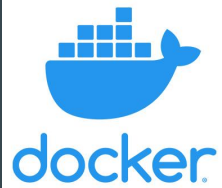
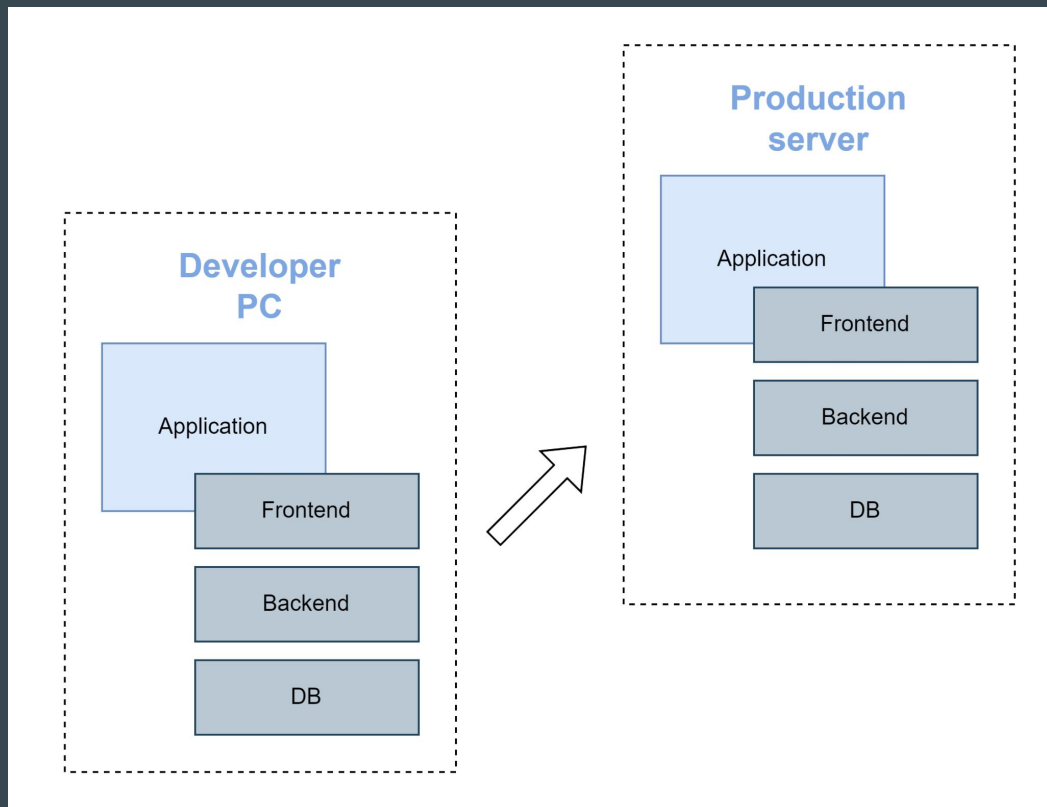


