SZAKMAI VIZSGA PRÓBA · INFOSZAKMAIVIZSGA.HU

INFORMATIKAI RENDSZER- ÉS ALKALMAZÁS-ÜZEMELTETŐ TECHNIKUS

PRÓBA SZAKMAI VIZSGA

Felhőszolgáltatás telepítése és beállítása
Nehéz „B” feladatlap

Projektfeladat – B) Hálózatok és szerverek telepítése és beállítása vizsgarész, III. feladat

Ajánlott időtartam: 60 perc
Elérhető pontszám: 20 pont

Beadott dokumentumok	
Piszkozati pótlapok száma	

Értékelésre az alábbi állományokat adom be:	
Képernyőképek (felhőszolgáltatás)	

INFOSZAKMAIVIZSGA.HU

Általános tudnivalók

- A feladatban egy saját, választott **felhőszolgáltatónál** alakítasz ki szolgáltatásokat az infoszakmaivizsga.hu tesztkörnyezetéhez. A működést a végén ellenőrzöd is.
- A feladat elkészítésére 60 perc áll rendelkezésedre. A felhős erőforrások létrehozási (kiépítési) idejét ebbe nem számítjuk bele.
- A megoldás során **kizárólag képernyőképeket** kell összegyűjtened. A feladatok végén zárójelben, dőlt betűvel jelöljük, miről kell képernyőképet készítened, és milyen néven kell elmentened. A képeket PNG formátumban a **képernyőképek helye** mappába mentsd! A fájlnév eleje mindig a feladat sorszáma, például: **`03_felhos_halozat.png`**. Ha egy feladathoz több kép kell, a kiterjesztés elé írj sorszámot: **`03_felhos_halozat_02.png`**.
- A felhőszolgáltatók eltérő elnevezéseket és felületeket használnak. Mindig a **leírt működést** valósítsd meg a saját szolgáltatód megfelelő eszközeivel; a konkrét menüpontok eltérhetnek.
- A feladatleírásban nem szereplő, de a működéshez szükséges paramétereket szabadon megválaszthatod.
- A képeken **jelszó, privát kulcs és fizetési adat ne látszódjon**. Ahol a feladat saját jelszót vagy kulcsot kér, azt te hozod létre; a vizsgán a házirend szerinti erős jelszót használj.
- A felhős erőforrások használata az előfizetésedtől függően **költséggel járhat**. A gyakorlás végén a képernyőképek mentése után töröld a létrehozott erőforrásokat (a részleteket a feladatsor végén találod).

infoszakmaivizsga.hu

Kiinduló állapot és szükséges szolgáltatások

Tovi megbízott, hogy helyezd üzembe az infoszakmaivizsga.hu **gyakorlási eredménytárának** tesztváltozatát, és bizonyítsd, hogy az adatai mentésből helyreállíthatók. Az alkalmazás két konténerből áll: egy webalkalmazásból és egy PostgreSQL-adatbázisból. A forrásokat készen kapod. Feladatod a rendszer elindítása, a belső adatbázis-kapcsolat kialakítása, egy mentés elkészítése, majd egy szándékosan törölt tesztadat helyreállítása.

Témakör	Megvalósítás
SaaS	OneDrive-mentéstár és fájlhelyreállítás
IaaS	Felhős hálózat, Linux VM és hozzáférési szabályok
VM és konténer	Kétkonténeres alkalmazás, belső hálózat, tartós kötet, adatbázismentés és helyreállítás

Szükséges: saját Microsoft-fiók (OneDrive), saját, aktivált felhő-előfizetés Linux VM-hez, böngésző és SSH-kliens. Üres felhős környezetből indulj: a VM-et, a Dockert és a konténereket is neked kell létrehoznod. A mintában kizárólag kitalált tesztadatok szerepelnek; a törlési és helyreállítási műveleteket csak az ehhez a feladathoz létrehozott környezetben végezd el. Programkódot és SQL-parancsokat nem kell írnod.



