SofaScore PHP akademija
10. Predavanje

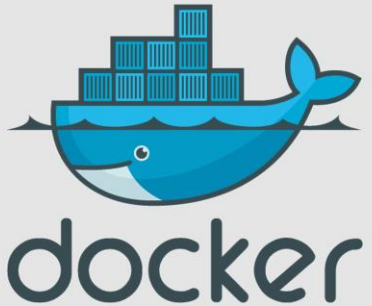
20. svibnja 2021.

Alen Murtić

Sadržaj

- Docker i nginx
- PostgreSQL
- DataGrip - praktična primjena aplikacije
- Doctrine i Symfony – nastavak prethodnog predavanja
- Redis baza podataka

Docker



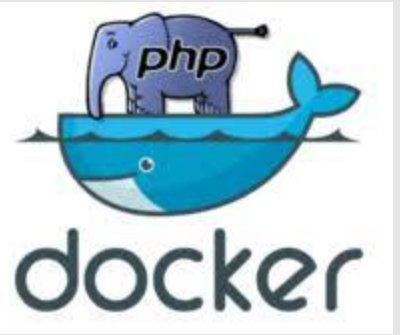
- Program za virtualizaciju softwarea u samostalne *containere*
  - Container - "poput virtualne mašine samo za jednu aplikaciju" [1]
  - Preporuka: jedan servis u jednom containeru [2]
    - Servisi: npr. HTTP server, baza podataka, PHP aplikacija
  - Servisi se mogu replicirati
- Jednako pogodan za razvoj i produkciju aplikacije
  - Brzo kreiranje okruženja u kojem aplikacija treba raditi
  - Rad aplikacije ne ovisi o okružnju u kojem radi Docker jer se ona sama vrti u containeru
  - Puno alata koji organiziraju containere u produkciji, najpopularniji Kubernetes [3]

[1] https://opensource.com/resources/what-are-linux-containers?intcmp=7016000000127cYAAQ

[2] https://docs.docker.com/config/containers/multi-service_container/

[3] https://kubernetes.io/

Docker i PHP



- Docker compose
  - Alat za definiranje i pokretanje Docker aplikacija s više containera
  - Definicija u čitljivoj YAML datoteci
- Kako pronaći/saznati željene definicije servisa?
  - ~~Googlanjem github repozitorija koji imaju konfiguracije~~
  - ~~Čitanje Medium članaka~~ (oni mogu biti korisni, ali ne treba c/p-ati njihove yaml definicije)
  - https://phpdocker.io/generator
    - Aplikacija koja stvara docker-compose konfiguracije, dockerfile konfiguracije za pojedini odabrani servis te kratku *cheat-sheet* datoteku osnovnih naredbi
    - Nudi odabir dodatnih, korisnih servisa poput različitih baza podataka, web servera, …
    - Napomena: nije ažuriran za PHP8, nadajmo se da će biti

Docker compose – primjer YAML datoteke

- Sadrži verziju docker composea I servise
  - Redis baza podataka
  - Postgres baza podataka
  - Nginx web server
  - PHP-FPM sa composerom
    - PHP verzija 7.4
- Vrlo jednostavno pokretanje i integracija symfonyja sa svim navedenim servisima

```yaml
##############################################################################
#                          Generated on phpdocker.io                         #
##############################################################################
version: "3.1"
services:

    redis:
      image: redis:alpine
      container_name: academy-redis

    postgres:
      image: postgres:11.1-alpine
      container_name: academy-postgres
      working_dir: /application
      volumes:
        - .:/application
      environment:
        - POSTGRES_USER=admin
        - POSTGRES_PASSWORD=admin
        - POSTGRES_DB=Academy
      ports:
        - "8769:5432"

    webserver:
      image: nginx:alpine
      container_name: academy-webserver
      working_dir: /application
      volumes:
          - .:/application
          - ./phpdocker/nginx/nginx.conf:/etc/nginx/conf.d/default.conf
      ports:
       - "8765:80"

    php-fpm:
      build: phpdocker/php-fpm
      container_name: academy-php-fpm
      working_dir: /application
      volumes:
        - .:/application
        - ./phpdocker/php-fpm/php-ini-overrides.ini:/etc/php/7.4/fpm/conf.d/99-overrides.ini
```

