

# Docker image készítés

Állítsuk le az előző workshopon készített konténereket (docker stop...) és a hozzájuk tartozó apache site-okat is tiltsuk le:

```
kovi91@docker:~$ a2dissite phpmyadmin
kovi91@docker:~$ a2dissite wordpress
```

Készítsünk egy index.php fájlt a /root/www mappába

```
kovi91@docker:~$ nano /root/www/index.php
```

Tegyük bele az alábbi tartalmat

```
<?php
$notesFile = 'notes.txt';

if (!file_exists($notesFile)) {
    file_put_contents($notesFile, '');
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if (!empty($_POST['note'])) {
        $newNote = $_POST['note'] . "\n";
        file_put_contents($notesFile, $newNote, FILE_APPEND | LOCK_EX);
    }
}
$notes = file_get_contents($notesFile);
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Feljegyzések</title>
</head>
<body>

<h1>Feljegyzések</h1>

<form method="post" action="">
    <label for="note">Új feljegyzés:</label>
    <input type="text" id="note" name="note" required>
    <button type="submit">Hozzáadás</button>
</form>

<pre>
<?php echo $notes; ?>
</pre>

</body>
```

```
</html>
```

Készítünk most egy image-et, ami egy apache+php kombóból fog elindulni, hozzáadjuk ezt az 1 fájlt és így elkészült egy bárki által használható egyszerű TODO manager image! A root mappában hozzunk létre egy Dockerfile nevű fájlt!

```
FROM php:apache-bookworm
MAINTAINER Kovács András <kovacs.andras@nik.uni-obuda.hu>
COPY www/index.php /var/www/html/index.php
RUN chown -R www-data:www-data /var/www
```

Itt rengeteg RUN parancs lehetne, amelyek 1-1 dolgot módosítanak az image-ben. Ezek mind 1-1 layert adnak hozzá a konténerhez. Buildeljük le az image-et!

```
root@docker:~# docker image build -t kovi91/dlister:latest .
```

Lépünk be a docker hubra!

```
root@docker:~# docker login
```

Töltsük fel az image-et!

```
root@docker:~# docker image push kovi91/dlister:latest
```

Futtassuk a konténert!

```
root@docker:~# docker run -d --rm --name php -p 80:80
kovi91/dlister
```

## ASP.NET + MSSQL deployment

Tegyünk fel MS SQL szerveret dockerbe!

```
docker run --restart=always -e "ACCEPT_EULA=Y" -e
"MSSQL_SA_PASSWORD=Almafa123!!!" -p 1433:1433 --name mssql --
hostname mssql -d mcr.microsoft.com/mssql/server:2022-latest
```

Kapcsolódjunk rá a lokális ASP.NET környezetből a connection string módosításával, erre egy példa:

```
Server=mssql;Database=mvc;User Id=sa;Password=Almafa123;
```

A server után írjuk be a szerver publikus ip címét, azure-ben engedjük át a tűzfalon az 1433-as portot! Töltsük fel a táblákat a package manager konzolból:

```
Update-Database
```

Ezután írjuk vissza az ip címet mssql-re (ez ugye az mssql konténerünk neve!)

Tegyük fel az ASP projektet githubra, majd ott készítsünk egy automatizmust arra, hogyha jön a masterre push/pr. akkor készítsen egy új image-et a docker hubra.

Actions → new workflow → Docker image

YAML fájl így nézzen ki, sárgával jelölt részek a saját projektre cserélendőek.

```
on:
  push:
    branches:
      - "master"
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      -
        name: Checkout
        uses: actions/checkout@v3
      -
        name: Login to Docker Hub
        uses: docker/login-action@v2
        with:
          username: ${ secrets.DOCKER_HUB_USERNAME }
          password: ${ secrets.DOCKER_HUB_ACCESS_TOKEN }
      -
        name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v2
      -
        name: Build and push
        uses: docker/build-push-action@v3
        with:
          context: .
          file: ./hellomvc/Dockerfile
          push: true
          tags: ${ secrets.DOCKER_HUB_USERNAME }/hellomvc:latest
```

A secretekhez az értékek betöltése: adott repó → settings → secrets and variables → actions → new

A docker hubon kell az access token-t legenerálni: account settings → security  
→ access tokens → new

Az ASP konténer futtatása kb. ugyanúgy történik, mint az első workshopon a  
wordpress, usernév/konténer-név docker hubról, csatolni kell az mssql  
konténert.

Fusson valamilyen porton, majd jöhet rá az apache virtualhost készítés + cert  
igénylés + vhost bővítés.