Tovi javaslata az infoszakmaivizsga.hu-tól

A PostgreSQL csak a **belső** konténerhálózaton érhető el, gazdagépi port nélkül; csak a webalkalmazás 8080-as portja publikus. Bináris mentés (pg_dump) átirányításakor **ne** használj pszeudoterminált – Compose esetén a `-T` kapcsolóval kapcsold ki! A kapott tesztalkalmazás az egyszerűség kedvéért az adatbázis-adminisztrátort használja; éles rendszerben külön, korlátozott fiók kellene.

Feladatok

1. Készíts felhős mentéstárat! 2 pont

A OneDrive-ban hozz létre egy mappát `Tovi-eredmenytar-mentesek` néven! A saját számítógépeden készíts UTF-8 kódolású **`tovi-mentesi-terv.txt`** fájlt, amely röviden leírja az eredménytár mentési tervét (különálló, kitalált adatokat tartalmazó eredménytár; a módosítás előtt PostgreSQL-mentés készül; a mentés a Linux-gépen és a OneDrive-ban is megőrződik; a visszaállítás külön tesztadatbázisban ellenőrződik). Töltsd fel a fájlt a mappába – a mentéstárhoz **ne** hozz létre nyilvános megosztást!

(Készíts képernyőképet a OneDrive-mappáról és a feltöltött tervről **`01_mentestar.png`** *néven, és a terv teljes tartalmáról* **`01_mentesi_terv.png`** *néven!)*

2. Hozd létre a felhős hálózatot! 2 pont

Beállítás	Elvárt érték
Hálózat neve	`tovi-eredmeny-net`
Hálózat címtartománya, ha külön megadható	`10.130.0.0/16`
Alhálózat neve, ha külön elnevezhető	`tovi-eredmeny-subnet`
Alhálózat címtartománya	`10.130.1.0/24`

Ha csak egy címtartomány adható meg, a `10.130.1.0/24` tartományt használd. Biztosíts internetelérést a létrehozandó VM számára a csomagok és a konténerképek letöltéséhez!

(Készíts képernyőképet a hálózatról és az alhálózatról **`02_felhos_halozat.png`** *néven, és az internetelérést biztosító beállításról* **`02_interneteleres.png`** *néven!)*

3. Hozd létre az eredménytár virtuális gépét! 2 pont

Beállítás	Elvárt érték
Név	`tovi-eredmenytar`
Operációs rendszer	Ubuntu Server 24.04 LTS
Processzor / Memória / Lemez	Legalább 1 vCPU / 2 GiB / 15 GiB
Magánhálózat	Az előző feladat hálózata
Internetes elérés / Hitelesítés	Nyilvános IPv4-cím / SSH-kulcs

A felhős hozzáférési szabályokban: `TCP 22` és `TCP 8080` csak a saját aktuális nyilvános IPv4-címedről, `/32` maszkkal; `TCP 5432` ne legyen internetes bejövő engedély! A webalkalmazás most saját használatú tesztfelület, ezért a 8080-as portot sem kell mindenki számára megnyitni. Jelentkezz be SSH-val!

(Készíts képernyőképet a VM nevéről, állapotáról és méretéről **`03_virtualis_gep.png`** *néven, és a hozzáférési szabályokról* **`03_halozati_vedelem.png`** *néven!)*

4. Telepítsd a Dockert, és készítsd elő a forrásokat! 2 pont

Telepítsd a Docker Engine-t és a Docker Compose bővítményt, és állítsd be, hogy a Docker fusson és rendszerindításkor automatikusan elinduljon! A kezdőkönyvtáradban hozz létre egy `tovi-eredmenytar` mappát, és másold ide a teljes forráscsomagot. A **`.env.example`** fájlról készíts **`.env`** nevű másolatot: a jelszó helyőrzőjét cseréld saját, legalább 20 karakteres, betűkből és számjegyekből álló jelszóra (a gyakorlat közben ne változtasd meg), és a **`.env`** fájl jogosultsága legyen `600`! A Docker és a Compose verzióját, valamint a forrásfájlok listáját mentsd a `telepites-ellenorzese.txt` fájlba, és listázd ki (a **`.env`** tartalmát ne jelenítsd meg)!

