



AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
Wydział Fizyki i Informatyki Stosowanej

Praca Magisterska

Sylwester Macura

kierunek studiów: **Informatyka Stosowana**

Wykorzystanie komponentów projektu Spring w Systemie Zarządzania Treścią

Opiekun: **dr inż. Barbara Kawecka-Magiera**

Kraków, Lipiec 2016

Oświadczam, świadomy odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem osobiście i samodzielnie i nie korzystałem ze źródeł innych niż wymienione w pracy. ·

.....

(czytelny podpis)

Merytoryczna ocena pracy przez opiekuna

Końcowa ocena pracy przez opiekuna:

Data:

Podpis:

Skala ocen: (6.0 – celująca), 5.0 – bardzo dobra, 4.5 – plus dobra, 4.0 – dobra, 3.5 – plus dostateczna, 3.0 – dostateczna, 2.0 - niedostateczna

Merytoryczna ocena pracy przez recenzenta

Końcowa ocena pracy przez recenzenta:

Data: Podpis:

Skala ocen: (6.0 – celująca), 5.0 – bardzo dobra, 4.5 – plus dobra, 4.0 – dobra, 3.5 – plus dostateczna, 3.0 – dostateczna, 2.0 – niedostateczna

Spis treści

1 Wstęp.....	6
2 Cel pracy.....	6
3 Narzędzia użyte w pracy.....	6
3.1 Java.....	6
3.2 Gradle.....	7
3.2.1 Użycie w pracy.....	7
3.3 Docker.....	7
3.3.1 Docker-compose.....	8
3.3.2 Użycie w pracy.....	9
3.4 Android.....	9
3.4.1 Android Annotations.....	10
3.5 Git.....	10
3.5.1 GitHub.....	11
3.6 TravisCI.....	11
3.7 Google Compute Engine.....	11
3.8 IntelliJ IDEA.....	12
4 Projekt Spring.....	12
4.1 Spring Core.....	12
4.1.1 Contener Benów.....	12
4.1.2 Dependency Injection.....	12
4.2 Spring Security.....	12
4.3 Spring Data.....	12
4.3.1 Spring Data Rest.....	12
4.4 Spring Boot.....	12
4.5 Spring Cloud.....	12
4.5.1 Eureka.....	12
4.5.2 Zuul.....	12
4.5.3 Hystrix.....	12
4.5.4 Ribbon.....	12
4.5.5 Config Server.....	12
4.5.6 Feign.....	12
5 Architektura aplikacji.....	12
5.1 Mikroserwisy.....	12
6 Funkcjonalności systemu.....	12
7 Opis aplikacji Android.....	12
8 Zawartość dołączonej płyty CD.....	12
9 Bibliografia.....	12

1 Wstęp

2 Cel pracy

3 Narzędzia użyte w pracy

3.1 Java

Java jest obiektywnym językiem programowania stworzonym w Sun Microsystems [1](obecnie część Oracle). Programy napisane w Javie są kompilowane do kodu pośredniego (bytecode) a następnie uruchamiane na wirtualnej maszynie (ang. Java Virtual Machine, JVM) . Dzięki takiemu rozwiązaniu skompilowany program jest niezależny od platformy wystarczy tylko że będziemy mogli zainstalować JVM. W JVM jest dodatkowo wbudowany mechanizm automatycznie zwalniający pamięć (ang. Garbage Collector) przez co nie musimy zajmować się zarządzaniem pamięcią. Składnia języka jest silnie wzorowana na C++ . Oprócz tego Java posiada aktywną i

rozbudowaną społeczność oraz wiele dostępnych narzędzi pomocniczych. Wszystko to sprawia że jest jednym z najpopularniejszych języków programowania.

3.2 Gradle

Gradle jest otwarto źródłowym narzędziem budującym pozwala na definiowanie skryptów budujących w języku Groovy [2]. Dzięki niemu możemy budować aplikacje na różne platformy napisane w różnych językach. Posiada bogatą kolekcję rozszerzeń która pozwala rozbudowywać oraz upraszcza skrypty budujące.

