Nom et Prénom :

Nidaazzi Elmehdi

Mghazli Mahdi

Elfarouki Ismail

Classe : 5.IIR.G2

GESTION_CONTACT

Aperçu du Projet de Gestion de Contacts avec Microservices:

Le projet de gestion de contacts avec microservices est une solution logicielle modulaire et évolutive qui vise à optimiser la gestion des informations relatives aux contacts. Il se compose de plusieurs microservices spécialisés, chacun ayant une fonctionnalité distincte. Les principaux microservices inclus dans ce projet sont le **CalendrierService, CategorieService, UtilisateurService, ContactService, EurekaService, et Gateway.**

- **CalendrierService:**

Responsabilités: Gère les fonctionnalités liées à la gestion des calendriers.

Fonctionnalités clés: Création, modification et suppression d'événements, gestion des rappels, synchronisation avec d'autres calendriers.

- **CategorieService:**

Responsabilités: Gère la classification et la catégorisation des contacts.

Fonctionnalités clés: Création et gestion de catégories, attribution de catégories aux contacts.

- **UtilisateurService:**

Responsabilités: Gère les informations relatives aux utilisateurs du système.

Fonctionnalités clés: Création de comptes utilisateur, gestion des autorisations et des rôles, authentification et autorisation.

- **ContactService:**

Responsabilités: Gère les informations détaillées sur les contacts.

Fonctionnalités clés: Ajout, modification, suppression de contacts, recherche et filtrage de contacts, gestion des relations entre contacts.

- **EurekaService:**

Responsabilités: Service de découverte pour la gestion dynamique des instances de microservices.

Fonctionnalités clés: Enregistrement et découverte automatiques des microservices, équilibrage de charge.

- **Gateway:**

Responsabilités: Point d'entrée centralisé pour les requêtes clients, gestion des requêtes et des réponses.

Fonctionnalités clés: Routage des requêtes vers les microservices appropriés, gestion des autorisations, agrégation de données.

Importance de l'Architecture Microservices:

- **Scalabilité:**

La conception modulaire permet une évolutivité indépendante des microservices, facilitant ainsi la montée en charge des composants spécifiques en fonction des besoins.

- **Flexibilité et Agilité:**

Les microservices favorisent le développement agile en permettant des mises à jour indépendantes, accélérant ainsi le déploiement des fonctionnalités.

- **Réutilisabilité:**

Chaque microservice peut être réutilisé dans différents contextes, favorisant une approche de développement orientée composant.

- **Facilité de Maintenance:**

Les microservices sont indépendants les uns des autres, facilitant la maintenance et la correction de bugs sans impacter l'ensemble du système.

- **Technologies Diverses:**

Chaque microservice peut être développé et déployé indépendamment, permettant l'utilisation de différentes technologies adaptées à chaque service spécifique.

2. Architecture Microservices

- **Architecture :**

l'architecture microservices repose sur des services indépendants, découpant les fonctionnalités en services spécifiques, avec des bases de données propres. Les microservices exposent des APIs publiques, facilitant la communication entre eux, et ils peuvent être déployés indépendamment pour des mises à jour sans impact global sur l'application.

- **Description des services :**

CalendrierService: Gère les opérations liées aux calendriers, y compris la création, la mise à jour et la suppression d'événements. Expose des APIs pour intégration avec d'autres services.

CategorieService: Gère la classification des contacts en catégories. Permet la création, la gestion et l'attribution de catégories. Possède des APIs pour la gestion de la catégorisation des contacts.

UtilisateurService: Gère les informations des utilisateurs, y compris la création de comptes, l'authentification et l'autorisation. Expose des APIs pour la gestion des utilisateurs et des rôles.

ContactService: Responsable de la gestion des informations détaillées sur les contacts. Expose des APIs pour l'ajout, la modification et la suppression de contacts, ainsi que la recherche et le filtrage.

