Dimanche, 22 Octobre 2023

Exposé sur l'Architecture de Microservices pour le Système de Gestion de l'Éducation Universitaire

1. Introduction

Contexte: Le domaine de l'éducation universitaire englobe une multitude d'interactions et de processus, des admissions à l'évaluation des étudiants. Pour gérer efficacement ces interactions, un système modulaire, évolutif et fiable est requis.

Objectif: Mettre en place une architecture de microservices qui facilite le développement, le déploiement et la gestion des diverses fonctionnalités du système universitaire.

2. Microservices Principaux

2.1. Service d'Admission

- Gère l'inscription des étudiants, les candidatures et l'enregistrement des pièces justificatives.
- **Base de données** : Relationnelle (pour stocker les détails des étudiants et des candidatures).

2.2. Service des Cours

- Gestion, planification des cours et attribution des ressources.
- **Base de données** : NoSQL (pour une structuration flexible des cours et des horaires).

2.3. Service d'Évaluation

- Examen, notes et évaluations des étudiants.
- **Base de données** : Relationnelle (pour la précision et l'intégrité des notes).

2.4. Service des Utilisateurs

- Profils des étudiants, des enseignants et du personnel administratif.
- **Base de données** : Relationnelle (pour les relations entre les étudiants, les cours et les enseignants).

2.5. Service de Communication

- Portail d'annonces, système de messagerie intégré.
- **Base de données** : NoSQL (pour la flexibilité des formats de communication).

3. Infrastructure et Communication

3.1. API Gateway

- Sert de point d'entrée pour toutes les demandes externes et gère l'authentification, la limitation du débit et le routage vers les microservices appropriés.

3.2. Service de Découverte

- Permet aux microservices de découvrir et de communiquer entre eux.

3.3. Communication Asynchrone

- Utilisation de **Kafka** ou **RabbitMQ** pour des événements tels que les notifications, les annonces et la mise à jour des données.

4. Résilience et Gestion des Failles

4.1. Circuit Breaker

- Prévenir la cascade des défaillances entre les services.

4.2. Strategy de Backup

- Chaque microservice est responsable de sauvegarder régulièrement sa base de données pour prévenir les pertes de données.

5. Sécurité

- Un service d'authentification centralisé pour gérer l'accès aux différents microservices.
- Chiffrement des communications et des données sensibles.

6. Surveillance et Maintenance

- Utilisation de **Prometheus** et **Grafana** pour la surveillance en temps réel des microservices.
- Centralisation des logs avec l'ELK Stack pour faciliter le dépannage.

Service d'Admission vs. Service des Utilisateurs

Service d'Admission:

Objectif principal: Gérer le processus d'admission des étudiants, de la réception des candidatures à l'inscription effective.

Responsabilités:

1. **Réception des Candidatures** : Accepter les demandes d'admission des étudiants potentiels.
2. **Vérification des Documents** : S'assurer que tous les documents nécessaires sont fournis (par exemple, diplômes, attestations, etc.).
3. **Évaluation des Candidatures** : Selon les critères définis par l'université, évaluer les demandes et décider des admissions.
4. **Notification** : Informer les candidats de la décision (accepté, en attente, rejeté).
5. **Inscription** : Une fois accepté, procéder à l'inscription formelle de l'étudiant dans l'université.

Utilisateurs: Administrateurs ou personnel dédié à la gestion des admissions.

Base de Données: Stocke les candidatures, les documents, l'état de chaque candidature et les informations sur les étudiants nouvellement inscrits.

Service des Utilisateurs:

Objectif principal: Gérer les profils et les informations d'authentification de tous les utilisateurs du système.

Responsabilités:

1. **Création de Profil** : Établir un profil pour chaque utilisateur (étudiants, enseignants, personnel administratif).
2. **Authentification** : Gérer les identifiants de connexion, la vérification des mots de passe et la génération de jetons d'accès.
3. **Gestion des Droits** : Attribuer et gérer les rôles ou les permissions de chaque utilisateur en fonction de leur statut (par exemple, un étudiant ne peut pas attribuer de notes, un professeur ne peut pas inscrire un nouveau cours, etc.).
4. **Mise à jour de Profil** : Permettre aux utilisateurs de mettre à jour leurs informations, de changer de mot de passe, etc.

