Prosit 1 :

**1. Architecture Logique**

Composants Principaux :

Interface Utilisateur (Front-end) :

Application Web : Permet aux candidats et recruteurs d’interagir avec le système. Développée en utilisant des technologies comme React, Angular ou Vue.js.

Application Mobile : Version mobile du job board pour les utilisateurs sur smartphones et tablettes.

Serveur d'Application (Back-end) :

API RESTful : Fournit des points d'accès pour les fonctionnalités du job board (inscription, gestion des annonces, recherche, etc.).

Microservices :

Service d'Authentification : Gère les inscriptions et les connexions.

Service de Gestion des Annonces : Gère la publication, la mise à jour et la suppression des annonces d'emploi.

Service de Candidature : Gère les candidatures des utilisateurs aux offres d'emploi.

Service de Connexion Sociale : Intègre les données des réseaux sociaux et les informations des CV pour l'évaluation des candidatures.

Service de Recommandations : Analyse les données pour proposer des recommandations personnalisées aux recruteurs et candidats.

Base de Données :

Base de Données Relationnelle : Stocke les informations structurées comme les utilisateurs, les annonces et les candidatures (par exemple, PostgreSQL, MySQL).

Base de Données NoSQL : Pour les données non structurées et les informations de réseau social (par exemple, MongoDB, Cassandra).

Intégration avec Réseaux Sociaux :

API des Réseaux Sociaux : Connexion aux réseaux sociaux comme LinkedIn, Facebook pour récupérer les informations des candidats et des recruteurs.

Système de Notification :

Notification en Temps Réel : Notifications pour les nouvelles offres d'emploi, les candidatures reçues, etc.

Système de Recherche :

Moteur de Recherche : Permet des recherches rapides et efficaces sur les annonces et les profils des candidats

Flux de Travail :

Inscription et Connexion :

Les candidats et recruteurs s'inscrivent via le front-end, et les données sont traitées par le service d'authentification.

Gestion des Annonces :

Les recruteurs publient des annonces via le front-end, qui sont traitées par le service de gestion des annonces et stockées dans la base de données.

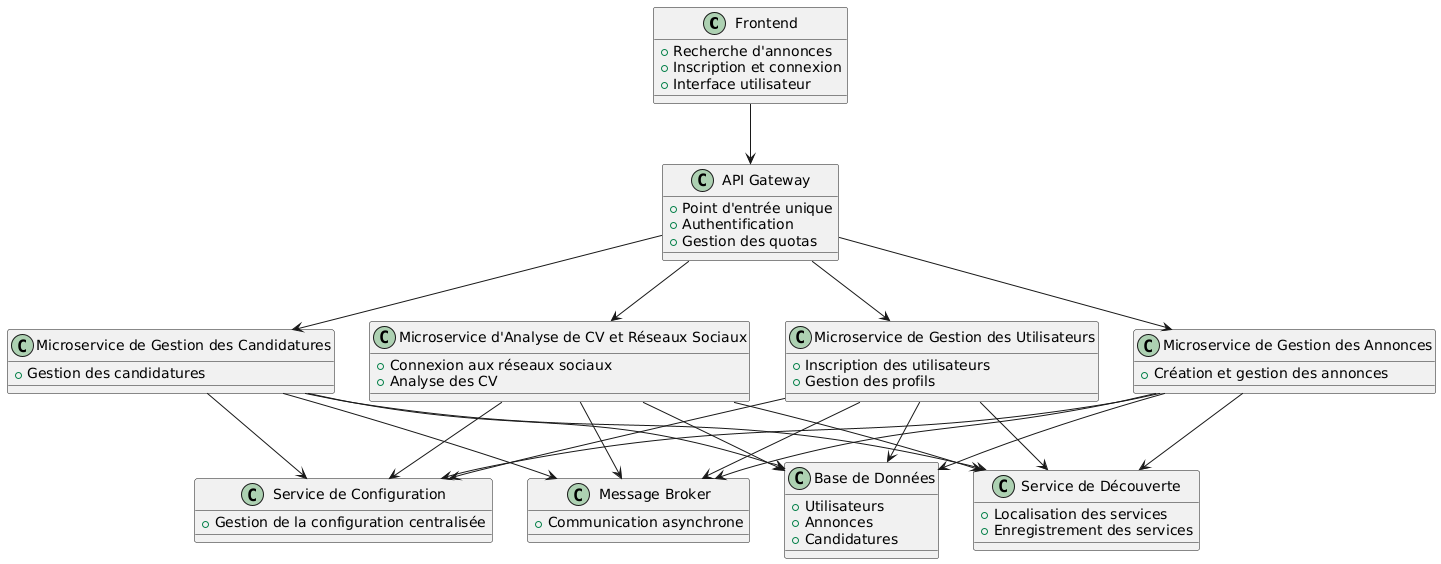
Candidature :