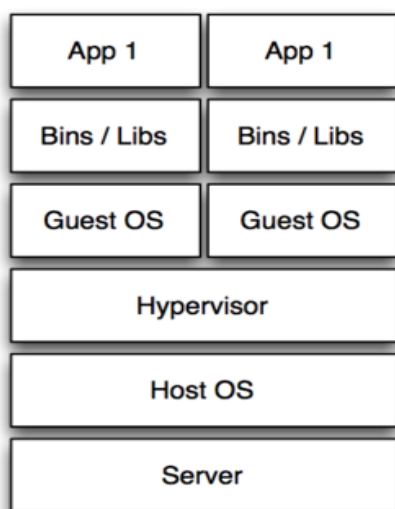
3.2.1 Użycie w pracy

W aplikacji gradle został wykorzystany do następujących rzeczy:

- Budowy części serwerowej
- Budowy aplikacji Android
- Generowanie plików Dockerfile
- Uruchamiania poszczególnych części aplikacji dla celów developerskich

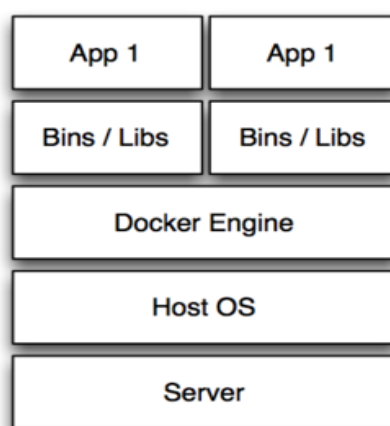
3.3 Docker

Docker jest to otwarto źródłowe narzędzie do uruchamiania aplikacji wewnątrz kontenerów. Kontener jest to lekka i przenośna maszyna wirtualna która może zostać uruchomiona na dowolnym serwerze z systemem Linux [3].



Virtual Machines

Ilustracja 1: Docker



Docker

<http://core0.staticworld.net/images/article/2016/07/dockerswarm-fig01-100671308-large.idge.png>

Pliku Dockerfile to przepis jak stworzyć obraz docker zawiera on wszystkie rzeczy potrzebne do uruchomienie aplikacji oraz zależności. Następnie z tego obrazu możemy stworzyć kontener czyli działającą maszynę wirtualną z naszą aplikacją. Dzięki wsparciu różnych platform PaaS dla Docker jest on idealnym rozwiązaniem do wgrywania różnych usług chmurowych zachowując przy tym niezależność od platformy na którą jest wgrywany.

3.3.1 Docker-compose

Docker-compose jest narzędziem pomocniczym dla docker które pozwala na uruchomienie oraz połączenie wielu kontenerów. Narzędzie opiera się o plik docker-compose.yml w którym konfigurujemy jakie obrazy dockera mają być ściągnięte oraz jak mają być połączone, dodatkowo tworzy wewnętrzną sieć dla naszych kontenerów. Możemy upublicznić niektóre porty z konkretnych kontenerów aby uzyskać dostęp do aplikacji.


```
discovery-server:
  image: magisterka-cms/image-discovery-server
  ports:
    - "8770:8080"
  expose:
    - "8080"
  external_links:
    - image-config-server:config-server
edge-server:
  image: magisterka-cms/image-edge-server
  ports:
    - "8769:8080"
  links:
    - discovery-server
  external_links:
    - image-config-server:config-server
```

