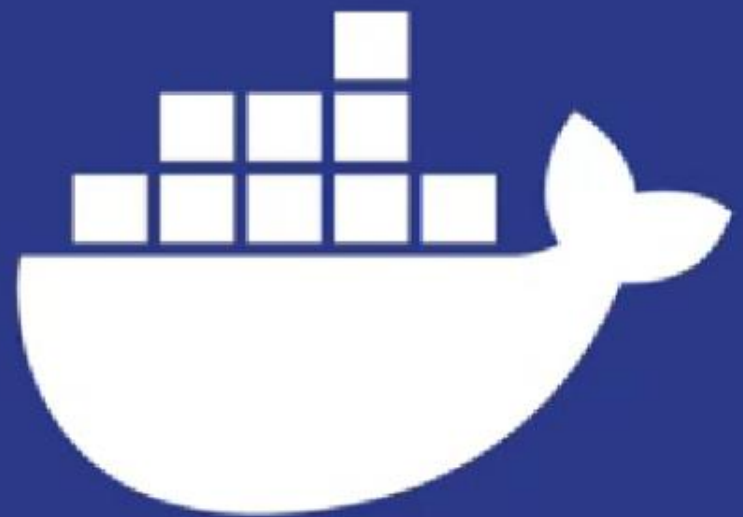


Docker alapok

Bevezetés a konténerek világába

