A24 – Déploiement sur l'infonuagique - Travail pratique 1 (14%)

Objectifs du TP

Ce travail pratique (TP) vise à évaluer votre compréhension des notions vues en cours à savoir :

- Déployer et gérer des applications avec Azure App Service.

Contexte

Ce travail peut être réalisé individuellement ou par groupe de 2 étudiants au maximum. La remise doit se faire à partir de LEA. Vous devez remettre un document word ou PDF contenant l'exercice 1 et les captures d'écran demandées pour les autres exercices.

Date de remise

Votre travail doit être remis au plus tard le dimanche 17 novembre 2024 à 23h59.

Critères d'évaluation

Votre travail doit respecter l'ensemble des critères suivants :

- Vous devez choisir les options permettant d'optimiser au maximum les couts;
- Les captures d'écran doivent être suffisamment complètes pour évaluer la réalisation du travail demandé;
- Les applications dans Azure doivent être fonctionnelles;
- -10 % par jour de retard;
- Note de 0 si le travail est remis après le retour à l'ensemble du groupe ou si le travail a été plagié en tout ou en partie.

Grille d'évaluation

	Excellent	Fonctionnel	Minimal	Insuffisant
Capacité 1 : Utiliser des services infonuagiques	**Utiliser et configurer App Service :** • Utilisation adéquate en tout temps de la mise à l'échelle; • Utilisation adéquate en tout temps des paramètres de configuration; • Utilisation adéquate en tout temps des emplacements de déploiement	**Utiliser et configurer App Service :** • Utilisation presque adéquate de la mise à l'échelle; • Utilisation presque adéquate des paramètres de configuration; • Utilisation presque adéquate des emplacements de déploiement	**Utiliser et configurer App Service :** • Utilisation partiellement adéquate de la mise à l'échelle; • Utilisation partiellement adéquate des paramètres de configuration; • Utilisation partiellement adéquate des emplacements de déploiement	**Utiliser et configurer App Service :** • Utilisation rarement adéquate de la mise à l'échelle; • Utilisation rarement adéquate des paramètres de configuration; • Utilisation rarement adéquate des emplacements de déploiement
Capacité 2 : Déployer sur l'infonuagique	**Déployer des ressources:** • Création des ressources en tenant compte en tout temps des enjeux de couts; • Déploiement des applications en tenant parfaitement compte des besoins.	**Déployer des ressources:** • Création des ressources en tenant presque toujours compte des enjeux de couts; • Déploiement des applications en tenant toujours compte des besoins.	**Déployer des ressources :** • Création des ressources en tenant presque toujours compte des enjeux de couts; • Déploiement des applications en tenant presque toujours compte des besoins.	**Déployer des ressources :** • Création des ressources en tenant rarement compte des enjeux de couts; • Déploiement des applications en tenant rarement compte des besoins.

Partie 1 : 4 points

L'entreprise **HealthPlus** développe des applications web et API pour aider les établissements de santé à gérer les informations des patients, la logistique des équipements médicaux, et les dossiers médicaux confidentiels. Le déploiement doit être réalisé sur **Azure App Service** avec des configurations spécifiques pour chaque application.

Détails des applications et leurs exigences

1. **API Gestion des patients**
    - Besoin : Doit pouvoir créer jusqu'à 4 instances en cas de besoin. Mise à l'échelle manuelle requise.
    - Exigence particulière : Application accessible publiquement, mais doit être scalable pour des pics de demande.
2. **API Gestion des équipements médicaux**
    - Besoin : Exécution longue durée, souvent plus de 90 minutes par jour pour surveiller l'inventaire.
    - Exigence particulière : Aucune.
3. **Application web Dossier médical électronique** (DME)
    - Besoin : Déployer avec zéro temps d'arrêt. Scalabilité automatique pour gérer des pics de consultations (jusqu'à 6 instances).
    - Exigence particulière : Haute disponibilité.
4. **API Archivage des documents**
    - Besoin : Application critique avec 8 backups quotidiens et un coût maximum de **350 $ par mois**.
    - Exigence particulière : Application hautement disponible, mais avec budget limité.
5. **Application Web SantéPlus**
    - Besoin : Domaine personnalisé, 120 ACU minimum et 3 Go de mémoire.
    - Exigence particulière : Performance élevée et personnalisation de l'URL.
6. **Intranet Gestion des RH**
    - Besoin : Accès limité au réseau privé de HealthPlus pour sécurité des données sensibles. Besoin minimum de 2 processeurs virtuels et 10 Go de mémoire.
7. **Intranet Gestion des dossiers de santé**
    - Besoin : Accès restreint au réseau privé de HealthPlus, à très haute sécurité et confidentialité.

8. **Application de télémédecine**
    - Besoin : Scalabilité automatique jusqu'à 5 instances, avec déclenchement géré par la plateforme Azure.
    - Exigence particulière : Performance élevée et zéro interruption de service.

Dans un tableau ayant la structure ci-dessous, recommandez un plan tarifaire pour chaque application et justifiez votre choix. Vous devez proposer le plan minimum permettant de répondre au besoin.