Les candidats postulent via le front-end, et les candidatures sont gérées par le service de candidature.

Évaluation et Recommandations :

Le service de connexion sociale et le service de recommandations analysent les données pour faire correspondre les candidats avec les annonces.

Diagrammes de composants :



**2. Architecture Physique**

2.1 Infrastructure :

Serveurs :

Description : Hôtes physiques ou virtuels pour déployer les microservices, les bases de données, et les composants de l'API Gateway.

Technologies : Serveurs physiques, VMs, ou instances cloud (AWS EC2, Google Cloud Compute Engine).

Conteneurisation et Orchestration :

Description : Utilisation de conteneurs pour empaqueter les microservices et d'un orchestrateur pour gérer leur déploiement.

Technologies : Docker pour la conteneurisation et Kubernetes pour l'orchestration.

Load Balancer :

Description : Répartit le trafic entre plusieurs instances de microservices pour assurer la haute disponibilité et la scalabilité.

Technologies : AWS ELB, Nginx.

CDN (Content Delivery Network) :

Description : Accélère la livraison des contenus statiques comme les images et les scripts.

Technologies : Cloudflare, AWS CloudFront.

2.2 Intégration avec les Réseaux Sociaux :

API des Réseaux Sociaux :

Description : Connecte avec les plateformes sociales pour extraire des données pertinentes et analyser les liens entre les candidats et les entreprises.

Technologies : APIs de Facebook, LinkedIn, Twitter.

Analyse de CV :

Description : Utilise des algorithmes et des outils d’analyse pour extraire et comparer les compétences et expériences des candidats avec les offres d’emploi.

Technologies : NLP (Natural Language Processing) pour l'analyse des CV.

Réponses aux Contraintes

Trouver des Liens entre la Société et les Chercheurs d'Emploi :

Approche : Utilisez des APIs pour intégrer des données des réseaux sociaux dans le système. Analysez les CV et les profils des réseaux sociaux pour identifier des connexions pertinentes comme des parcours scolaires communs ou des connexions professionnelles.

Exemple : Lorsqu'un candidat postule à une offre, le système peut vérifier si le candidat a des connexions avec des employés de l'entreprise via LinkedIn.

Scalabilité :

Approche : Déployez les microservices dans des conteneurs Docker orchestrés par Kubernetes pour gérer les mises à l'échelle horizontales. Utilisez des load balancers pour distribuer le trafic.

Exemple : Si le nombre de requêtes augmente, vous pouvez ajouter plus d'instances des microservices en fonction des besoins.

Haute Disponibilité :

Approche : Déployez les microservices en plusieurs instances dans différentes zones géographiques pour éviter les points de défaillance unique. Utilisez des mécanismes de redondance pour les bases de données.

Exemple : Les microservices peuvent être déployés dans différentes régions AWS pour garantir la disponibilité même en cas de panne d'une région.

Tests :

Approche : Mettez en place des tests unitaires, des tests d'intégration et des tests de charge pour chaque microservice. Utilisez des environnements de staging pour simuler des conditions réelles avant la mise en production.

Exemple : Testez la fonctionnalité de recherche d'annonces avec un grand nombre de données pour vérifier les performances du système.