EurekaService: Service de découverte qui permet l'enregistrement et la découverte automatiques des instances de microservices. Facilite l'équilibrage de charge et la résilience du système.

Gateway: Point d'entrée centralisé pour les requêtes clients. Gère le routage des requêtes vers les microservices appropriés, la gestion des autorisations et l'agrégation de données.

❖ Mécanismes de communication :

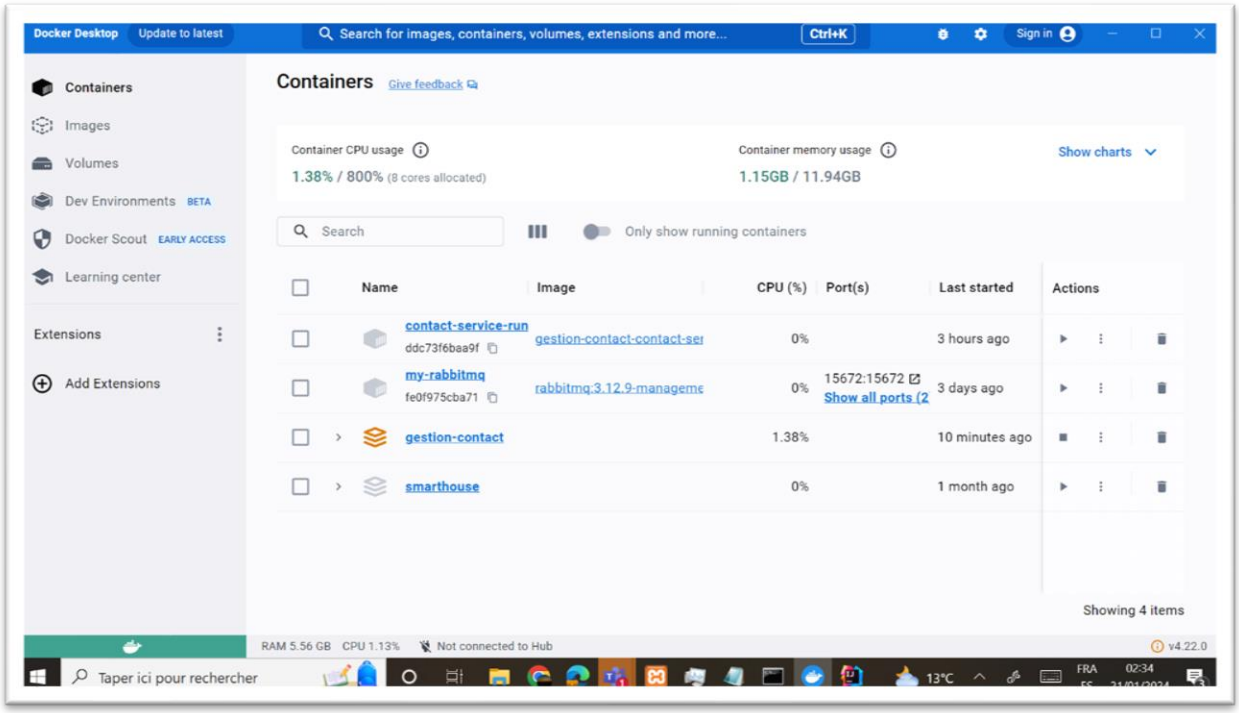
Protocoles de Communication : Les microservices communiquent généralement via des protocoles standard tels que HTTP/HTTPS pour les requêtes RESTful, gRPC pour les appels de procédure à distance (RPC), ou d'autres protocoles basés sur les besoins spécifiques du projet.

API Gateway: L'API Gateway joue un rôle crucial en gérant les requêtes clients, en effectuant le routage vers les microservices appropriés, et en fournissant une couche d'authentification et d'autorisation.

Découverte de Services: Le service de découverte (EurekaService) facilite la découverte automatique des instances de microservices, permettant ainsi une communication dynamique et évolutive entre les services.