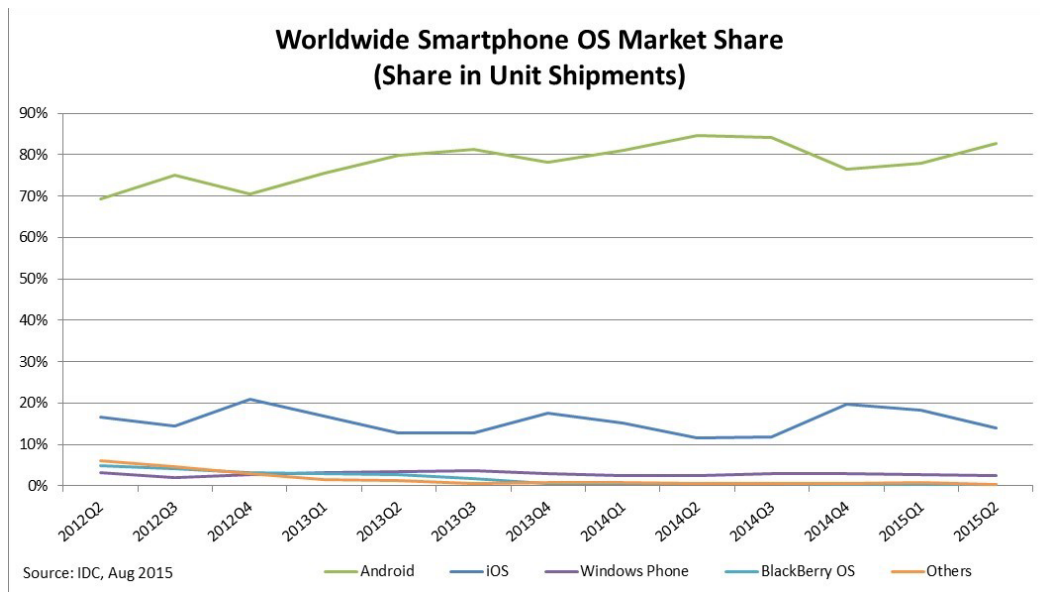
Kod 1: docker-compose.yml Przykład konfiguracji

3.3.2 Użycie w pracy

Docker został użyty aby wgrywać aplikację na serwer zewnętrzny Google Compute Engine. Dzięki użyciu narzędzia pomocniczego docker-compose wgranie całej plikacji sprowadza się do wykonania jednego polecenia.

3.4 Android

Android jest to system operacyjny z jądrem Linux dla urządzeń mobilnych takich jak telefony komórkowe, smartfony, tablety i netbooki[4]. Obecnie jest najpopularniejszym systemem operacyjny na urządzenia mobilne.



Ilustracja 2: Popularność systemów mobilnych

Oprócz dużej społeczności posiada bardzo rozwinięte narzędzia developerskie. Aplikacje natywne są tworzone w języku Java a warstwa widokowa za pomocą plików XML, oprócz tego posiada zaawansowane narzędzia do dostosowywania widoków, kolorów w zależności od urządzenia.

3.4.1 Android Annotations

Android annotations jest otwarto źródłowym narzędziem wspomagającym tworzenie aplikacji na platformę Android[5]. Dzięki użyciu adnotacji możemy bardzo uprościć nasz kod oraz zwiększyć jego przejrzystość.

```
@EActivity(R.layout.activity_create_template)
@OptionsMenu(R.menu.create_template_menu)
public class ApplicationTemplateDetailsActivity extends AppCompatActivity {
    public static final String TEMPLATE_INTENT = "DOCUMENT_INTENT";
    ApplicationTemplateDTO template;
    @RestService
    ApplicationTemplateClient templateClient;
    @Bean
    ApplicationTemplateDetailsAdapter adapter;
    @Bean
    ActionsAdapter actionsAdapter;
}
```

Kod 2: Przykład użycia android Annotations

3.5 Git

Git jest rozproszonym systemem kontroli wersji pierwotnie stworzonym na potrzeby rozwoju jądra Linuxa [6]. Obecnie jest jednym z najpopularniejszych systemów kontroli wersji (ang. Version Control System z skrócie VCS) używanych w tworzeniu oprogramowania. Jego rozproszona natura umożliwia łatwą współpracę w czasie tworzenia aplikacji. Dodatkowo istnieje wiele usług

hostujących repozytoria Git np. Bitbucket czy GitHub.