Pokretanje docker composea

- Preduvjet: instalacija Docker programa
- Pokretanje konzolnom naredbom **docker-compose up**
  - Dodatni argument **–d** za pokretanje kao pozadinski proces
- Naredba **docker ps** za ispis pokrenutih containera:

```
murta@alens-MBP-2 academy % docker ps
CONTAINER ID        IMAGE                 COMMAND                  CREATED             STATUS              PORTS                    NAMES
04a33d75dc65        academy_php-fpm       "/usr/sbin/php-fpm7.…"   32 hours ago        Up 3 seconds        9000/tcp                 academy-php-fpm
3551e75cdb69        nginx:alpine          "nginx -g 'daemon of…"   32 hours ago        Up 3 seconds        0.0.0.0:8765->80/tcp     academy-webserver
9bd7d4d3a37f        redis:alpine          "docker-entrypoint.s…"   32 hours ago        Up 3 seconds        6379/tcp                 academy-redis
339e427aa502        postgres:11.1-alpine  "docker-entrypoint.s…"   32 hours ago        Up 3 seconds        0.0.0.0:8769->5432/tcp   academy-postgres
```

- Pristup konzoli php-fpm containera naredbom:
  - **docker-compose exec php-fpm bash**
  - Sad možemo inicijalizirati symfony projekt composer naredbom

Nginx

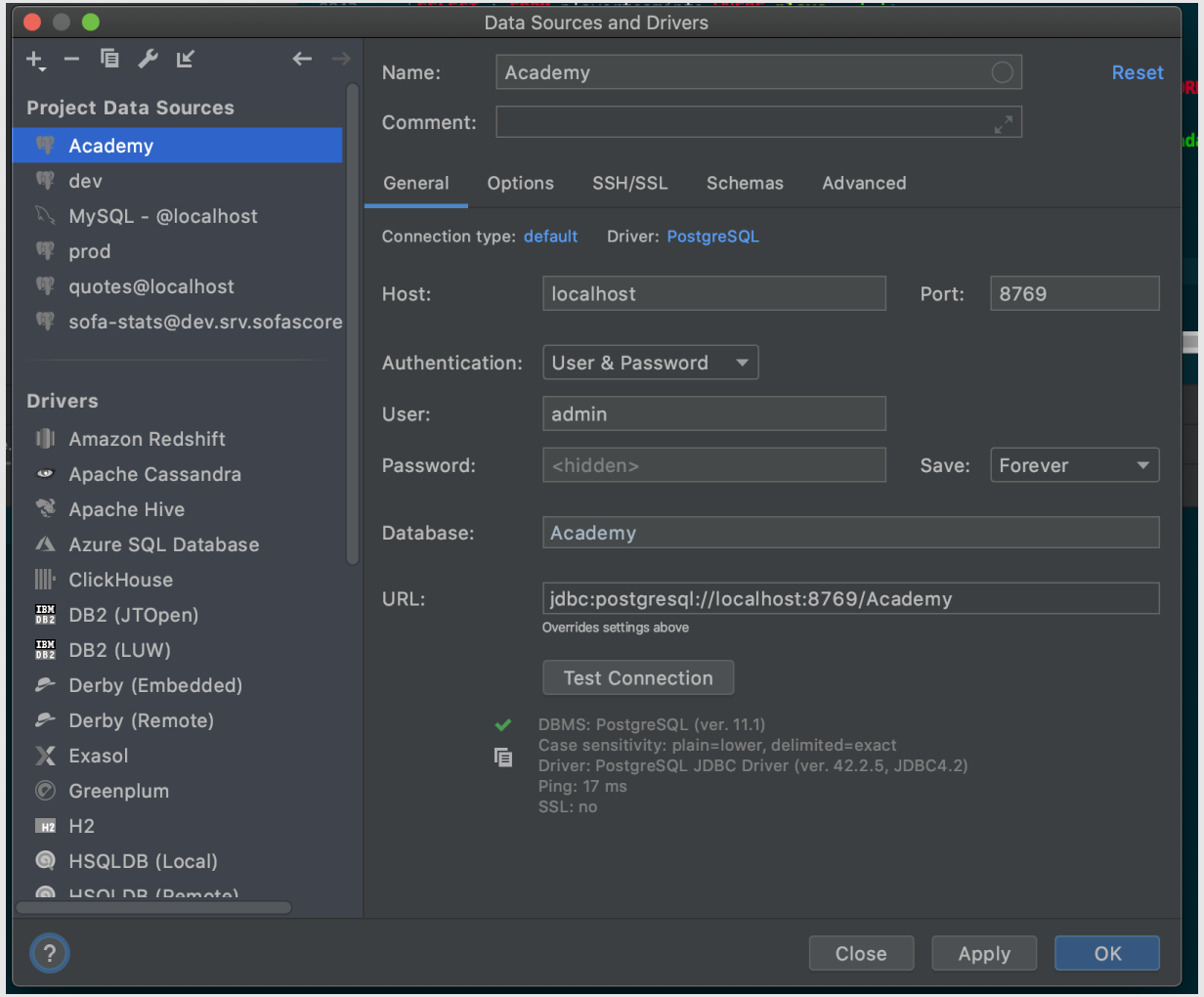
- Iznimno moćan alat koji je počeo samo kao web server, ali u međuvremenu je dobio dodatne značajke poput proxy
- Detaljnije:
  - Kako je nastao? https://www.nginx.com/resources/glossary/nginx/
  - Koje funkcionalnosti ima? https://nginx.org/en/
- Konfiguracija u nginx.conf datoteci
  - Može je generirati https://phpdocker.io/generator
- Detaljnije mogu istražiti oni koji žele znati više, u sklopu ove edukacije ćemo pretpostaviti da nam nginx magično radi