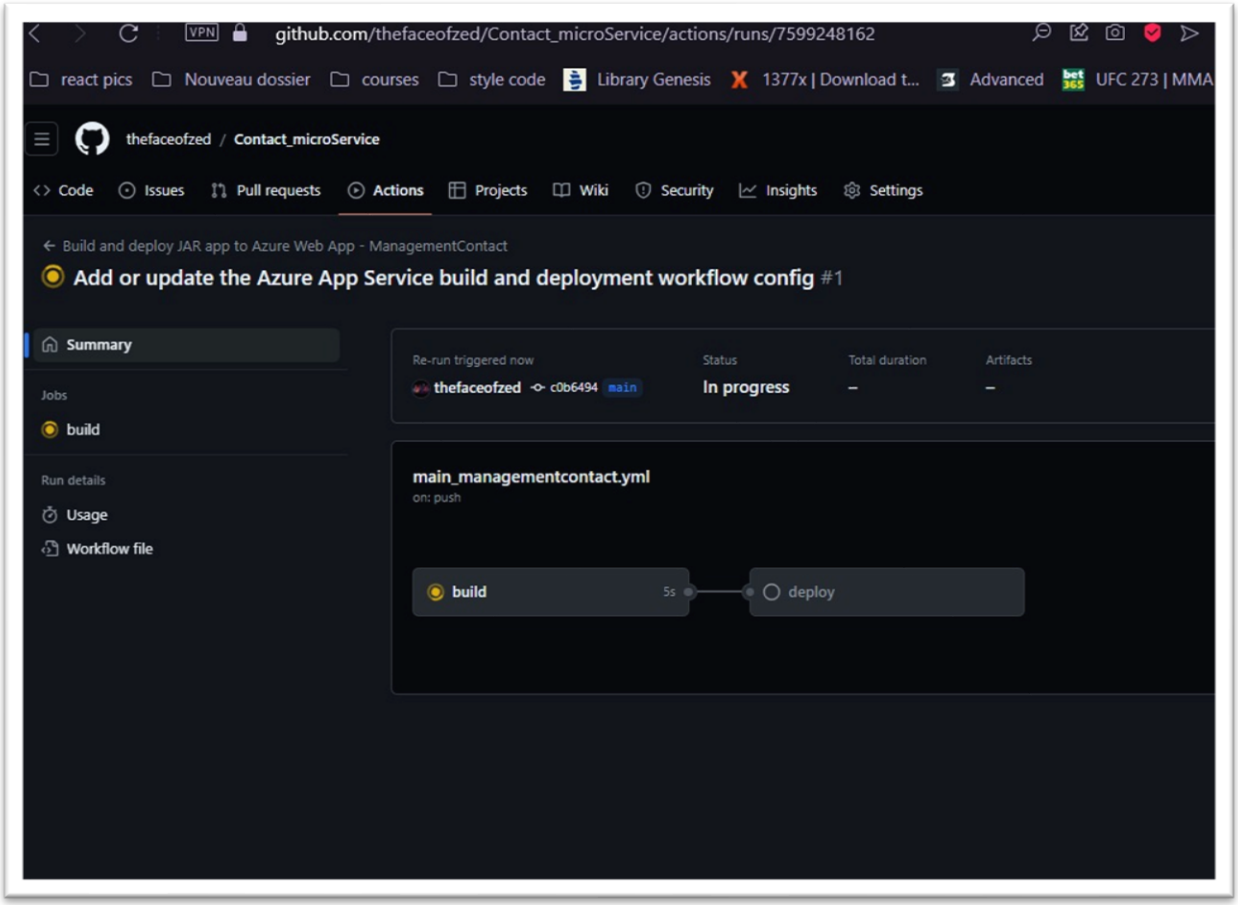
3. Conteneurisation avec Docker :