Dr.Gerják István PhD

WSNE

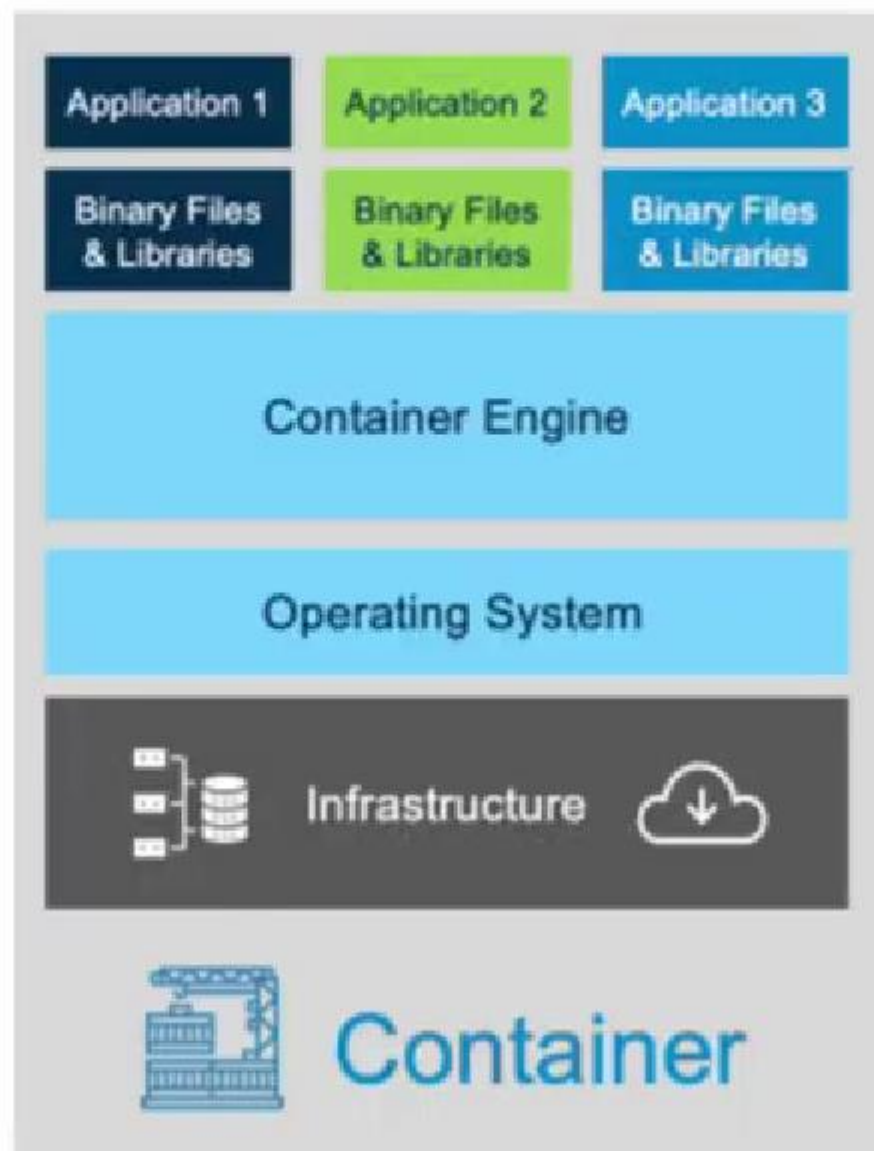
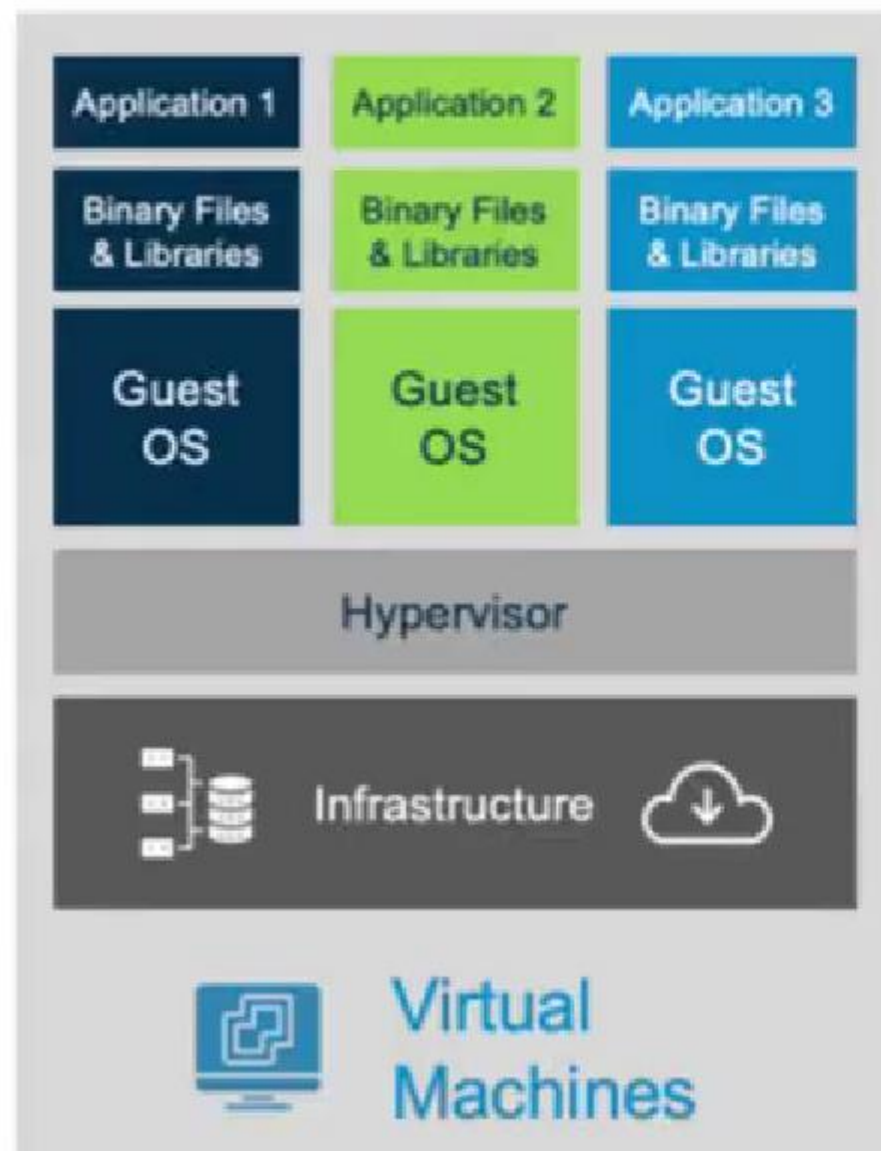