Utilisateurs: Tous ceux qui accèdent au système : étudiants, enseignants, administrateurs, etc.

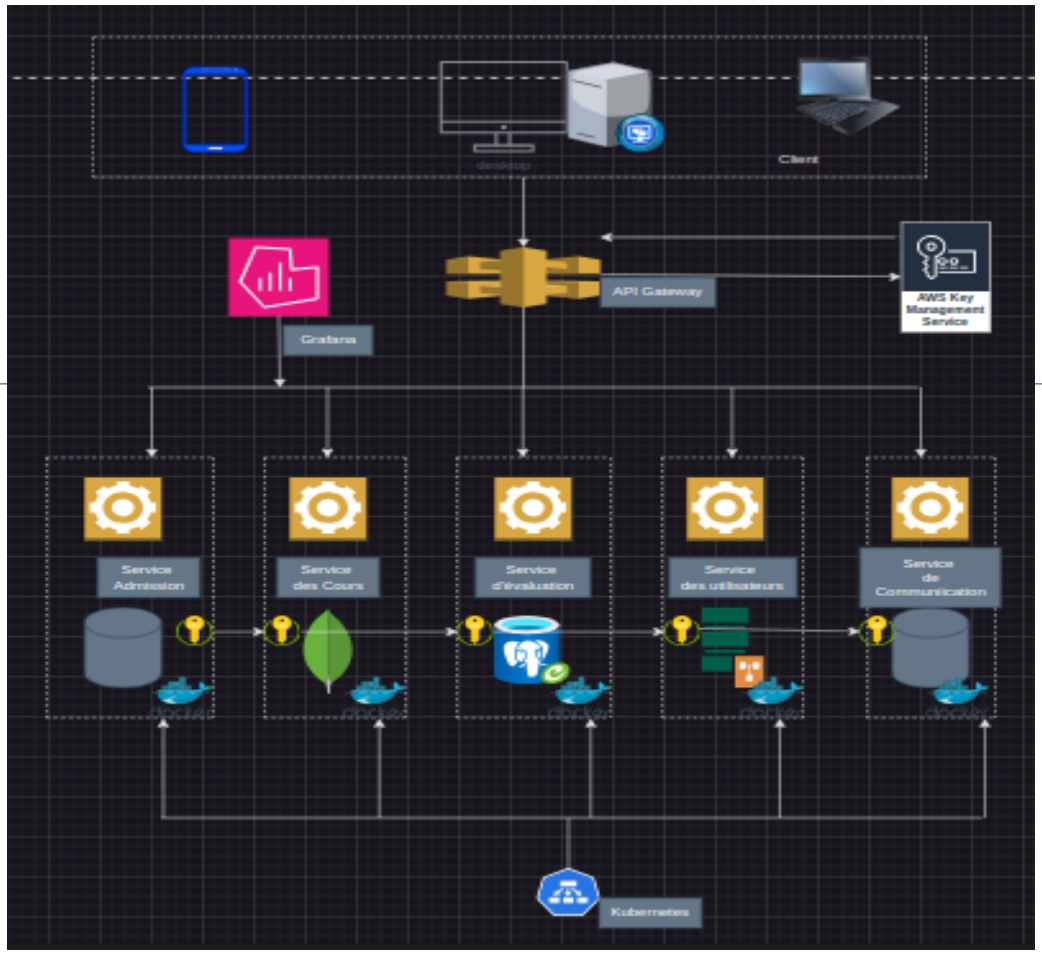
Base de Données: Stocke les informations du profil, les détails d'authentification, les rôles et les permissions.

En séparant clairement ces services, vous évitez la redondance et assurez que chaque service ne s'occupe que de sa propre responsabilité. Le Service d'Admission est axé sur le processus

