

Webový server

Jiří Zacpal



KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

KMIWEBA Webové aplikace

Zprovoznění webového serveru a dalších služeb



- Webový server je software, který zprostředkovává webovou stránku klientům, kteří o ni požádají pomocí HTTP požadavku.
- Existuje celá řada webových serverů. Mezi ty nejznámější patří: Apache, IIS, nginx, node.js (později uvidíme, že node.js je něco víc než jen webový server).
- Zprovoznit webový server je poměrně snadné. Stačí jej nainstalovat a spustit. Pokud chceme na webovém serveru provozovat statické webové stránky, jednoduše je umístíme soubory (HTML, CSS, JS) na určené místo a máme hotovo. V případě, že chceme provozovat dynamické webové stránky, musíme k webovému serveru přidat server aplikační (zabezpečující vykonávání programového kódu, který stránku sestavuje) a případně databázový (sloužící pro perzistentní uložení dat). V principu se opět nejedná o nic složitého.

Zprovoznění webového serveru a dalších služeb



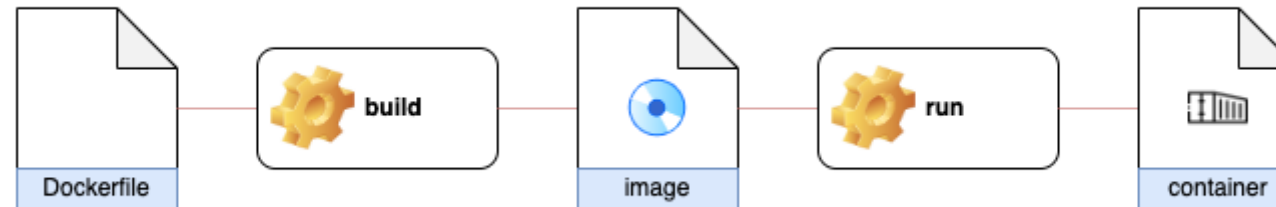
- V následujících seminářích si mimo jiné ukážeme tvorbu dynamických stránek s využitím jazyka PHP a databáze MySQL. Nyní si ukážeme, jak připravit vše potřebné.
- Ve spojení s výše uvedenými technologiemi se obvykle využívá webový server Apache. Standardně je tato trojice součástí tzv. LAMP balíčků. Podle OS jsou nejběžnější: WAMP, XAMPP, MAMP. Případně je možné nainstalovat jednotlivé služby samostatně.
- Nevýhodou tohoto řešení je, že instalace je společná pro všechny vyvíjené projekty. Navíc je obvykle třeba služby konfigurovat a přenesení této konfigurace, nemluvě o problémech s různými verzemi software (či knihoven), na jiný stroj může být obtížné. Lepším způsobem je využít při nasazení webové stránky/aplikace kontejnerizaci.

- Docker je dnes běžně používaná technologie pro kontejnerizaci. Kontejner představuje izolované prostředí, ve kterém běží (ne nutně webová) aplikace. Přesněji řečeno v kontejneru běží (docker) image, který aplikaci obsahuje.

Instalace

- Instalace je popsána na stránkách docs.docker.com.
- Aplikace Docker desktop obsahuje vše potřebné pro práci s kontejnery.
- Kontrolu instalace je možné provést následujícím příkazem
docker version
je třeba ověřit, že běží klient i server a verze je větší nebo stejná jako 20.10.17

Docker – základní pojmy



- **Dockerfile** - je textový soubor, ve kterém píšeme postup, jak se mají vytvářet images (obrazy). Popisujeme zde všechny kroky, které je potřeba vykonat. Zároveň zde můžeme provést i nutná nastavení pro hladký běh naší aplikace.
- **Build** - je proces pro vytvoření obrazu z Dockerfile. Proces se spouští příkazem `docker build`, který má několik volitelných argumentů.
- **Image** - Ze souboru Dockerfile si pomocí procesu build vytvoříme image, což je soubor používaný ke spuštění kódu v kontejneru. Docker image je vytvořený tak, aby se dal „spustit“ na jakémkoliv počítači, kde je nainstalovaný program Docker. Image si můžeme připodobnit k datovému nosiči (například CD, DVD nebo flash disk).
- **Run** - je proces, kterým se spustí instance z image. Výhoda Dockeru je, že z jednoho image můžeme spustit prakticky neomezené množství kontejnerů. Proces spouštíme příkazem `docker run`.
- **Container** - jsou balíčky softwaru, které obsahují všechny nezbytné prvky pro běh našeho programu v jakémkoli prostředí. Docker kontejner je instancí Docker image ve kterém se spustí nějaký (definovaný) proces. Kontejner má svůj vlastní životní cyklus a může se nacházet v různých stavech.