Docker

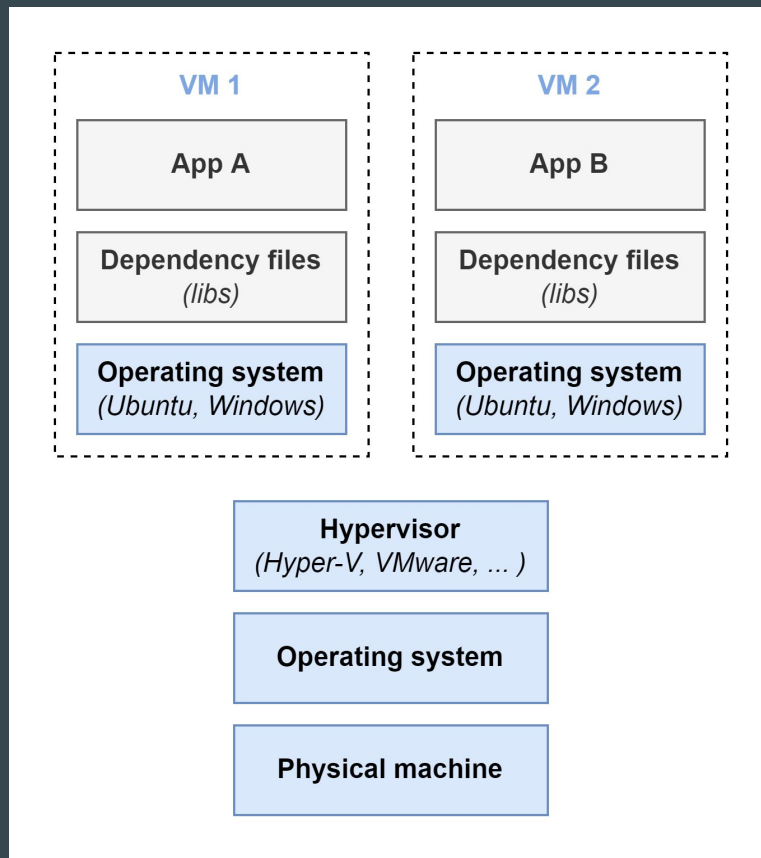


- Prečo docker ?
- Architektúra
- Demo

Aký problém chceme riešiť ?

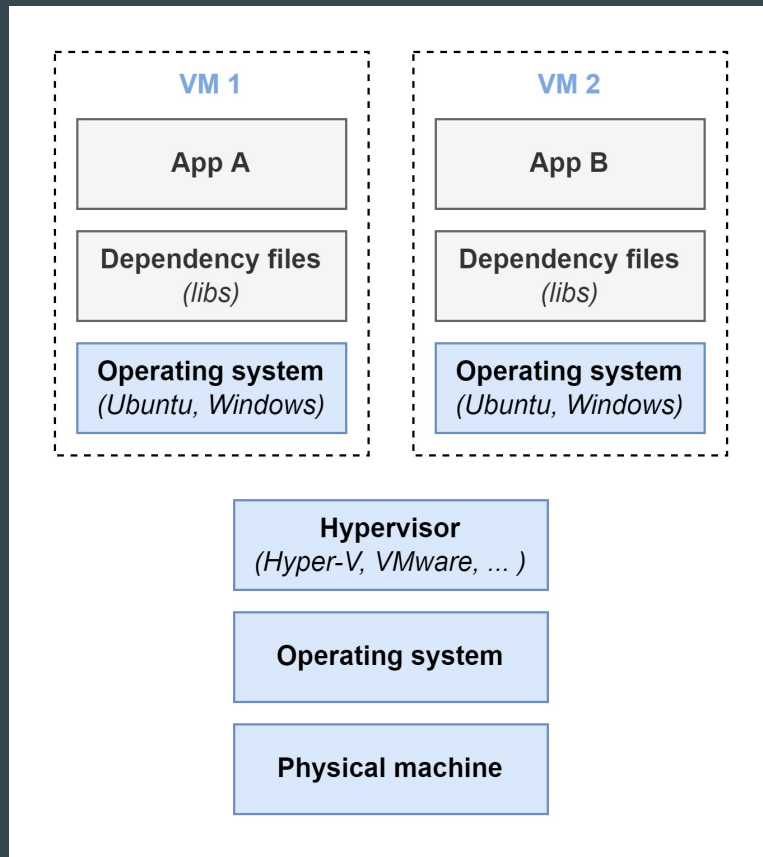


VM - virtuálny stroj



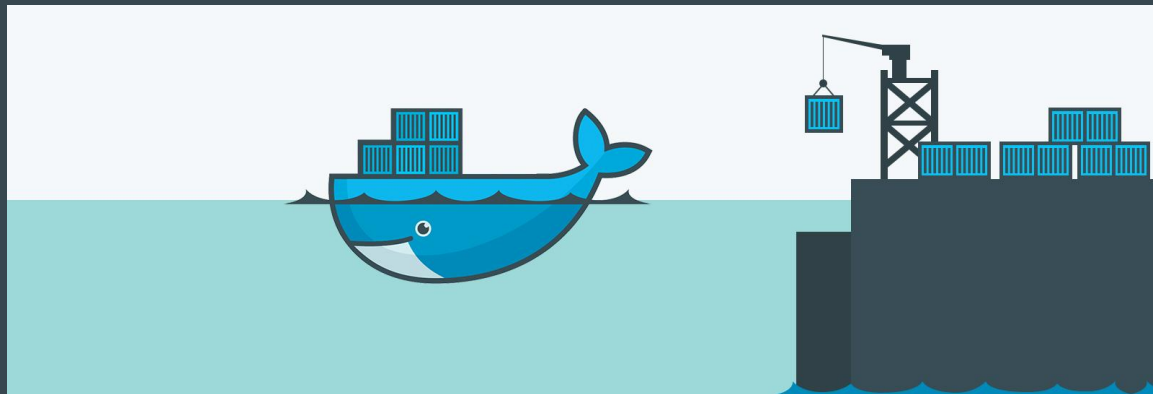
VM - virtuálny stroj

- 1 služba - jeden VM
- N služieb - jeden VM
- Pre každú službu jeden VM



Docker

- Nástroj na kontajnerizáciu
- Docker je iba jedna z možností ako riešiť problém kontajnerov
- Kontajner je abstrakcia v operačnom systéme
- Vytváranie, spúšťanie a manažovanie kontajnerov
- Windows, Linux, ...

