**COMPTE RENDU**

**Création des Portfolio**

**LAGHDACHE YAAKOUB**

**ABERANE DIA EDDINE**

**1. Introduction**

**Aperçu du Projet**

Ce projet vise à développer une plateforme pour création des portfolio d’un façon efficace et dynamqiue.

**Importance de l'Architecture Microservices**

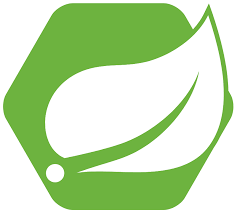
L'architecture microservices, adoptée dans notre projet, offre une gestion flexible et isolée de chaque composant du système. Cette approche décompose le système en services plus petits et indépendants, chacun s'exécutant dans son propre processus. Elle facilite la mise à l'échelle de parties spécifiques du système en réponse à des demandes changeantes, sans affecter l'ensemble de l'infrastructure. De plus, cette isolation améliore significativement la facilité de maintenance, permettant des mises à jour et des corrections individuelles sans perturber les autres services.

Cette architecture favorise ainsi une évolution et une expansion agiles du système, tout en assurant une haute disponibilité et une réactivité accrue aux besoins changeants du projet. La modularité inhérente aux microservices permet une gestion plus efficace des fonctionnalités, des bases de données et des dépendances, offrant ainsi une meilleure flexibilité pour répondre aux évolutions du projet. Elle favorise également la diversité des technologies au sein du système, permettant l'utilisation des outils les plus adaptés à chaque service.

En résumé, l'adoption de l'architecture microservices constitue un choix stratégique pour notre projet, améliorant la flexibilité, la scalabilité, la maintenance et la réactivité, tout en favorisant une évolution agile et une haute disponibilité.

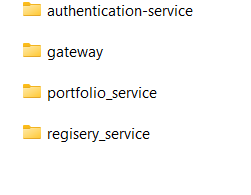
**2. Architecture Microservices**

**Architecture**



En employant **Eureka** comme service de découverte au cœur de notre architecture microservices, nous garantissons une communication fluide et dynamique entre les divers composants. Eureka sert de registre centralisé où chaque microservice s'enregistre et peut localiser d'autres services nécessaires pour fonctionner. Cette intégration permet à chaque composant de découvrir et de communiquer avec les autres services de manière efficace et transparente, sans configuration rigide ou dépendance directe. Cette flexibilité est essentielle pour le maintien d'une architecture réactive et adaptable, facilitant les interactions entre services même en cas d'évolution ou de modification de l'infrastructure sous-jacente. En somme, Eureka contribue à une architecture robuste, capable de gérer dynamiquement la localisation et la disponibilité des services dans un environnement distribué.

Les services :



**4. Conteneurisation avec Docker**

Une image contenant Graphique, dessin humoristique, clipart, conception

Description générée automatiquement

Docker encapsule chaque service de notre architecture microservices, y compris Jenkins, dans des conteneurs distincts. Cette isolation garantit une uniformité de l'environnement d'exécution, quel que soit le déploiement. Docker facilite également l'intégration et le déploiement continus (CI/CD) via Jenkins, en permettant des tests et des déploiements cohérents et automatisés.

**5. CI/CD avec Jenkins**

Une image contenant dessin humoristique, Dessin animé, illustration, clipart

Description générée automatiquement

Jenkins, au cœur de notre système, automatise le pipeline CI/CD, intégrant développement, tests et déploiement. Cet outil clé standardise notre flux de travail, optimisant l'efficacité et la cohérence. Il minimise les erreurs humaines, accélérant ainsi le cycle de vie du développement. Jenkins permet une intégration continue des modifications de code et un déploiement continu des applications, garantissant ainsi une mise en production rapide et fiable des nouvelles fonctionnalités et mises à jour. Cette automatisation assure une réponse agile aux besoins changeants du projet.