Nom de l'application	Plan tarifaire	Justification
Application xxx		

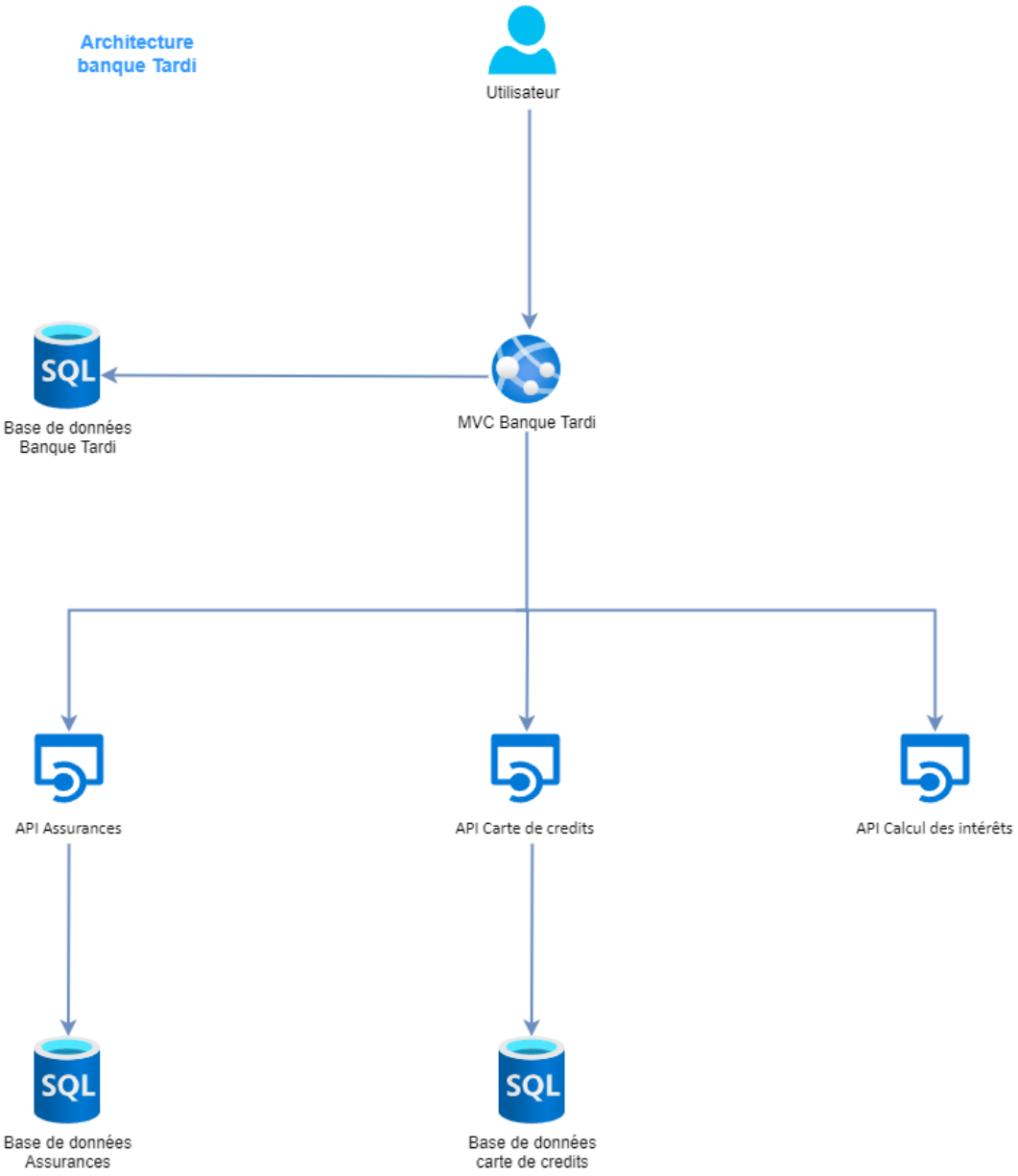
Partie 2 : 10 points

Mise en contexte

La banque Tardi aimerait déployer ses applications développées en microservices en utilisant Azure App Service.

Le code de l'application est joint avec l'énoncé du TP.

Ci-dessous l'architecture de l'application :



Exercice 1 : 4 points

1. À partir du portail Azure, créez les services nécessaires pour exécuter les applications, en tenant compte de ce qui suit :
    a. L'application MVC et l'API Calcul des intérêts doivent utiliser le même plan tarifaire. Ce dernier doit permettre de disposer de la mise à l'échelle automatique et des emplacements de déploiement;
    b. Les autres applications doivent avoir le plan gratuit et utiliser le même plan.
2. Configurez l'application MVC afin qu'elle puisse communiquer avec les API;
3. Déployez vos applications dans les ressources créées en utilisant Visual Studio et assurez-vous que tout fonctionne correctement

4. Configurez les API afin d'afficher l'interface de Swagger dans le navigateur

Exercice 2 : 2 points

1. Apportez les configurations nécessaires à l'API Calcul des intérêts afin que les déploiements se fasse avec zéro temps d'arrêt.
2. Pour l'application MVC, créez un emplacement de déploiement Staging et configurez la variable d'environnement ASPNETCORE_ENVIRONMENT avec respectivement pour valeur « Production » et « Staging » pour les slots « Production » et « Staging » de l'application MVC.

Exercice 3 : 4 points

Mettez en place la mise à l'échelle automatique pour l'application MVC en tenant compte de ce qui suit :

- Une nouvelle instance doit être automatiquement provisionnée en cas de consommation du processeur supérieure à 80%. Le nombre maximal d'instances doit être limité à 4.
- En consultant les journaux de l'application MVC, on s'est rendu compte qu'elle recevait beaucoup de trafic toutes les fins de semaine (samedi et dimanche), de 7h à 18h et que 4 instances étaient parfois insuffisantes pour répondre à la montée en charge durant cette période. Mettez en place une autre condition de mise à l'échelle automatique qui permettra pendant cette période de bénéficier d'un maximum de 8 instances.

NB :

1. Vous devez fournir des captures d'écran pour chacune des configurations effectuées. La capture doit permettre d'identifier le compte qui fait la configuration;
2. Vous devez fournir l'URL de l'application MVC et de toutes les APIs;
3. Vous devez ramener le plan tarifaire de l'application MVC a gratuit et attendre mon retour avant de supprimer vos ressources.

Bon travail!