(Készíts képernyőképet a verziókról, a forrásfájlokról és a **`.env`** *fájl jogosultságáról, és mentsd el* **`04_docker_es_forrasok.png`** *néven!)*

5. Indítsd el a kétkonténeres rendszert! 2 pont

A kapott Compose-fájl alapján építsd fel és indítsd el az alkalmazást! Ellenőrizd, hogy a `db` szolgáltatás működik és egészséges, a `web` szolgáltatás fut, kizárólag a webalkalmazás portja van a gazdagépre publikálva, a PostgreSQL a belső konténerhálózaton érhető el, és az adatbázis adatai a névvel ellátott kötetben találhatók. A saját számítógépeden nyisd meg a `http://A_SZERVER_NYILVANOS_IPV4_CIME:8080` címet – a kezdőlapon három teszteredménynek kell megjelennie!

(Készíts képernyőképet a Compose-szolgáltatások állapotáról és portjairól **`05_kontenerek.png`** *néven, a belső hálózatról és az adatbázis kötetcsatolásáról* **`05_halozat_es_kotet.png`** *néven, és a webalkalmazásról három rekorddal* **`05_eredeti_eredmenytar.png`** *néven!)*

6. Készíts adatbázismentést! 2 pont

Hozz létre egy `mentesek` könyvtárat a projektmappában! A PostgreSQL `pg_dump` eszközével készíts **custom formátumú** mentést a `tovi_eredmenyek` adatbázisról a `mentesek/tovi-eredmenyek.dump` fájlba (a futó adatbázis-konténerből is használhatod; bináris mentés átirányításakor Compose esetén a `-T` kapcsolóval kapcsold ki a pszeudoterminált). Ellenőrizd a mentést a `pg_restore --list` funkciójával, és a listát mentsd a `mentesek/mentes-tartalma.txt` fájlba! Készíts SHA-256 ellenőrzőösszeget a `mentesek/tovi-eredmenyek.sha256` fájlba. Listázd ki a mentés méretét, tartalomjegyzékét és ellenőrzőösszegét!

(Készíts képernyőképet a mentési fájl nevéről és nem nulla méretéről **`06_adatbazismentes.png`** *néven, a mentés tartalomjegyzékéről* **`06_mentes_tartalma.png`** *néven, és a mentés SHA-256 értékéről* **`06_mentes_ellenorzoosszege.png`** *néven!)*

7. Ellenőrizd a felhős mentés helyreállíthatóságát! 2 pont

Másold le a mentési fájlt a saját számítógépedre, majd töltsd fel a OneDrive `Tovi-eredmenytar-mentesek` mappájába! Ezután töröld a mentési fájlt a OneDrive-ból, a OneDrive lomtárából állítsd vissza, töltsd le a visszaállított fájlt, és másold vissza a Linux-gép `mentesek` könyvtárába `tovi-eredmenyek-onedrive.dump` néven! Számítsd ki a visszatöltött fájl SHA-256 értékét, és hasonlítsd össze az eredeti mentésével – a két értéknek egyeznie kell.

(Készíts képernyőképet a mentésről a OneDrive lomtárában **`07_mentes_lomtarban.png`** *néven, a helyreállított mentésről a OneDrive-mappában* **`07_mentes_visszaallitva.png`** *néven, és az eredeti és a visszatöltött fájl egyező SHA-256 értékéről* **`07_azonos_mentesek.png`** *néven!)*

8. Modellezd egy teszteredmény elvesztését! 2 pont

Az eredeti `tovi_eredmenyek` adatbázison futtasd a kapott **`tesztek/tesztadat-torlese.sql`** fájlt – ez kizárólag a 2-es azonosítójú, kitalált Linux-teszteredményt törli! Frissítsd a webalkalmazást, és ellenőrizd, hogy már csak két eredmény szerepel benne, és a Linux-rekord hiányzik. Futtasd az **`ellenorzes.sql`** fájlt is, és az eredményt mentsd a `torles-utani-ellenorzes.txt` fájlba, majd listázd ki!