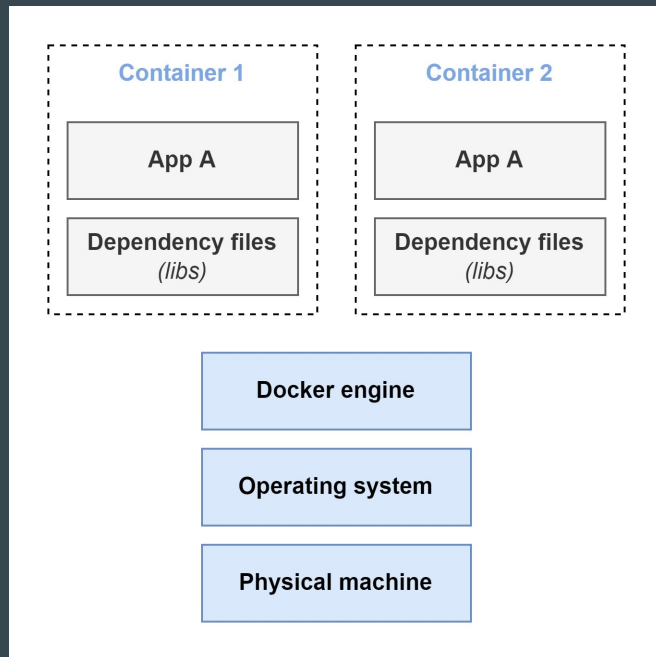


Docker Kontajner

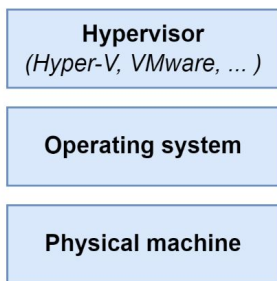
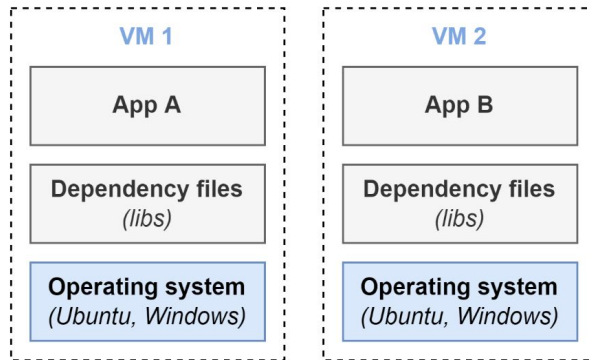
- Kontajner obsahuje všetko potrebné pre beh danej služby
- Pravidlo: jedna služba, jeden kontajner

Princípy kontajnerizácie:

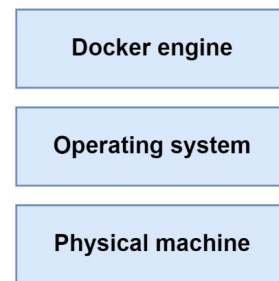
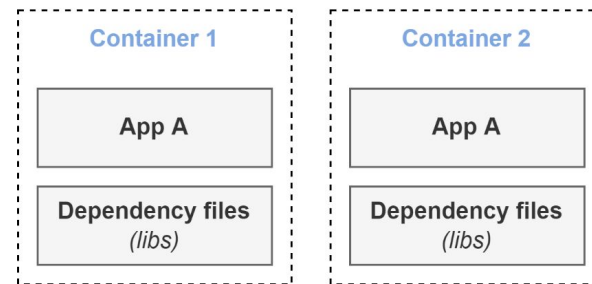
- Štandardnosť (Standard)
- Jednoduchosť (Lightweight)
- Izolovanosť (Isolated)