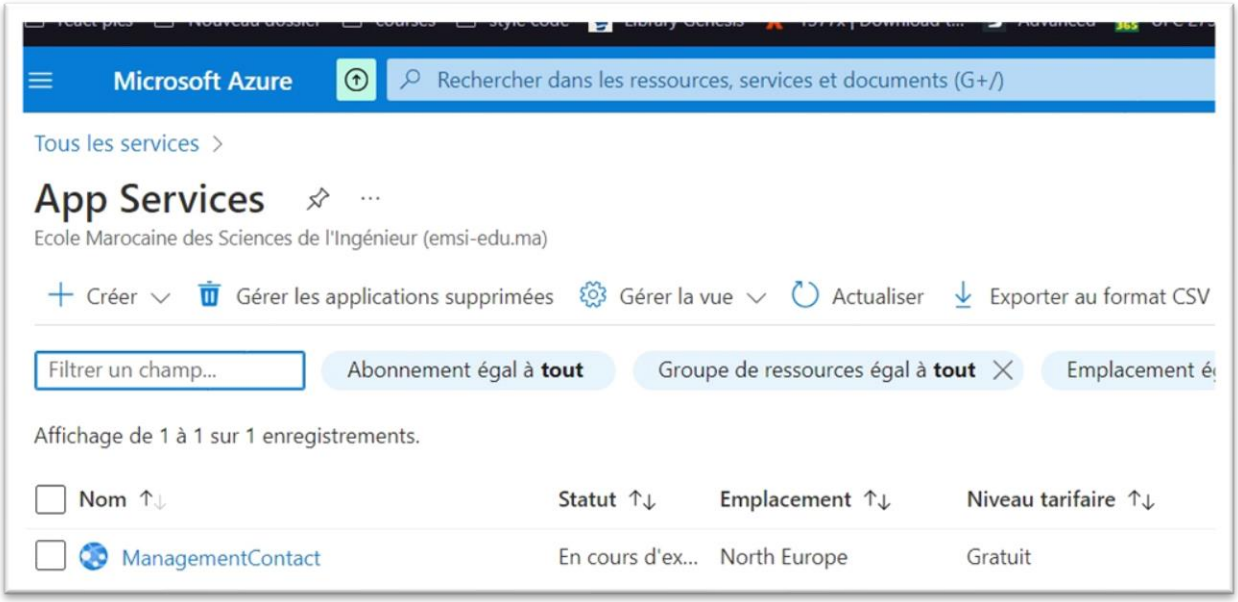
La conteneurisation avec Docker est devenue une pratique courante dans le développement logiciel moderne. Elle permet d'encapsuler une application et ses dépendances dans un conteneur, garantissant une portabilité et une cohérence entre les environnements de développement, de test et de production.