(Készíts képernyőképet a webalkalmazásról a hiányzó Linux-rekorddal **`08_hianyzo_eredmeny.png`** *néven, és az adatbázis-ellenőrzésről két rekorddal* **`08_torles_ellenorzese.png`** *néven!)*

9. Állítsd vissza a mentést külön adatbázisba! 2 pont

Ugyanabban a PostgreSQL-kiszolgálóban hozz létre egy új, üres `tovi_helyreallitott` adatbázist! A OneDrive-ból visszaállított és ellenőrzött `mentesek/tovi-eredmenyek-onedrive.dump` mentést töltsd vissza ebbe az új adatbázisba a `pg_restore` segítségével (hibánál leálló végrehajtással; a konténerbe másolt fájlból vagy szabványos bemenetről, pszeudoterminál nélkül)! Az eredeti, két rekordot tartalmazó adatbázist most ne módosítsd. Futtasd az **`ellenorzes.sql`** fájlt a helyreállított adatbázison, és az eredményt mentsd a `helyreallitas-ellenorzese.txt` fájlba! Elvárt eredmény: **3 rekord**, a Linux-teszteredmény ismét szerepel, és az elért pontszámok összege **51**.

(Készíts képernyőképet a visszaállítás sikeres végrehajtásáról **`09_adatbazis_helyreallitasa.png`** *néven, és a helyreállított adatbázis nevéről és ellenőrzési eredményéről* **`09_helyreallitott_adatok.png`** *néven!)*

10. Irányítsd át az alkalmazást a helyreállított adatbázisra! 2 pont

A **`.env`** fájlban az `APP_DB_NAME` értékét módosítsd `tovi_helyreallitott`-ra! A webalkalmazás konténerét hozd létre újra úgy, hogy az új környezeti változó érvényesüljön (az adatbázis-konténert és kötetét ne töröld)! Ellenőrizd, hogy az alkalmazás ismét három

rekordot mutat, a Linux-teszteredmény visszakerült, és az oldalon a használt adatbázis neve `tovi_helyreallitott`. Indítsd újra az adatbázis-konténert, és miután ismét egészséges, frissítsd az alkalmazást, és ellenőrizd az adatok megmaradását! A Linux-gépről a weboldal HTTP-válaszfejlécét mentsd a `vegso-http-ellenorzes.txt` fájlba, és listázd ki (sikeres, 200-as HTTP-válasz szerepeljen benne)!

(Készíts képernyőképet a webalkalmazásról a helyreállított adatbázissal és három rekorddal **`10_helyreallitott_alkalmazas.png`** *néven, az újraindítás után egészséges adatbázis-konténerről* **`10_adatbazis_ujrainditas.png`** *néven, és a sikeres HTTP-ellenőrzésről* **`10_http_ellenorzes.png`** *néven!)*

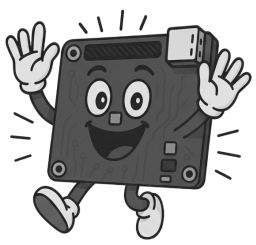
Pontösszesítő

#	Feladat	Elérhető	Elért
1.	Felhős mentéstár (OneDrive)	2 pont	
2.	Felhős hálózat és internetelérés	2 pont	
3.	Eredménytár virtuális gépe és tűzfal	2 pont	
4.	Docker és a források előkészítése	2 pont	
5.	A kétkonténeres rendszer indítása	2 pont	
6.	Adatbázismentés készítése	2 pont	
7.	Felhős mentés helyreállíthatósága	2 pont	
8.	Teszteredmény elvesztésének modellezése	2 pont	
9.	Visszaállítás külön adatbázisba	2 pont	
10.	Átirányítás a helyreállított adatbázisra	2 pont	
	Összesen	**20 pont**	



Tovi takarítási tanácsa

Ha már nincs szükséged a tesztkörnyezetre, a képernyőképek mentése után **töröld a létrehozott felhős erőforrásokat** (virtuális gépek, adatbázisok, alkalmazások, hálózatok). Ellenőrizd a megmaradt lemezeket, mentéseket és a lefoglalt nyilvános IP-címeket is: egy erőforrás leállítása önmagában nem feltétlenül szüntet meg minden költséget. A felhős fiókokat, tesztfelhasználókat és megosztásokat is érdemes megszüntetni.



