INF5190 - Révision pour l'examen final

Jean-Philippe Caissy

27 novembre

Évaluation des enseignants

- Vous avez jusqu'au 2 décembre pour soumettre mon évaluation du cours
- En date de mardi soir le 26, seulement 8 répondants sur 45.

L'évaluation de l'enseignement contribue à la qualité de la formation offerte par notre université et favorise, par le fait même, l'apprentissage de nos étudiantes et étudiant

Examen intra

- Dans deux semaines : mercredi le 11 décembre 2019
- 3 heures : 18h à 21h
- 2 salles :
  - SH-2560
  - ? (à confirmer)
- Feuille de note 8.5" x 11" recto-verso, aucun autre matériel permis
- L'examen va couvrir toute la matière de la session et les laboratoires
- L'objectif de l'examen final est de démontrer la compréhension de la matière
  - Établir des liens entre les différents concepts vu en classe

14 - Virtualisation

- Qu'est-ce que la virtualisation?
  - Concepts clés : hôte (*host*) et invité (*guest*)
  - Virtualisation complète et paravirtualisation
- Quel est l'utilité de la virtualisation dans le contexte d'une application Web?
- En quoi la virtualisation peut être plus sécuritaire?

14 - Conteneurs

- Pourquoi utiliser des conteneurs pour une application Web?
  - Pourquoi utiliser des conteneurs et non pas de la virtualisation complète?
- Que se passe-t-il lorsqu'on lance une application dans un conteneur?
  - Quelles ressources sont partagées, qu'est-ce qui est isolé?
- Une application Web utilise Redis, Postgres et un serveur HTTP. Quelle est la différence entre l'utilisation de conteneurs et la virtualisation complète?

15 - Déploiement

- Le modèle à 4 couches (*4-tiers*) : qu'est-ce que c'est?
  - Comment est-ce qu'un tel modèle s'applique dans le contexte de l'intégration continue?
- Types d'hébergement :
  - Serveur physique
  - Serveur virtuel
  - IaaS (*Infrastructure as a Service*)
  - PaaS (*Platform as a Service*)
- Donnez un exemple de déploiement d'une application Web sur l'un des types d'hébergements mentionnés
- Quel est l'utilité de Docker Compose? Si vous aviez à l'utiliser pour déployer, comment est-ce que ça fonctionnerait?

15 - Déploiement

- Lors d'un déploiement, à quoi sert l'orchestration des conteneurs?
- À quoi sert le déploiement progressif? Quelle est la différence avec le déploiement canarie?
  - Expliquez le principe des déploiements en canarie
  - Qu'est-ce que le déploiement canarie nous permet de faire?
- Vous travaillez sur un système utilisant un déploiement progressif. Votre code introduit une nouvelle route dans l'application qui est accessible à partir d'une nouvelle page.
  - Quel est le risque de votre déploiement? Comment y remédier?

16 - Maintenance

- Quelles sont les manières de détecter des défaillances dans une application Web?
- Lorsqu'une défaillance est détectée, qu'est-ce qu'il devrait se passer?

16 - Maintenance

- Il existe plusieurs types de métriques opérationnelles. Qu'est-ce que les métriques suivantes nous indique sur l'application ?
  - Le temps de réponse moyen de l'application Web
  - Le nombre de requête HTTP par minute sur l'application Web
  - L'espace disque utilisé par la base de donnée:w
- Pour chacune des métriques ci-haut, comment établir une valeur problématique?

16 - Maintenance

- Vous gérer une application Web qui répond à une API REST. Un utilisateur soumet un bug sur votre Github comme quoi une route de l'API retourne toujours un code HTTP 500. Vous n'êtes pas au courant de cela.

Quelles étapes prendrez-vous pour régler cette situation?

16 - Maintenance

- Dans le modèle de gestion SRE (*Site Reliability Engineer*), il existe le principe d'accepter le risque.
  - Accepter le risque sous-entend que l'application va éventuellement être défaillant. Qu'est-ce que ça veut dire?
    - Est-ce une bonne chose? Pourquoi?
- Pourquoi est-ce que le modèle SRE met l'emphase sur l'élimination des tâches lourdes?
- Quel est le lien entre la surveillance des systèmes et les SLO (*Service Level Objective*)?

17 - Résilience

```python
@app.route("/login", methods=["POST"])
def login():
    user = User.retrieve_with(
        email=request.form['email'],
        password=request.form['password']
    )

    if user:
        # opérations de création de session
        return redirect(url_for("profile"))
    else:
        return "Le compte n'existe pas"
```

La méthode `retrieve_with` fait un appel distant à un API. Cet appel d'API prend en moyenne une seconde.

- Comment rendre cette méthode plus résiliente à des problèmes réseaux?

17 - Résilience

- Qu'est-ce que la redondance, et pourquoi est-ce que cela permet de rendre une application Web resiliente?
- À quoi sert l'utilisation d'un algorithme de recul exponentiel?
- Qu'est-ce que le patron disjoncteur (*circuit breaker*) dans le contexte d'une API Rest? Comment est-ce utile?
  - Comment est-ce que vous intégreriez les délais maximums (*timeout*) avec ce patron?
- Quel est l'un des désavantages majeur de l'utilisation de tâches en arrière plan?