Příklad – 1. Dockerfile



- Vytvořte si adresář `docker` a v něm adresář `public-html` (do něj s umístěte soubor `index.html` s nějakým html kódem).
- Vytvořte textový soubor `Dockerfile.yml` (ve windows by to měl být soubor bez přípony).
- Soubor bude obsahovat následující příkaz:

```
FROM httpd:2.4
```

```
COPY ./public-html/ /usr/local/apache2/htdocs/
```

- Pomocí příkazu `FROM` definujeme, co se použije jako výchozí image.
- Všechny obrazy najdete na stránkách Dockerhubu (<https://hub.docker.com/>)
- Jelikož vytváříme webový server, vybrali jsme si image `httpd`.
- Image si můžete stáhnout příkazem `docker pull httpd`
- Formát příkazu `FROM` se skládá ze dvou částí, ve formátu `obraz:verze`.
- Pokud verzi necháte prázdnou, automaticky se stáhne obraz s tagem `latest`.
- Většina imagů má jasně definovanou strukturu tagu, kterou si můžete přečíst na Dockerhub.

Příklad – 2. Vytvoření kontejneru

- Následně vytvoříme image následujícím příkazem:
`docker build -t my-apache .`
- Ted' máme vytvořenou image s názvem `my-apache` a může vytvořit kontejner:
`docker run -d --name my-running-apache -p 8080:80 my-apache`
- Vytvořený a spuštěný kontejner běží na portu 8080:
<http://localhost:8080>
- Ukončíme jej pomocí CTRL + C. Případně je možné spustit kontejner na pozadí pomocí parametru -d. Přehled kontejnerů je možné zobrazit pomocí příkazu:
`docker container ls`

Využití existujícího image



- Jednou z možností, jak vytvořit image obsahující webový, aplikační a databázový server, je využít existující image, například ubuntu obsahující jádro OS a do něj doinstalovat potřebné náležitosti. Následující konfigurační soubor ukazuje instalaci a spuštění Apache.

```
FROM ubuntu:latest
```

```
RUN apt update
```

```
RUN apt -y install apache2
```

```
RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf
```

```
CMD ["apachectl", "-D", "FOREGROUND"]
```

- Nevýhodou uvedeného způsobu ale monolitická architektura (vše v jednom kontejneru).
- Zajímavější alternativou je vytvořit několik spolupracujících kontejnerů. K tomuto slouží nástroj docker-compose.

- Nástroj docker-compose umožňuje vytvořit několik spolupracujících kontejnerů. Opět je nejprve potřeba vytvořit a nakonfigurovat image, případně využít existující.
- Postup sestavení všech image je specifikován v souboru docker-compose.yml. Výhodou docker-compose je možnost specifikování všech parametrů přímo v konfiguračním souboru.
- Spouštění kontejnerů je tedy jednodušší.
- Nyní si vytvoříme komplexní prostředí obsahující webový server Apache, podporu jazyka PHP, MySQL databázi a administrační rozhraní pro databázi. Pro databázi a její administraci nám stačí existující nezměněné image.
- Pro webový server, který bude podporovat PHP a MySQL databázi, musíme modifikovat výchozí image. Následuje obsah docker-compose.yml.

Soubor docker-compose

version: "3.9"

services:

Apache a PHP

www:

build: ./docker/php-apache # cesta k Dockerfile

volumes:

- ./src:/var/www/html/

ports:

- 8080:80

Soubor docker-compose



MySQL databáze

db:

image: mysql:latest

restart: always

environment:

MYSQL_ROOT_PASSWORD: toor # nastavení hesla pro uživatele root

administrace databáze

adminer:

image: adminer:latest

restart: always

ports:

- 8081:8080

docker-compose

- Dockerfile umístěný ve složce ./docker/php-apache vypadá následovně.

```
FROM php:8.1-apache
```

```
RUN docker-php-ext-install mysqli && docker-php-ext-enable mysqli
```

Image sestavíme a spustíme příkazy.

- sestavení

```
docker-compose build
```

- spuštění

```
docker-compose up
```

Případně je možné obojí spustit současně `docker-compose up --build`.

- Po spuštění běží webový server na portu 8080 a administrační rozhraní na portu 8081. Konfigurační soubory a adresářová struktura je předpřipravena v souboru `docker-web-server.zip`.

ÚKOL 1

Spustěte nebo naistalujte některý z LAMP balíčků. Na školních počítačích je nainstalován XAMPP. Vytvořte HTML soubor a spustě jej na webové serveru.

ÚKOL 2

Zprovozněte webový server Apache, s podporou PHP a MySQL v Docker kontejnerech.