PROBLÉMA-ANALÍZIS ENTERPRISE RENDSZEREKBEN

4. forduló

adnovum

A kategória támogatója: Adnovum Hungary Kft.

RENDELKEZÉSRE ÁLLÓ IDŐ:

12:00

Ismertető a feladathoz

Fontos információk

A forduló után a megoldások publikálásával együtt iránymutatásként elérhetőek lesznek a **helyezéssel kapcsolatos információk**, látni fogod, hogy a kategóriában a játékosok 20%, 40% vagy 60%-a közé tartozol-e épp.

Felhívjuk figyelmedet, hogy a következő, **5. fordulótól az egyes kategóriák csak a kijelölt napokon lesznek megoldhatóak 7-22 óra között**, érdemes letöltened a naptárat a Kategóriáim menüpontban.

Felhasznált idő: 01:49/12:00 **Elért pontszám:** 0/12

1. feladat 0/6 pont

A következő kódrészlet azt az API végpont definíciót és adatbázis lekérdezést mutatja be, amit a rendszer egy publikusan elérhető weboldalán elhelyezett kereső használ. A weboldalon egy szabad szöveges keresőfeltétel alapján jelennek meg a mindenki számára látogatható információs pontok. A rendszer az információs pontok nyilvánosan hozzáférhető adatait kezeli a világ minden tájáról.

Milyen javításokat igényel a kódrészlet ahhoz, hogy optimálisan működjön?

```
@RestController
public class SearchController {
    @Autowired
    InformationBoothRepository informationBoothRepository;
    @GetMapping("/search")
    public List<InformationBooth> search(@RequestParam(value = "name") String name) {
```

```
            return informationBoothRepository.searchByName(name);
        }
    }
    public interface InformationBoothRepository extends JpaRepository<InformationBooth, Long> {
        @Query("SELECT ib FROM InformationBooth ib WHERE ib.name LIKE CONCAT('%', :name, '%')")
        List<InformationBooth> searchByName(@Param("name") String name);
    }
```

Válaszok

- [ ] SQL Injection elleni védelmet kell beraknunk: reguláris kifejezéssel megszorítjuk a lekérhető neveket
- [x] Indexet kell létrehoznunk a név oszlopon
- [x] Limitálnunk kell a lekérhető oszlopok mennyiségét, mert különben túlterhelődhet a rendszer
- [ ] Anonimizálnunk kell az visszaadott adatokat, mivel különben sértjük a GDPR-t

Magyarázat

SQL Injection elleni védelmet kell beraknunk: reguláris kifejezéssel megszorítjuk a lekérhető neveket: Nem, a kód paraméterezett JPQL-t használ, ami védelmet nyújt az SQL injection ellen.

Indexet kell létrehoznunk a név oszlopon, illetve Limitálnunk kell a lekérhető oszlopok mennyiségét, mert különben túlterhelődhet a rendszer: Igen, ha üres stringet küld be a felhasználó, az egész adatbázis tábla megjelenik az eredményhalmazban.

Anonimizálnunk kell az visszaadott adatokat, mivel különben sértjük a GDPR-t: Nem, a rendszer nem kezel felhasználói adatot.

2. feladat 0/6 pont

Az egyik adatfeldolgozó szerverünkön az alábbi parancsot futtatjuk:

docker run --rm -v $(pwd):/work awesome-service/calculate-statistics

A program a rendelkezésre álló adatokból készít statisztikát. A folyamat nagyon sokáig tart és már fut is egy ideje, így nem tudjuk többször lefuttatni. A futás eredményeképpen az alábbi fájlok jönnek létre a konténer fáljrendszerén:

- /work/hello.txt

- /opt/world.txt

Miután a folyamat befejeződött, az alábbi állítások közül melyik igaz?

Válasz

- ( ) Mindkét fájl elérhető lesz az aktuális könyvtárban.
- ( ) Egyik fájl sem érhető el az aktuális könyvtárban, de mindkettőhöz hozzáférhetünk a konténer fájlrendszerén.
- ( ) A /opt/world.txt elérhető az aktuális könyvtárban, a másik törlésre került.

- ◯ A /work/hello.txt elérhető az aktuális könyvtárban, a másik törlésre került.
- ◯ A /opt/world.txt elérhető az aktuális könyvtárban, a másikhoz hozzáférhetünk a konténer fájlrendszerén.
- ◯ A /work/hello.txt elérhető az aktuális könyvtárban, a másikhoz hozzáférhetünk a konténer fájlrendszerén.

Magyarázat

A -v kapcsoló után megadott mount miatt a konténer /work könyvtára a gazda rendszer aktuális könyvtárára lesz leképezve, így az ott létrehozott állományok megmaradnak és elérhetőek a gazda rendszeren.

Az --rm kapcsoló miatt a konténerhez tartozó névtelen volumek törlésre kerülnek a futás után, így a többi fájl, köztük a /opt könytár tartalma a művelet után már nem hozzáférhető.

3. feladat 0/0 pont

A következők közül mely intézkedések segitenek egy Denial of Service támadás megállításában?

Válaszok

- ☑ Rate limiting
- ☑ Tűzfal
- ☐ Alkalmazás memóriájának limitálása
- ☑ Feldolgozandó kérések aszinkron queue-ba helyezése

Magyarázat

Alkalmazás memóriájának limitálása: Nem, a memória limitálása csak az alkalmazás korábbi túlterhelődéséhez fog vezetni.

Feldolgozandó kérések aszinkron queue-ba helyezése: Igen, a Queue-ba való aszinkron feldolgozással mi választjuk meg a feldolgozás ütemét és ezzel az egyébként kiszolgálhatatlan kéréseket is kiszolgálhatjuk amint fölszabadulnak erőforrások a későbbiekben.

Frissítés (2021.11.30): versenyzői visszajelzések alapján mind a négy válaszlehetőség mellett lehet érvelni, ezért a kérdést 0 pontosra állítottuk.

KÉSZÍTETTE

Megjelenés

Világos