Bevezető

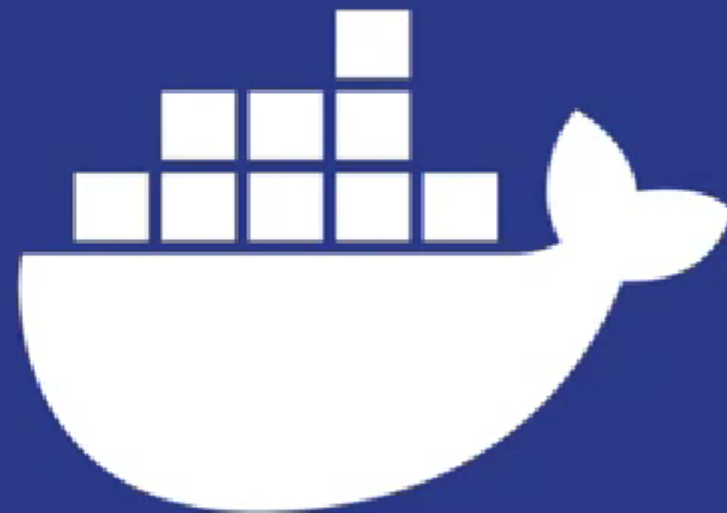
- Modern megközelítés az alkalmazások fejlesztésére, telepítésére és futtatására.
- **Konzisztens környezetekben** történő munkavégzés lehetősége.



Konténerizáció vs Virtualizáció



Docker konténerek kezelése

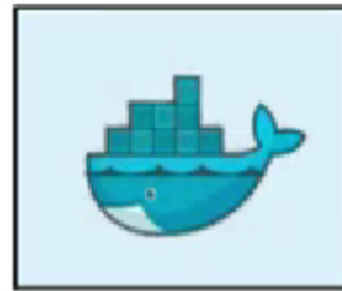


Dockerfile → Docker Image → Container



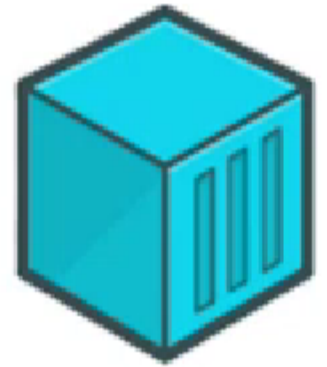
Dockerfile

Build →



Docker
Image

Run →



Docker
Container

Docker Hub

- <https://hub.docker.com>
- ``docker pull <image>``
- ``docker login``
- ``docker push <image>``

Alap docker parancsok

- ``docker run``
- ``docker logs``
- ``docker start``, ``docker stop`` és ``docker restart``
- ``docker ps``, ``docker images``
- ``docker build``
- ``docker exec``

1. gyakorlat

Docker konténerek kezelése

1. modul workshop - Docker Desktop telepítése, konténerek kezelése

Docker környezetben

- Docker Desktop telepítése
- Egy hivatalos Docker image futtatása (nginx)
- Futó konténerek listázása, részletes adatok lekérdezése
- Port-forward szabály beállítása
- A konténer logjainak vizsgálata
- A konténer megállítása, újraindítása, törlése

Cél:

- Telepítsd a Docker Desktopot
- Indítsd el és kezeld az első konténeredet

1. Docker Desktop telepítése

- Töltsd le és telepítsd a Docker Desktopot a hivatalos weboldalról.

A telepítés után ellenőrizd a sikeres telepítést az alábbi parancs futtatásával:

```
docker --version
```



2. Nginx konténer letöltése és futtatása

- Parancs:

```
docker pull nginx  
docker run -d -p 8080:80 nginx
```



- Ez a parancs letölti az Nginx képet a Docker Hubról, és elindítja azt úgy, hogy a gépeden a 8080-as portot hozzárendeli a konténer belső 80-as portjához.

3. Futtatott konténerek listázása

- Parancs:

```
docker ps
```



- Ez a parancs kilistázza a jelenleg futó konténereket. Jegyezd meg a konténer ID-ját.

4. Konténer naplók ellenőrzése

- Parancs:

```
docker logs <container_id>
```



- Cseréld ki a `<container_id>` részt a futó konténer ID-jára, hogy megnézd a naplókat.

5. Konténer leállítása és eltávolítása

- Parancsok:

```
docker stop <container_id>
docker rm <container_id>
```



- A `docker stop` parancs leállítja a futó konténert, a `docker rm` pedig eltávolítja azt a rendszerből.