PostgreSQL

- Besplatan sustav za upravljanje relacijskim bazama podataka (RDBMS) otvorenog koda
- Cilja na implementaciju punog SQL standarda
  - Što to znači za korisnika? Jako puno značajki
  - Bolje se skalira u odnosu na MySQL u pogledu količine podataka i upita
- CAP teorem (consistency, availability, partition tolerance) - CA fokus
- Osnovna funkcionalnost slična s već naučenim MySQL-om

DataGrip

- IDE (integrirana razvojna okolina) za pristup bazama podataka
- Besplatan za studente
- Primjer konfiguracije za pristup bazi pokrenutoj na slajdu 6:



Doctrine i PostgreSQL - uvod

- Doctrine pruža potpunu podršku korištenju osnovnih značajki PostgreSQL baze podataka, ali neke naprednije ili specifičnije (u odnosu na druge RDBMS-ove) značajke nisu podržane
- Ali, Doctrine podržava ekstenzije koje je već "netko na internetu" implementirao i mogu se dodati u projekt, npr. https://github.com/martin-georgiev/postgresql-for-doctrine ili https://github.com/opensoft/doctrine-postgres-types, ili čak napisati vlastite

Doctrine i PostgreSQL - povezivanje

- Kod stvaranja website sekelton projekta sa Symfony.com, u config/packages direktoriju je stvorena doctrine.yaml datoteka u kojoj postoji doctrine: dbal: url: zapis čija je inicijalna vrijednost: '%env(resolve:DATABASE_URL)%'
  - To znači da u *.env* datoteci postoji DATABASE_URL vrijednost na koju se referencira ovaj parametar, a ona je incijalno postavljena za MySQL
  - Za bazu sa slajda 6, potrebno je postaviti sljedeću vrijednost:
    - DATABASE_URL=postgresql://admin:admin@academy-postgres:5432/Academy?serverVersion=11&charset=utf8
  - Korisnik iz DataGripa na računalu koje je Docker poslužitelj bazi može pristupiti 127.0.0.1:5432 (localhost) adresom, ali Symfony aplikacija može samo simboličkom academy-postgres:5432 adresom
  - To je jedina stvar koju je potrebno promijeniti da aplikacija koristi željenu bazu

Doctrine – embedded (ugrađeni) entiteti

- Konkretni problemi iz SofaScore domene:
  - Igrač, klub, sudac, trener, stadion i utrka trebaju informaciju o svojoj državi
    - Podatak treba biti spremljen u svaku od tablica na efikasan i razumljiv način
    - Država ima zastavu, ime, IOC kod, ISO kod, kontitent, …, idealno je spremati jedan od kodova, a ostale podatke računati na temelju njega. Treba li copy-pasteati logiku za pretvorbu na svaki od entiteta?
  - Utakmica ima rezultate po poluvremenima, četvrtinama, produžecima, izmjenama (inning) te konačan rezultat za svaki od timova
    - Ako to spremamo u Score entitet, a utakmica ima dvije jedan-na-jedan (OneToOne) poveznice na Score za domaći i gostujući tim, to znači da kod svakog čitanja utakmice iz baze moramo obaviti dva spajanja sa Score tablicom (iako je to indeksirano spajanje, svejedno imamo pad efikasnosti)
    - Možemo copy-pasteati svaki atribut s nazivima poput homeScoreHalftime i awayScoreHalftime
- Rješenje su embedded entiteti