Porovnanie

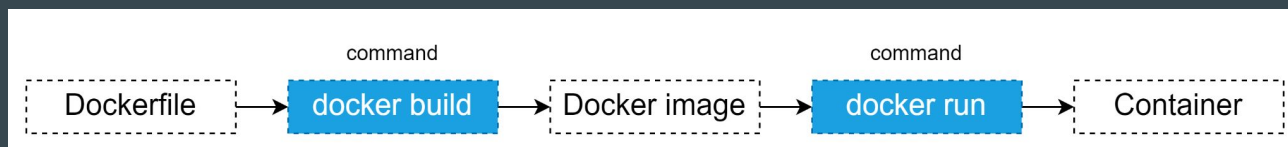
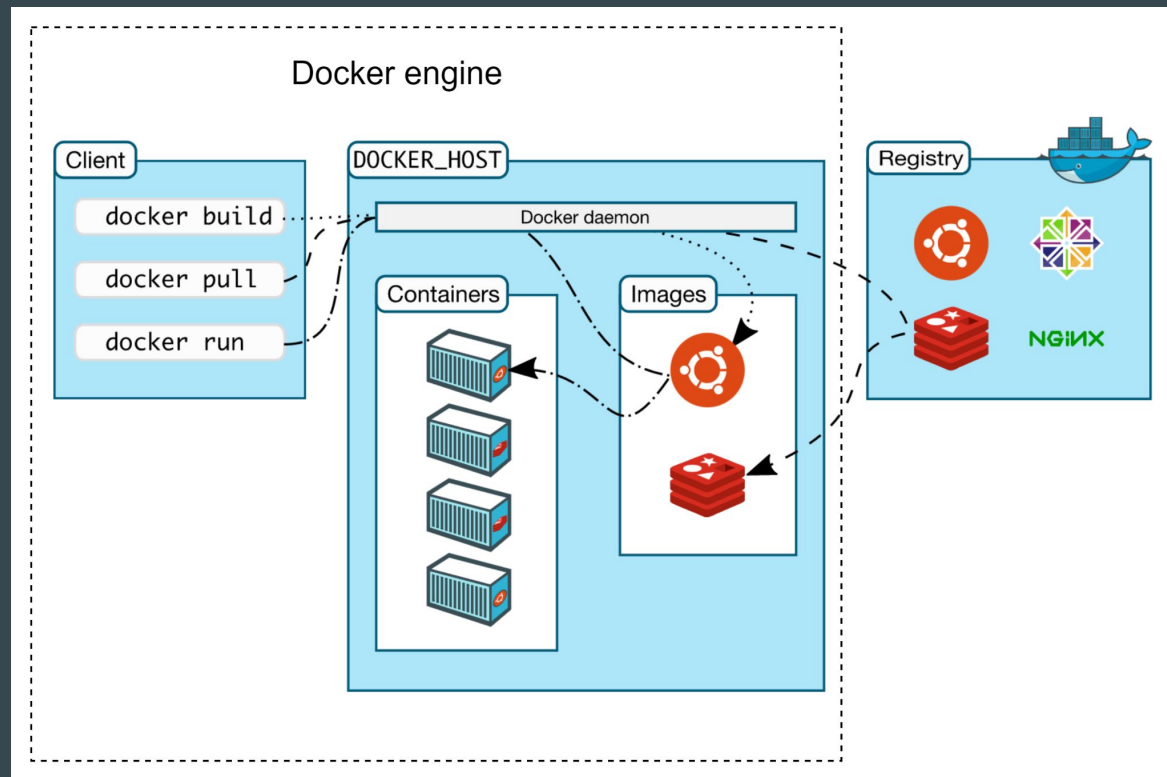


