Креира 3 сервиси: MongoDB, Flask backend и React frontend.

* Сите се поврзани на ист Docker network.
* mongo сервисот користи перзистентен волумен (mongo-data).
* backend зависи од Mongo, frontend зависи од backend.
* frontend користи порт 3000 надворешно, а внатрешно работи на 80.

🗣️ **На презентација кажи:**

„За локален развој и тестирање користам Docker Compose. Со ова можам да го покренам целиот стек – база, backend и frontend – со една команда. Исто така, базата е перзистентна преку Docker volume.“

**frontend dockerfile**„React frontend-от го градиме во multistage Docker image – прво се билдa со Node, а потоа финалните статиќни фајлови се сервираат преку nginx. Користам и environment variable за да го поставам backend API URL-то при build-time.“

**backend dockerfile**Backend-от е Flask апликација во Python. Се користи минимална слика (slim), што го намалува финалниот image. Flask серверот е конфигуриран да слуша на 5000 и е подготвен за контејнеризација.“

**stateful set**  
Наместо Deployment, тука користиш StatefulSet за MongoDB.

* Обезбедува уникатно име на подот и стабилно складирање преку PersistentVolumeClaim.
* Складиштето останува и ако подот се избрише или ресетира.

🗣️ **На презентација кажи:**

„Во Kubernetes користам StatefulSet за MongoDB за да овозможам стабилно и перзистентно складирање. Ова е потребно бидејќи базата има состојби кои треба да се зачуваат дури и при ресетирање на подот.“

**Ingres**  
Овозможува надворешен пристап преку worldapp.local.

* / оди до frontend.
* /api оди до backend.

🗣️ **На презентација кажи (повторување):**

„Ingress контролерот овозможува да пристапам до апликацијата преку едно URL – worldapp.local. Рутите / и /api се пренасочуваат до соодветните сервиси, со што frontend и backend се одделени, но функционираат заедно.“

Ingress е Kubernetes објект кој овозможува **надворешен пристап** до сервисите во кластерот преку HTTP/HTTPS.

**🔧 *Се користи за:***

* Рутирање на барања според патека или домен (пример: /, /api, app.domain.com)
* Служи како „врата“ кон апликацијата (влезна точка)
* Се поставува над Service-ите, и се користи со Ingress Controller (на пр. nginx)