Nom complet : .....

Code permanent : .....

Examen Final INF5190 - Automne 2019

Directives

1. Écrivez vos réponses directement sur l'examen.
2. Aucune documentation n'est permise à l'exception d'une feuille de note recto-verso.
3. Placez votre carte étudiante sur votre bureau. Le surveillant va passer pour la valider.
4. **Fermez et rangez votre téléphone cellulaire.** Tout contact avec votre téléphone cellulaire ou tout autre appareil électronique sera considéré comme une tentative de plagiat.
5. L'examen comporte 7 questions sur 15 pages pour un total de 100 points.

1. Voici un fichier `Dockerfile` :

```
FROM python:3.7-alpine

COPY poll /poll
RUN apk add gcc musl-dev python3-dev libffi-dev openssl-dev
RUN pip install peewee flask pymysql cryptography
EXPOSE 5000
VOLUME /poll/data
ENV DB_PATH
CMD FLASK_APP=poll flask run –-host=0.0.0.0
```

Expliquez ce que chaque option de l'exemple ci-haut font. **6 points (2 points par question)**

a) `COPY poll /poll`

.....

.....

.....

.....

.....

b) `EXPOSE 5000`

.....

.....

.....

.....

.....

c) `VOLUME /poll/data`

.....

.....

.....

.....

.....

2. Avec une image bâtie à partir du `Dockerfile` de la question #1, je lance le conteneur avec la commande suivante :

   `docker run -p 80:5000 --volume /home/poll/data:/poll/data`

   Expliquez ce que font les paramètres de la commande :

   a) `-p 80:5000`

      2 points

      .....

      .....

      .....

      .....

      .....

      .....

      .....

      .....

      .....

   b) `--volume /home/poll/data:/poll/data`

      2 points

      .....

      .....

      .....

      .....

      .....

      .....

      .....

      .....

      .....

3. Voici un extrait d'un `Dockerfile` :

```
ENV DB_NAME poll
ENV DB_USER poll_flask
ENV DB_PASSWORD password
ENV DB_HOST 10.65.22.13
ENV DB_PORT 5432
```

Et un extrait d'une application Flask :

```
import os
from peewee import Model, MySQLDatabase
class BaseModel(Model):
    class Meta:
        database = MySQLDatabase(os.environ['DB_NAME'], {
            "user": os.environ['DB_USER'],
            "password": os.environ['DB_PASSWORD'],
            "host": os.environ['DB_HOST'],
            "port": os.environ['DB_PORT']
        })
```

a) Je lance un conteneur à partir de l'image Docker de l'application Flask décrite ci-haut avec la commande suivante :

`docker run --env DB_NAME=poll_test –env DB_HOST=192.168.2.55`

Quelles seront les informations de connexion à la base de données (nom de la base de données, nom d'usager, mot de passe, hôte et port) que l'application va utiliser?

5 points

.....

.....

.....

.....

.....

.....

.....

.....

.....

b) Expliquez pourquoi combiner de la configuration (e.g. : les informations de connexion à une base de données) dans du code n'est pas une bonne pratique. **10 points**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

4. Voici un exemple de code Python résilient à des défaillances causées par des appels API distants. **15 points (5 points chaque)**

```
1  def get_connected_user(self):
2      retries = 0
3      while retries < 5:
4          try:
5              # Cette appel d'API est problématique, le temps moyen
6              # d'exécution est d'environs une seconde, des fois plus.
7              # Je limite donc à 3 secondes le temps pour l'appel d'API
8              user = User.get(id=request.user_id, timeout=3)
9              user.logged_in = True
10             user.guest = False
11             return user
12         except ConnectionError:
13             sleep((2 ** retries) * 0.1)
14             retries += 1
15     return User(id=None, logged_in=False, guest=True)
```

Cet exemple de code comprend les patrons de résilience suivants : délais maximal, algorithme de recul exponentiel et défaillance partielle (`graceful failure`). Pour chacun de ces patrons, expliquez **1) ce que le patron fait** et **2) ce qu'il permet de mitiger**.

a) Délais maximal

.....