3.5.1 GitHub

GitHub jest platformą hostującą repozytoria Git. Mamy do wyboru bezpłatne repozytoria publiczne oraz płatne repozytoria prywatne. Oprócz tego mamy do dyspozycji issue tracker który pozwala na monitorowanie naszych postępów w pisaniu kodu, zgłaszanie błędów, podpinanie commitów pod poszczególne zgłoszenia oraz wyznaczanie kamieni milowych naszej aplikacji. Dodatkowo mamy do dyspozycji bogatą kolekcję webhooks która pozwala na integrację z zewnętrznymi narzędziami np. TravisCI.

3.6 TravisCI

TravisCI jest narzędziem do Ciągłej Integracji (ang. Continuous Integration, CI). Ciągła integracja polega na jak na ciągłym budowaniu i testowaniu aplikacji aby jak najszybciej wyłapywać błędy powstałe podczas tworzenia systemu. Travis posiada pakiet darmowy który choć z ograniczonymi funkcjonalnościami stanowi i tak bardzo przydatne narzędzie do CI. Nasze środowisko konfiguruje za pomocą pliku `.travis.yml`, pozwala to na łatwe dodawanie usług do naszego środowiska testowego np. Javy, Dockera.

```
sudo: required
language: java
services:
  - docker
script:
  - ./gradlew build -Pbuild-travis="" -x test
after_success:
  - docker login -e="$DOCKER_EMAIL" -u="$DOCKER_USERNAME" -p="$DOCKER_PASSWORD";
```

Tekst 1: .travis.yml Przykład konfiguracji

3.7 Google Compute Engine

Google Compute Engine jest narzędziem IAAS i jest częścią Google Cloud Platform. Infrastructure as a Service (IAAS, z ang. „infrastruktura jako usługa”) to jeden z modeli chmury obliczeniowej. Jest to usługa polegająca na dostarczeniu całej infrastruktury informatycznej, takie jak wirtualizowany sprzęt, skalowany w zależności od potrzeb użytkownika[7]. W aplikacji został wykorzystany do hostowania Dockera na którym jest uruchomiona aplikacja.

3.8 IntelliJ IDEA

IntelliJ IDEA jest zintegrowanym środowiskiem programistycznym (ang. Integrated Development Environment, IDE) stworzonym przez firmę JetBrains. Jest to jedno z najpopularniejszych narzędzi tego typu. Zawdzięcza to świetną integracją z wieloma frameworkami oraz narzędziami

wspomagającymi developerów np. Spring, SpringBoot, Docker, Git i wiele innych. Dodatkowo możemy rozszerzyć jego funkcjonalność dzięki bogatej bazie pluginów. Oprócz tego jest to wydajne środowisko oraz posiada intuicyjny interfejs.

3.8.1 Android Studio

Android Studio jest to wersja IntelliJ IDEA przeznaczona specjalnie do tworzenia aplikacji androidowych. Jest to oficjalne środowisko zalecane przez Google, posiada dodatkową integrację z usługami Google.

4 Projekt Spring

4.1 Spring Core

4.1.1 Contener Benów

4.1.2 Dependency Injection

4.2 Spring Security

4.3 Spring Data

4.3.1 Spring Data Rest

4.4 Spring Boot

4.5 Spring Cloud

4.5.1 Eureka

4.5.2 Zuul

4.5.3 Hystrix

4.5.4 Ribbon

4.5.5 Config Server

4.5.6 Feign

5 Architektura aplikacji

5.1 Mikroserwisy

6 Funkcjonalności systemu

7 Opis aplikacji Android

8 Zawartość dołączonej płyty CD

9 Bibliografia

1: <https://pl.wikipedia.org/wiki/Java>

2: <https://en.wikipedia.org/wiki/Gradle>

- 3: [https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))
- 4: [https://pl.wikipedia.org/wiki/Android_\(system_operacyjny\)](https://pl.wikipedia.org/wiki/Android_(system_operacyjny))
- 5: <http://androidannotations.org/>
- 6: [https://pl.wikipedia.org/wiki/Git_\(oprogramowanie\)](https://pl.wikipedia.org/wiki/Git_(oprogramowanie))
- 7: https://pl.wikipedia.org/wiki/Infrastructure_as_a_Service