6. Konténer futtatása volume-mal:

- Hozz létre egy `index.html` fájlt a következő tartalommal `index.html` néven

```
<html>
<head><title>My Docker Page</title></head>
<body><h1>Hello from Docker!</h1></body>
</html>
```



- Csatold fel a létrehozott fájlt az `/usr/share/nginx/html` mappa alá

- Parancs:

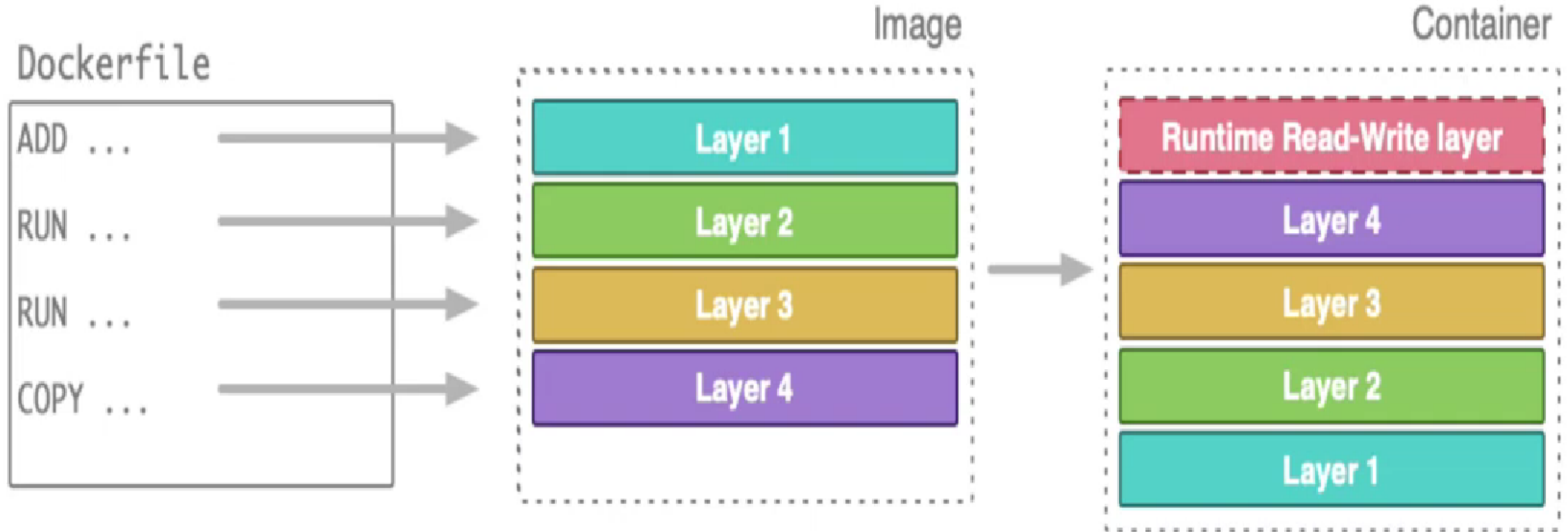
```
docker run -d -p 8080:80 -v /path/to/local/directory:/usr/share/nginx/html nginx
```



Saját dockerfile készítése



Dockerfile felépítése és rétegei



Dockerfile példa



```
1 FROM node:18.16.0-alpine
2
3 RUN mkdir -p /opt/app
4 WORKDIR /opt/app
5
6 COPY src/package.json src/package-lock.json .
7
8 RUN npm install
9
10 COPY src/ .
11 EXPOSE 3000
12
13 CMD [ "node", "app.js"]
```

Hasznos parancsok

- `docker images`
- `docker rmi <image_id>`
- `docker tag <image_id> felhasznalonev/image-nev`
- `docker push felhasznalonev/image-nev`

2. gyakorlat

Saját Docker image készítése

2. modul workshop - Saját Docker image készítése

- https://docs.docker.com/get-started/workshop/02_our_app/
- Dockerfile elkészítése
- Docker image buildelése a Dockerfile alapján
- Docker image elindítása (1. szekció alapján)
- Docker image publikálása a Docker Hubra (regisztráció ha szükséges, docker login, docker push)

Cél:

- Egy egyszerű alkalmazás dockerizálása
- Saját docker image létrehozása és buildelése
- (Opcionális) Docker image megosztása Docker Hubon

1. Alkalmazás repo klónozása Klónozd vagy tölts le a példaalkalmazást GitHubról: <https://github.com/docker/getting-started-app/>