VM



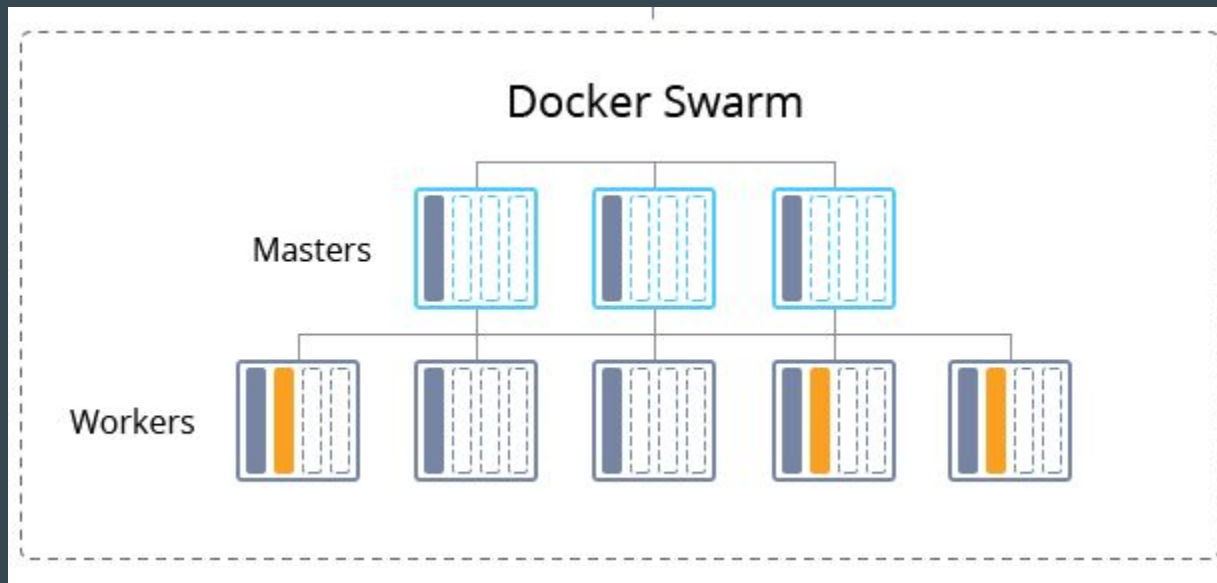
Containers

Architektúra



Docker swarm

- Spravovanie viacerých inštancií Dockeru
- Zabezpečuje orchestráciu prostredia



Demo

aspnetcoreapp [Home](#) [Privacy](#)

Welcome

Learn about [building Web apps with ASP.NET Core](#).

Dockerfile

Dockerfile > ...

```
1  # Inicializacia noveho buildu zo zadaneho imagu
2  FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build-env
3  # Nastavenie pracovneho adresara na /app
4  WORKDIR /app
5
6  # Skopirovanie csproj suboru do danej cesty
7  COPY *.csproj ./
8  # Spustenie prikazu dotnet restore
9  RUN dotnet restore
10
11 # Skopirovanie vsetkych suborov do app ktory sa nachadza v kontajnery
12 COPY . /app
13 # Spustenie prikazu dotnet publish -c Release -o out
14 RUN dotnet publish -c Release -o out
15
16 FROM mcr.microsoft.com/dotnet/aspnet:5.0
17 WORKDIR /app
18 COPY --from=build-env /app/out .
19 ENTRYPOINT ["dotnet", "aspnetcoreapp.dll"]
```

