Testcontainers

a SpringBoot

Testcontainers

- Opensource a zadarmo
  - Monetizacia cez AtomicJar (Testcontainer Desktop, Testcontainers Cloud)
- Podpora viacerých programovacích jazykov
  - Java, Python, Node.js, .net ...
- Docker s integráciou pre unit testing
- Databázy, message brokre ...
  - Ale aj Selenium a Wiremock
  - Alebo čokoľvek čo môže byť dokerizované

Prečo sa používajú Testcontainery

- Integračné testovanie – databáza, message broker, keycloak ...
  - „Out-of-process dependency“
- Testovanie vrstvy prístupu k dátam – databáza
- UI/Akceptačné testovanie - Selenium

- Umožňujú jednoduchšie testovanie business logiky – požiadaviek klientov

Testy

- SpringBoot
  - Potrebujeme vedieť typ a verziu databázy
  - Pridať „testcontainer“ anotácie:
    - @Testcontainers – anotuje sa trieda
    - @Container – anotuje sa premenná testcontainera
    - **@ServiceConnection** – anotuje sa **statická** premenná testcontainera
      - Pozor na kontext – kontajner sa posúva medzi testovacími metódami, zmeny sú perzistentné pre všetky @Test metódy v rámci testovacej triedy
      - Pozor na vzájomne ovplyvňovanie sa testov

Testy 2

- SpringBoot
  - Potrebujeme vedieť typ a verziu databázy
  - Pridať „testcontainer“ anotácie:
    - @Testcontainers – anotuje sa trieda
    - @Container – anotuje sa premenná testcontainera
  - Pridať „springboot“ anotáciu:
    - **@DynamicPropertySource** – anotuje sa statická metóda
      - Flexibilita pri nastavovaní premenných

Testy 3

- Generický testcontainer
  - Potrebujeme vedieť názov docker image
  - Pridať „testcontainer“ anotácie:
    - @Testcontainers – anotuje sa trieda
    - @Container – anotuje sa premenná testcontainera
  - Pridať „springboot“ anotáciu:
    - **@DynamicPropertySource** – anotuje sa statická metóda
      - Flexibilita pri nastavovaní premenných

Testy 4

- ContainersConfig trieda
  - Samostatná trieda na konfiguráciu testcontainerov
  - Konfiguračná trieda musí byt anotovaná
    - @TestConfiguration
  - Testcontainer konfigurácie sú definované ako @Bean

- V testovacej triede:
  - Okrem @Testcontainers pridáme @Import(ContainersConfig.class) a testcontajner začne fungovať automagicky

Vlastná konfigurácia

- Testcontainers umožňujú prepísať základne nastavenie:
- **Linux:** /home/myuser/.testcontainers.properties
- **Windows:** C:/Users/myuser/.testcontainers.properties
- **macOS:** /Users/myuser/.testcontainers.properties

- Vhodné ak sa spravuje vlastný docker hub (nevhodné ak sa na to zabudne a robí sa súkromný projekt)

  - hub.image.name.prefix=docker.moja-firma.com/

Testcontainers a maven - build projektu

- Testcontainers testy sú pomalé, oproti klasickým unit testom
  - viac väzieb, komplexnejšie nastavenie prostredia
  - Preto maven-failsafe plugin
  - Maven fázy:
    validate => compile => **test** => package => **verify** => install => deploy

Testcontainers – overené praktiky

- Nepoužívajte hardcoded values
- Nepoužívajte „latest“ – používajte rovnakú verziu ako je na produkcii
- Ak je niečo potrebne nakopírovať, tak kopírujte, nie pripájajte
- Dodržujte správny životný cyklus

Ďakujem za pozornosť