Ha szeretnél felkészülni a 13-os informatikai szakmai vizsgára rövid videókon, online futtatható szervereken és automatikus kiértékelésű feladatokon keresztül, akkor keresd fel az **infoszakmaivizsga.hu**-t!

infoszakmaivizsga.hu

Kiadható forráscsomag

Az alábbi fájlok a **forrás/tovi-eredmenytar** mappában találhatók (az `init` és `tesztek` alkönyvtárral együtt). A PostgreSQL-hez nincs gazdagépi port publikálva.

compose.yaml

```
name: tovi-eredmenytar

services:
  db:
    image: postgres:16-alpine
    environment:
      POSTGRES_DB: tovi_eredmenyek
      POSTGRES_USER: tovi_admin
      POSTGRES_PASSWORD: ${DB_PASSWORD:?Allitsd_be_a_DB_PASSWORD_erteket}
    volumes:
      - db_adatok:/var/lib/postgresql/data
      - ./init:/docker-entrypoint-initdb.d:ro
    networks:
      - belso
    healthcheck:
      test:
        - CMD-SHELL
        - pg_isready -U tovi_admin -d tovi_eredmenyek
      interval: 5s
      timeout: 3s
      retries: 20
    restart: unless-stopped

  web:
    build: .
    environment:
      DB_HOST: db
      DB_PORT: "5432"
      DB_NAME: ${APP_DB_NAME:-tovi_eredmenyek}
      DB_USER: tovi_admin
      DB_PASSWORD: ${DB_PASSWORD:?Allitsd_be_a_DB_PASSWORD_erteket}
    ports:
      - "8080:8000"
    networks:
      - kulso
      - belso
    depends_on:
      db:
        condition: service_healthy
    restart: unless-stopped

volumes:
  db_adatok:

networks:
  kulso:
  belso:
    internal: true
```

.env.example

```
DB_PASSWORD=CSERELD_LE_SAJAT_HOSSZU_JELSZORA
APP_DB_NAME=tovi_eredmenyek
```

Dockerfile

```
FROM python:3.12-slim

ENV PYTHONDONTWRITEBYTECODE=1
ENV PYTHONUNBUFFERED=1

WORKDIR /app

COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt \
    && useradd --uid 10001 --create-home appuser

COPY app.py .

USER appuser

EXPOSE 8000

CMD ["gunicorn", "--bind", "0.0.0.0:8000", \
     "--workers", "2", "--access-logfile", "-", "app:app"]
```

requirements.txt

```
Flask>=3.1,<4
gunicorn>=23,<24
psycopg[binary]>=3.2,<4
```

init/01-adatok.sql

```
CREATE TABLE public.eredmenyek (
    azonosito INTEGER PRIMARY KEY,
    tanulo VARCHAR(50) NOT NULL,
    temakor VARCHAR(20) NOT NULL,
    pontszam INTEGER NOT NULL,
    maximum INTEGER NOT NULL,
    CHECK (pontszam >= 0 AND pontszam <= maximum)
);

INSERT INTO public.eredmenyek
    (azonosito, tanulo, temakor, pontszam, maximum)
VALUES
    (1, 'Teszt Elek', 'Windows', 18, 20),
    (2, 'Minta Anna', 'Linux', 16, 20),
    (3, 'Pelda Bence', 'Cisco', 17, 20);
```

tesztek/ellenorzes.sql

```
SELECT current_database() AS adatbazis;

SELECT
    azonosito,
    tanulo,
    temakor,
    pontszam,
    maximum
FROM public.eredmenyek
ORDER BY azonosito;

SELECT
    COUNT(*) AS rekordok_szama,
    SUM(pontszam) AS osszes_pontszam
FROM public.eredmenyek;
```

tesztek/tesztadat-torlese.sql

```
BEGIN;

DELETE FROM public.eredmenyek
WHERE azonosito = 2
  AND tanulo = 'Minta Anna'
  AND temakor = 'Linux';

COMMIT;
```