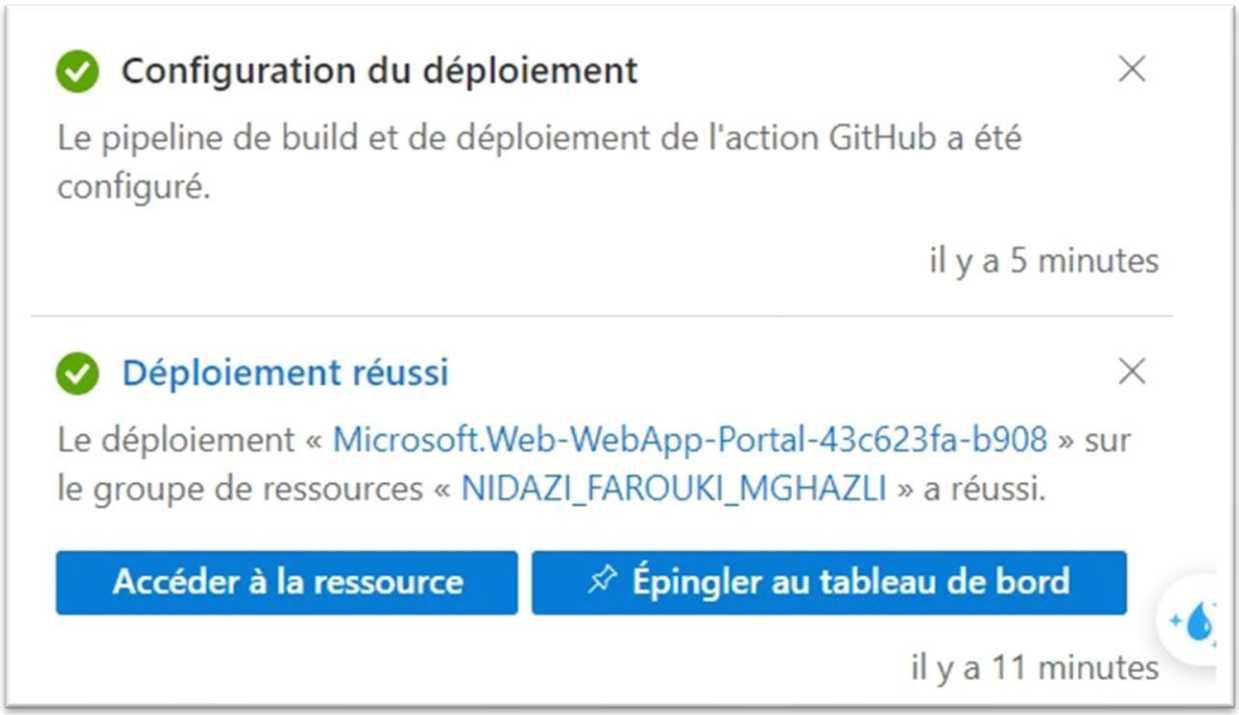


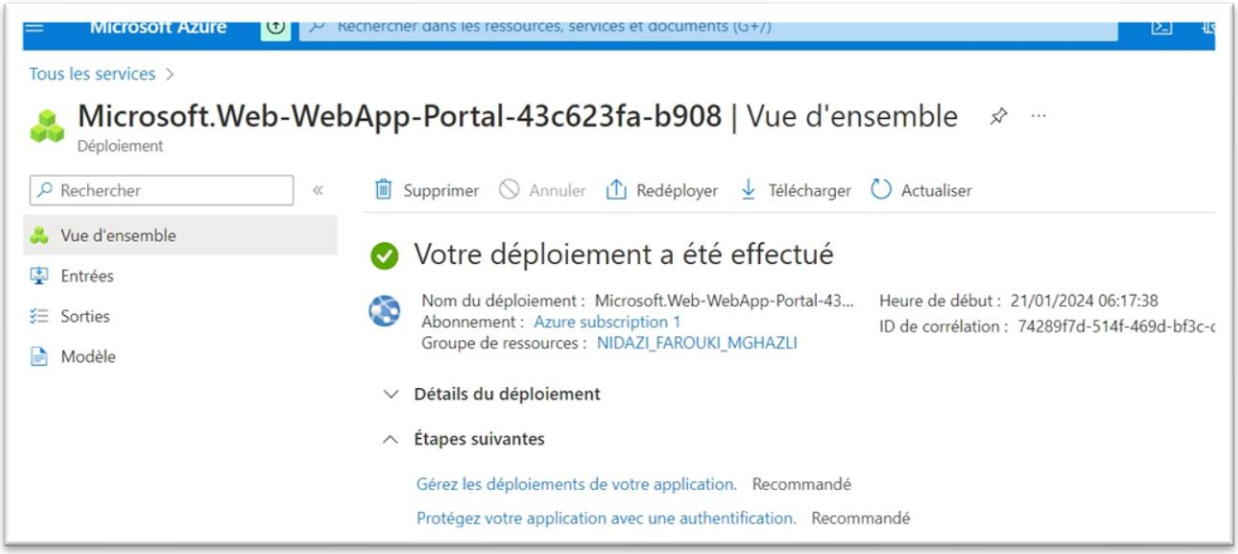
4. Déploiement Automatique : Azure Cloud

Microsoft Azure propose une gamme complète de services cloud pour le déploiement automatique. Il inclut des services tels que Azure DevOps, Azure App Service, et Azure Kubernetes Service (AKS).