.....

.....

.....

.....

.....

.....

.....

b) Algorithme de recul exponentiel

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

c) Défaillance partielle (`graceful failure`)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

5. L’exemple du code Python de la question #4 ne contient pas le patron de résilience disjoncteur (*circuit breaker*).

   Expliquez ce qu’est le patron disjoncteur et ce qu’il permet de mitiger dans le contexte d’une application Web. **10 points**

6. Déploiements progressif et en canaries.

   a) Quelle est la différence majeure entre un déploiement progressif et un déploiement en canarie ? **5 points**

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   b) L'un des problèmes de ces deux types de déploiement est, qu'à tout moment, plusieurs version différentes de l'application Web peuvent être en ligne. Expliquez les problèmes que cela peut causer et comment il est possible d'éliminer ce risque.

   5 points

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   .....

7. Vous êtes embauché par la Bibliothèque et Archives nationales du Québec (BanQ) pour mettre en place une application Web sous forme d'une API REST. Cette API doit permettre de :

   a) Obtenir les informations d'un livre. Un livre est identifié par un un identifiant unique (*id*).

   b) Créer et récupérer un membre de la bibliothèque. Un membre est identifié par un courriel unique.

   c) Effectuer une recherche par mot-clé qui retourne une liste de livres.

   d) Obtenir la liste des livres réservés par un membre.

   e) Obtenir la liste des membres qui ont réservé un livre.

   f) Effectuer la réservation d'un livre pour un membre.

   g) Terminer une réservation lorsqu'un membre retourne un livre.

   L'application est responsable de la gestion des membres et des réservations de la bibliothèque. Par contre, la gestion des livres (l'ajout, la suppression, la recherche, etc) se fait par un système externe sur lequel vous n'avez aucun contrôle. Cela veut dire que la recherche de livres par mot-clé et l'obtention des informations d'un livre doit se faire en appelant cette API distante externe. Cette API distante prend plusieurs secondes à chercher par mot-clé les millions de livres que la bibliothèque possède. Récupérer les informations d'un livre à partir de son identifiant unique tout aussi lent et nécessite élégamment un appel distant qui peut prendre jusqu'à une seconde.

   Voici un exemple de code qui appelle l'API distant (pour la question 7D) :

```
import json
from urllib.request import urlopen
@app.route("/livres/<int:livre_id>")
def get_livre(livre_id):
    url = f"https://api.banq.qc.ca/livre/{livre_id}"
    with urlopen(url) as response: # temps moyen : 1 seconde
        raw_data = response.read()
    return json.loads(raw_data)
```

a) Identifiez et décrivez toutes les ressources nécessaires pour bâtir l'API REST. **5 points**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

b) À partir des ressources identifiées à la question précédente, identifiez toutes les routes de cette API REST. Une route doit contenir la méthode HTTP, ainsi que l'URI. L'URI peut contenir une variable, tel que vu dans les laboratoires et dans le travail de session.

5 points

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

c) Identifiez les métriques opérationnelles qui vous permettent de valider que votre application fonctionne comme prévu et expliquez pourquoi vous pensez qu'elles sont pertinentes.

10 points

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

d) La spécification précise que la récupération des informations d'un livre se fait à partir d'une API externe. Appeler cette API externe peut prendre jusqu'à une seconde avant de retourner la réponse (voir exemple de code à la page 10). Heureusement, les informations d'un livre (auteur, titre, etc) ne changent jamais. Expliquez ce que vous pouvez mettre en œuvre pour améliorer la performance de votre application qui communique avec cette API distant. **10 points**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

e) Albert est un développeur dans l'équipe d'opérations de la BanQ. Décrivez les étapes qu'Albert doit suivre pour déployer votre système en ligne. Justifiez votre réponse.

10 points

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....