Parancs:

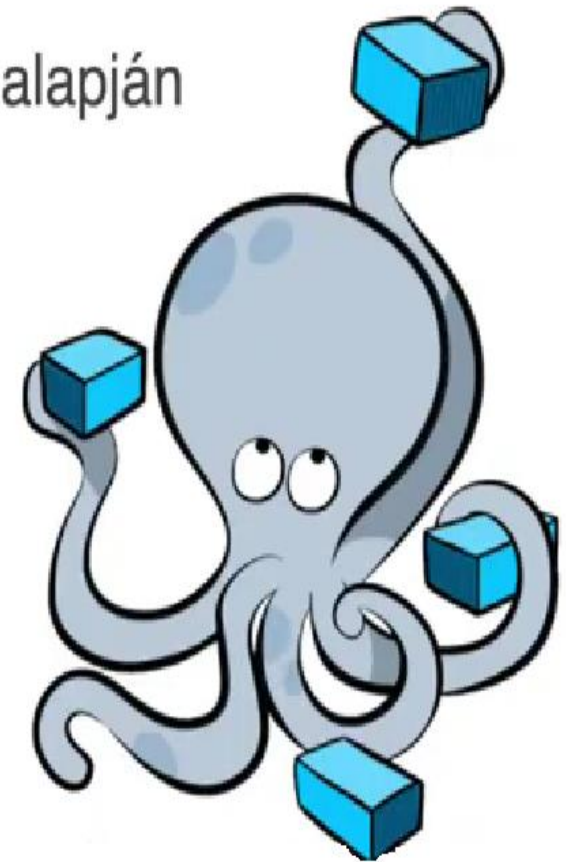
```
git clone https://github.com/docker/getting-started-app/ example-app
```



Több konténeres alkalmazások

Docker compose

- Több konténer futtatása egy ``docker-compose.yml`` leíró alapján
- Belső hálózat a szolgáltatások közötti kommunikációhoz
- Port továbbítás és fájlrendszer csatolása
- Egymásra épülő szolgáltatások kezelése



Kubernetes (k8s)

- pod-ok, telepítések, service-k
- skálázódás, leállásmentes telepítések
- hálózati szabályok, komplex hálózatkezelés
- több szerver összekapcsolása



docker-compose.yml

```
1 services:
2   app:
3     image: my-app
4     ports:
5       - "4000:3000"
6     environment:
7       MYSQL_HOST: mysql
8       MYSQL_USER: root
9       MYSQL_PASSWORD: secret
10      MYSQL_DB: itmp
11
12  mysql:
13    image: mysql:8.0
14    volumes:
15      - ./mysql-data:/var/lib/mysql
16    environment:
17      MYSQL_ROOT_PASSWORD: secret
18      MYSQL_DATABASE: itmp
```

Hasznos parancsok

- `docker-compose up`
- `docker-compose down`
- `docker-compose ps`
- `docker-compose logs`
- `docker-compose exec <szolgáltatás_neve> <parancs>`

docker-compose.yml

```
1 services:
2   app:
3     image: my-app
4     ports:
5       - "4000:3000"
6     environment:
7       MYSQL_HOST: mysql
8       MYSQL_USER: root
9       MYSQL_PASSWORD: secret
10      MYSQL_DB: itmp
11
12   mysql:
13     image: mysql:8.0
14     volumes:
15       - ./mysql-data:/var/lib/mysql
16     environment:
17       MYSQL_ROOT_PASSWORD: secret
18       MYSQL_DATABASE: itmp
```