Doctrine – embedded (ugrađeni) entiteti

- Embedded entiteti u Doctrineu su PHP objekti čiji su atributi *ugrađeni dio* u tablice entiteta koji ih sadrži
- Oni imaju atribute definirane uobičajenom @ORM\Column(… anotacijom [1], a umjesto @ORM\Entity imaju @ORM\Embeddable()
- Rješenje drugog problema s prošlog slajda:
  - Napraviti Score klasu koja je embedded i ima atribute halftime, overtime, final, period1, period2, …
  - U klasi koja predstavlja utakmicu, dodati dva atributa homescore i awayscore koji su tipa Score
  - Doctrine će generirati atribute poput homescore_halftime u tablici utakmice u bazi

- [1] https://www.doctrine-project.org/projects/doctrine-orm/en/2.7/tutorials/embeddables.html

Doctrine - nasljeđivanje

- U objektno orijentiranom programiranju jedan od osnovnih obrazaca modeliranja stvarnog svijeta je nasljeđivanje
- Doctrine podržava dva tipa nasljeđivanja:
  - SINGLE TABLE - svi entiteti se spremaju u jednu tablicu koja ima stupac "discriminator column" kojim se određuje klasa/tip entiteta
    - Pogodno za situaciju kada klase imaju jako puno zajedničkih atributa, a razlikuju se u tek nekoliko ili samo po ponašanju u PHP-u
  - CLASS TABLE – svaki tip entiteta se sprema u zasebnu tablicu
    - Načelno jako slično copy-pastanju atributa, samo bolje 😂

Doctrine i Symfony - zašto?

- Punjenje objekata podacima iz baze dodaje sloj aplikaciji koji usporava njen rad, a istovremeno stvara novu točku pucanja
- Ako zanemarimo modeliranje baze programskim kodom i rad s objektima umjesto PHP polja dohvaćenih iz baze, postoji li opravdan razlog korišenja Doctrinea?
  - Lakša migracija na neku drugu bazu, npr. MySQL -> PostgreSQL, ako je većina upita pisana query builderom
  - Doctrine prati promjene nad entitetom
    - Ne stvara SQL upite ako je aplikacija pokušala postaviti vrijednost atributa na onu koja je već postavljena
    - Slušanje promjena nad entitetom
  - ORM cache

Redis

- Red poslova zahtijeva komunikaciju između više procesa
- Redis je brza i jednostavna NoSQL baza koja podatke drži u memoriji
  - Opcionalna trajnost podataka
  - **Ne treba se koristiti umjesto relacijskih baza, nego uz njih**
  - Može se koristiti i kao cahce, npr. HTTP cache Ili cache podataka s diska
  - Osnova rada: spremanje podataka u obliku ključ => vrijednost
- Ideja korištenja:
  - dinamički flagovi koje treba brzo provjeriti
  - kratkotrajno zanimljvi (privremeni) podaci
  - JSON vrijednosti prema ključu
    - JavaScript Object Notation - čitljiv način prenošenja podataka
  - dijeljenje podataka između različitih procesa na brz i jednostavan način

Symfony servisi i dependency injection

- Servis je način organizacije koda
- Symfony je počeo koristeći ServiceContainer koncept, ali novije verzije forsiraju dependency injection
  - Servisi se ne mogu dohvatiti iz containera, oni su privatni
  - Ovisnost jednog servisa o drugom definira kroz konstruktor
  - U services.yml datoteci treba postaviti sljedeće parametre:

```yaml
services:
    _defaults:
        autowire: true      # Automatically injects dependencies in your services.
        autoconfigure: true # Automatically registers your services as commands, event subscribers, etc.
        public: false       # Allows optimizing the container by removing unused services; this also means
                            # fetching services directly from the container via $container->get() won't work.
                            # The best practice is to be explicit about your dependencies anyway.
```

- Pojedini servisi ipak mogu biti javni, ali za njih eksplicitno treba definirati public: true

11. predavanje

- Koncept cachea
  - HTTP caching
  - ORM caching
- Red poslova (job queue)
  - Slušanje promjena nad entitetom
  - Izrada jednostavne verzije uz pomoć Doctrinea i Redisa
- Testiranje

Hvala na pozornosti!

- Sljedeće predavanje je na rasporedu 27. svibnja 2021.

- Alen Murtić, alen.murtic@sofascore.com
- Karlo Knežević, knezevic.karlo1@gmail.com