`docker build -t asp_net_core_app_image .`

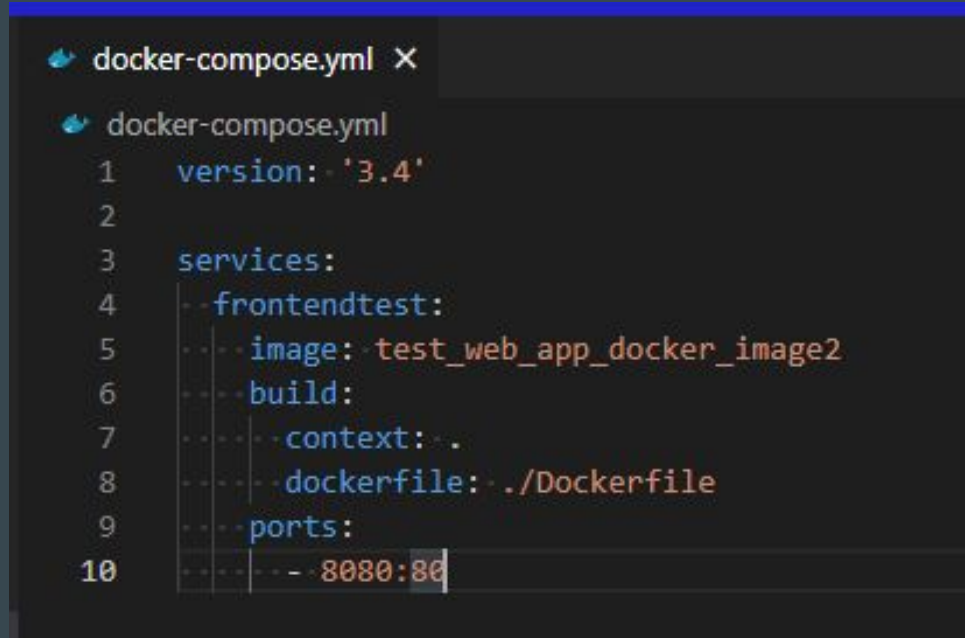
Spustenie kontajneru

1. Cez príkaz docker run

```
docker run -d -p 8080:80 --name asp_net_core_app asp_net_core_app_image
```

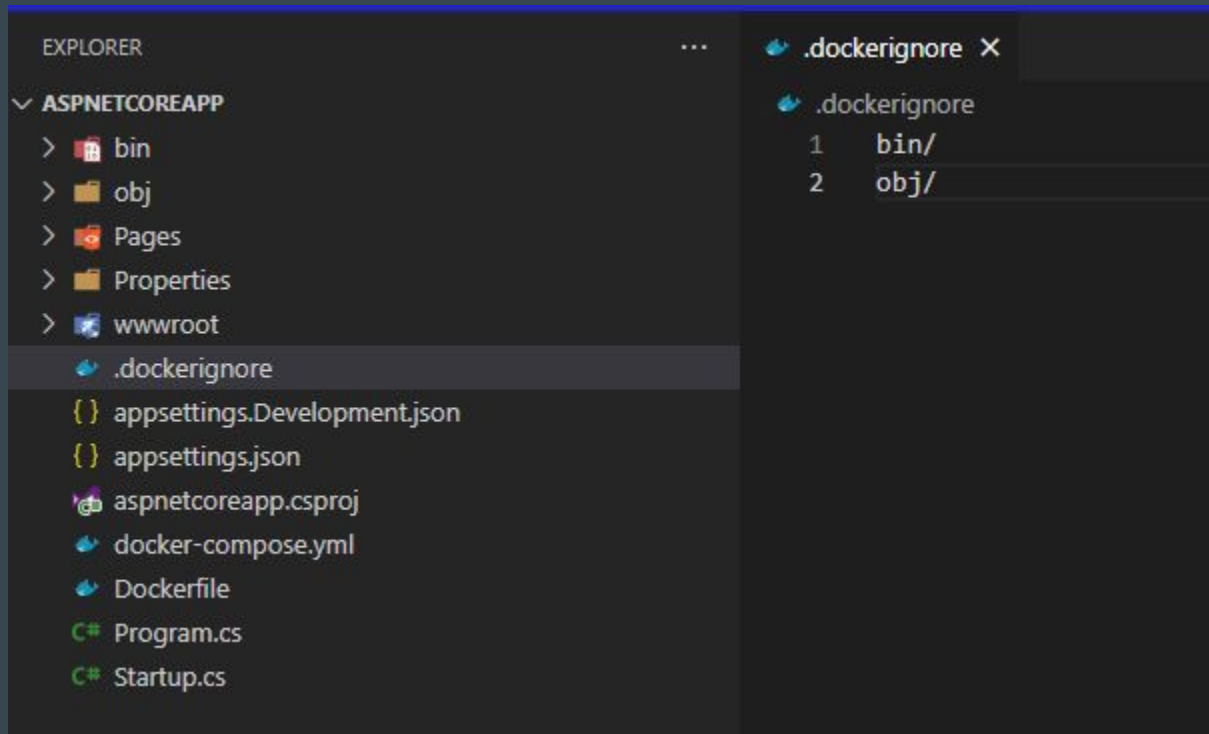
2. Cez docker-compose

```
docker-compose up -d
```

A screenshot of a code editor with a dark theme. The editor has a tab at the top labeled 'docker-compose.yml' with a close button. The main area shows the content of the file, which is a YAML configuration for Docker Compose. The text is as follows:

```
1  version: '3.4'
2
3  services:
4    - frontendtest:
5      image: test_web_app_docker_image2
6      build:
7        context: .
8        dockerfile: ./Dockerfile
9      ports:
10     - 8080:80
```

.dockerignore



Užitočné príkazy

- **docker ps -a**
 - zobrazenie všetkých kontajnerov
- **docker logs [id]**
 - zobrazenie logu z daného kontajneru
- **docker image ls**
 - zobrazenie imagov
- **docker rmi [id]**
 - vymazanie imagu

Prečo docker použiť ?

- Reprodukovanosť
- Izolácia
- Rýchlosť
- Bezpečnosť

Linky

- GitHub:
 - <https://github.com/SpurnyMarek/docker>
- Docker Documentation:
 - <https://docs.docker.com/>