3. gyakorlat

A Docker Compose használata

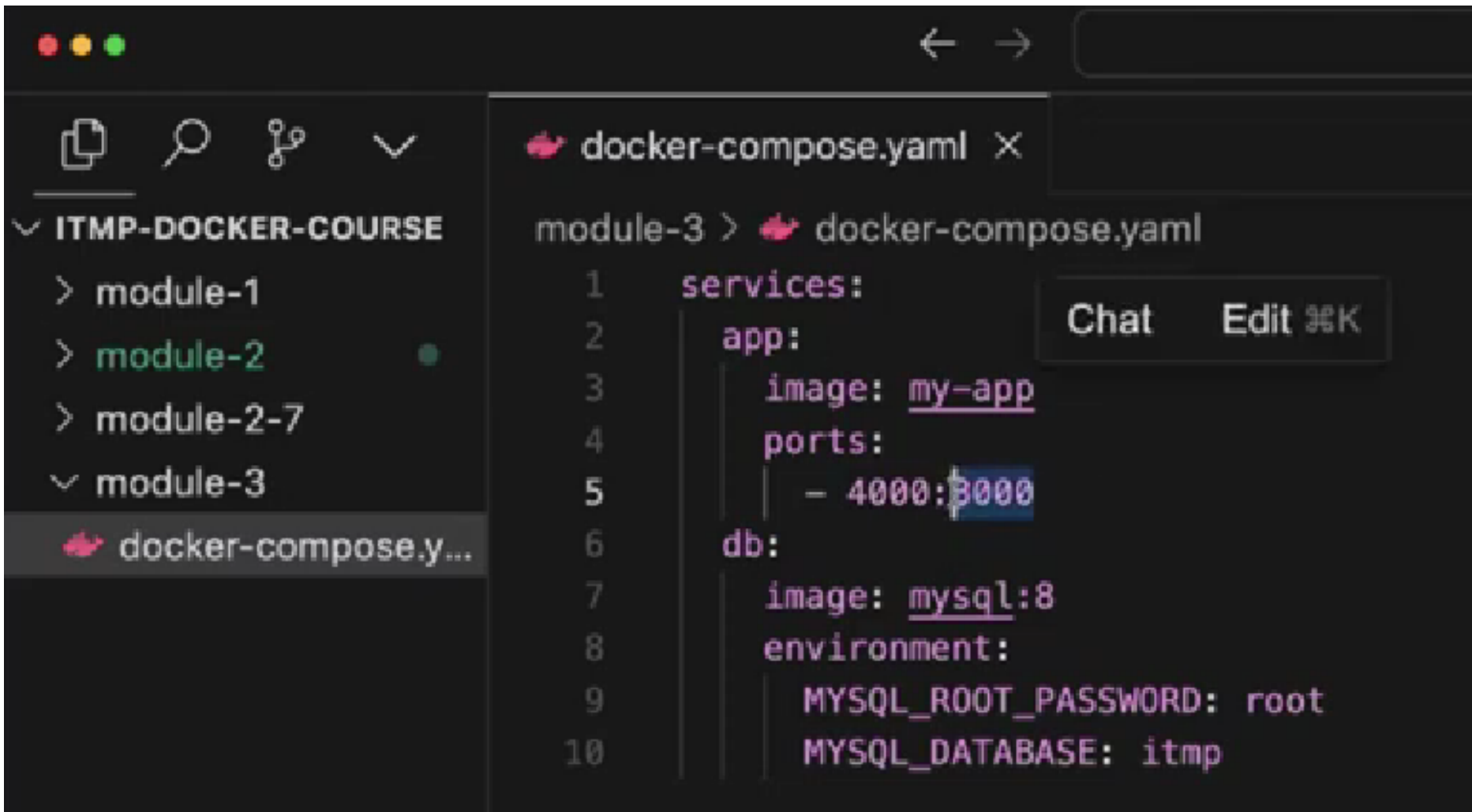
3. Alkalmazás konfigurálása Konfiguráld az alkalmazást környezeti változók segítségével, hogy az előző lépésben létrehozott adatbázist használja: `MYSQL_HOST`, `MYSQL_USER`, `MYSQL_PASSWORD`, `MYSQL_DB`.

Figyelj arra, hogy a belső hálózaton az adatbázist nem a localhost-on, hanem a service néven éred el, jelen esetben ez `mysql` lesz.

A teljes `docker-compose.yml` leíró ezen a ponton így néz ki:

```
services:
  app:
    image: my-app
    ports:
      - 127.0.0.1:3000:3000
    environment:
      MYSQL_HOST: mysql
      MYSQL_USER: root
      MYSQL_PASSWORD: secret
      MYSQL_DB: todos

  mysql:
    image: mysql:8.0
    volumes:
      - ./mysql-data:/var/lib/mysql
    environment:
      MYSQL_ROOT_PASSWORD: secret
      MYSQL_DATABASE: todos
```



KÖSZÖNJÜK A
FIGYELMET!