app.py

```
import os

import psycopg
from psycopg.rows import dict_row
from flask import Flask, render_template_string

app = Flask(__name__)

TEMPLATE = """
<!doctype html>
<html lang="hu">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Tovi gyakorlási eredménytára</title>
  <style>
    * { box-sizing: border-box; }
    body { margin: 0; background: #f3f6fb; color: #18243a;
           font: 17px/1.7 system-ui, sans-serif; }
    main { max-width: 1000px; margin: auto; padding: 40px 20px; }
    h1 { font-size: clamp(2rem, 5vw, 3rem); line-height: 1.2; }
    a { color: #214fac; }
    .card { margin: 22px 0; padding: 24px; background: white;
            border: 1px solid #dce3ee; border-radius: 16px; }
    .error { padding: 18px; background: #fff0f0; border-left: 4px solid #a52525; }
    .table-wrap { overflow-x: auto; }
    table { width: 100%; border-collapse: collapse; }
    th, td { padding: 12px; border-bottom: 1px solid #dce3ee; text-align: left; }
    th { background: #eef3fc; }
    footer { color: #4b5b73; font-size: .9rem; }
  </style>
</head>
<body>
<main>
  <a href="https://infoszakmaivizsga.hu">infoszakmaivizsga.hu</a>

  <h1>Tovi gyakorlási eredménytára</h1>

  <p>
    Az infoszakmaivizsga.hu a 13. osztályos diákokat
    készíti fel az informatikai szakmai vizsgára.
  </p>

  <p>
    Rövid, animált videókkal tanulhatsz.
    Az online Windows-, Linux- és Cisco-eszközszimulátorok
    azonnal kiértékelik a megoldásaidat.
  </p>
```

```
  {% if error %}
    <p class="error" role="alert">{{ error }}</p>
  {% else %}
    <section class="card">
      <h2>Tesztadatok</h2>
      <p>Használt adatbázis: <strong>{{ database }}</strong></p>
      <p>Rekordok száma: <strong>{{ rows|length }}</strong></p>

      <div class="table-wrap">
        <table>
          <thead>
            <tr>
              <th scope="col">Azonosító</th>
              <th scope="col">Tesztelő</th>
              <th scope="col">Témakör</th>
              <th scope="col">Pontszám</th>
            </tr>
          </thead>
          <tbody>
          {% for row in rows %}
            <tr>
              <td>{{ row.azonosito }}</td>
              <td>{{ row.tanulo }}</td>
              <td>{{ row.temakor }}</td>
              <td>{{ row.pontszam }} / {{ row.maximum }}</td>
            </tr>
          {% endfor %}
          </tbody>
        </table>
      </div>
    </section>
  {% endif %}

  <footer>
    Kitalált mintadatokat tartalmazó felhőüzemeltetési
    gyakorlóalkalmazás. Nem kapcsolódik az infoszakmaivizsga.hu
    valódi eredményeihez.
  </footer>
</main>
</body>
</html>
"""


@app.after_request
def disable_cache(response):
    response.headers["Cache-Control"] = "no-store"
    return response
```

```
@app.get("/")
def index():
    rows = []
    database = ""
    error = ""
    status = 200

    try:
        with psycopg.connect(
            host=os.environ["DB_HOST"],
            port=int(os.environ.get("DB_PORT", "5432")),
            dbname=os.environ["DB_NAME"],
            user=os.environ["DB_USER"],
            password=os.environ["DB_PASSWORD"],
            connect_timeout=5,
            row_factory=dict_row,
        ) as connection:
            with connection.cursor() as cursor:
                cursor.execute(
                    "SELECT current_database() AS database"
                )
                database = cursor.fetchone()["database"]

                cursor.execute(
                    """
                    SELECT azonosito, tanulo, temakor,
                           pontszam, maximum
                    FROM public.eredmenyek
                    ORDER BY azonosito
                    """
                )
                rows = cursor.fetchall()

    except Exception:
        app.logger.error(
            "TOVI-DB: Az adatbazis lekerdezese sikertelen."
        )
        error = (
            "Az eredménytár jelenleg nem érhető el. "
            "Ellenőrizd az adatbázis állapotát és "
            "az alkalmazás kapcsolati beállításait!"
        )
        status = 503

    return render_template_string(
        TEMPLATE,
        rows=rows,
        database=database,
```

```
        error=error,
    ), status
```