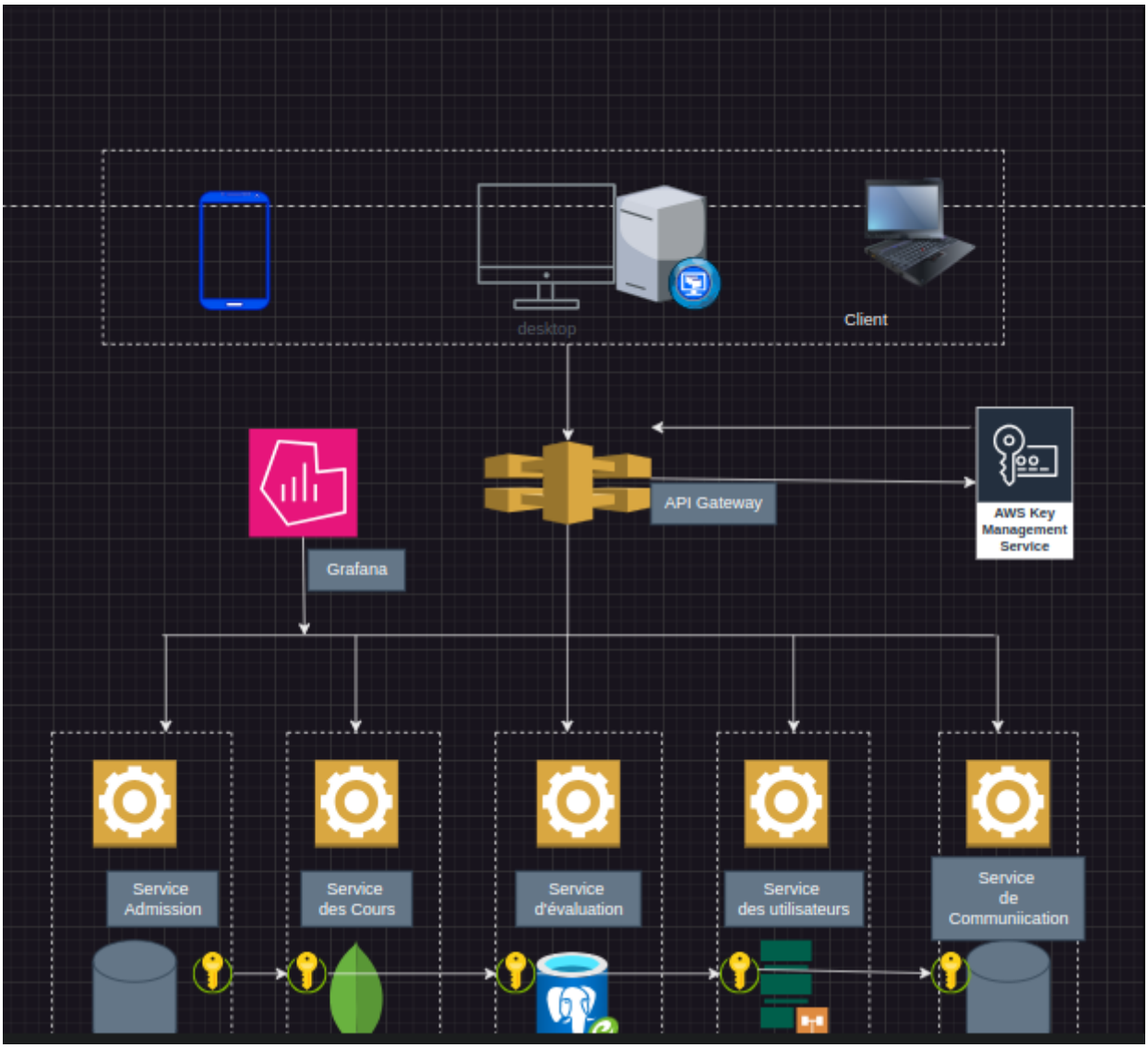
administratif d'admission des étudiants, tandis que le Service des Utilisateurs s'occupe de la gestion des profils et de l'authentification de tous les utilisateurs, quel que soit leur rôle dans le système. Cette distinction claire favorise la modularité et la maintenabilité de votre système.

7. Conclusion

L'adoption d'une architecture de microservices pour le système de gestion de l'éducation universitaire permet une grande modularité, une meilleure évolutivité et la capacité de répondre rapidement aux besoins changeants du domaine universitaire. Il s'agit d'une approche moderne qui, bien que complexe, offre des avantages significatifs en termes de développement, de déploiement et d'exploitation.




Management
Service
Grafana
Service
Admission
Service
des Cours
Service
d'évaluation
Service
des utilisateurs
Service
de
Communiication


desktop
Client
API Gateway
AWS Key
Management
Service
Grafana
Service
Admission
Service
des Cours
Service
d'évaluation
Service
des utilisateurs
Service
de
Communiication