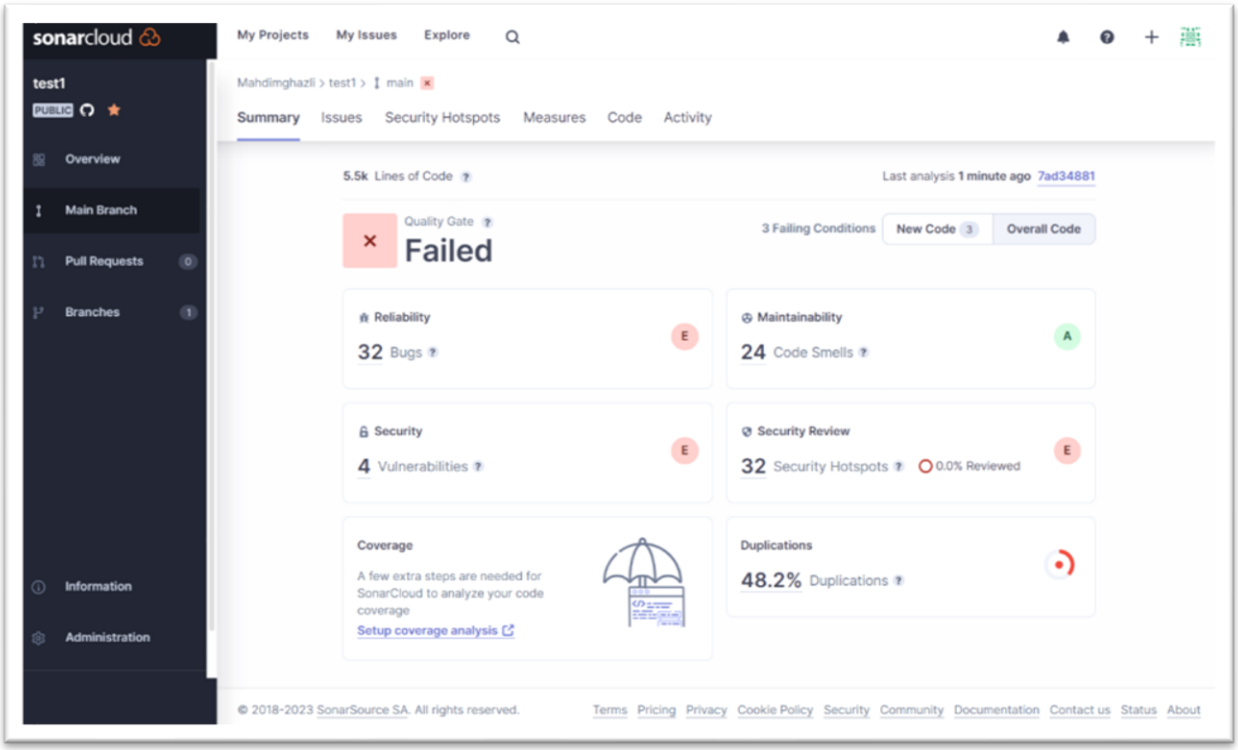

Microsoft Azure
Rechercher dans les ressources, services et documents (G+/)
Tous les services
App Services
Ecole Marocaine des Sciences de l'Ingénieur (emsi-edu.ma)
Créer
Gérer les applications supprimées
Gérer la vue
Actualiser
Exporter au format CSV
Filtrer un champ...
Abonnement égal à tout
Groupe de ressources égal à tout
Affichage de 1 à 1 sur 1 enregistrements.
Nom
Statut
Emplacement
Niveau tarifaire
ManagementContact
En cours d'ex...
North Europe
Gratuit

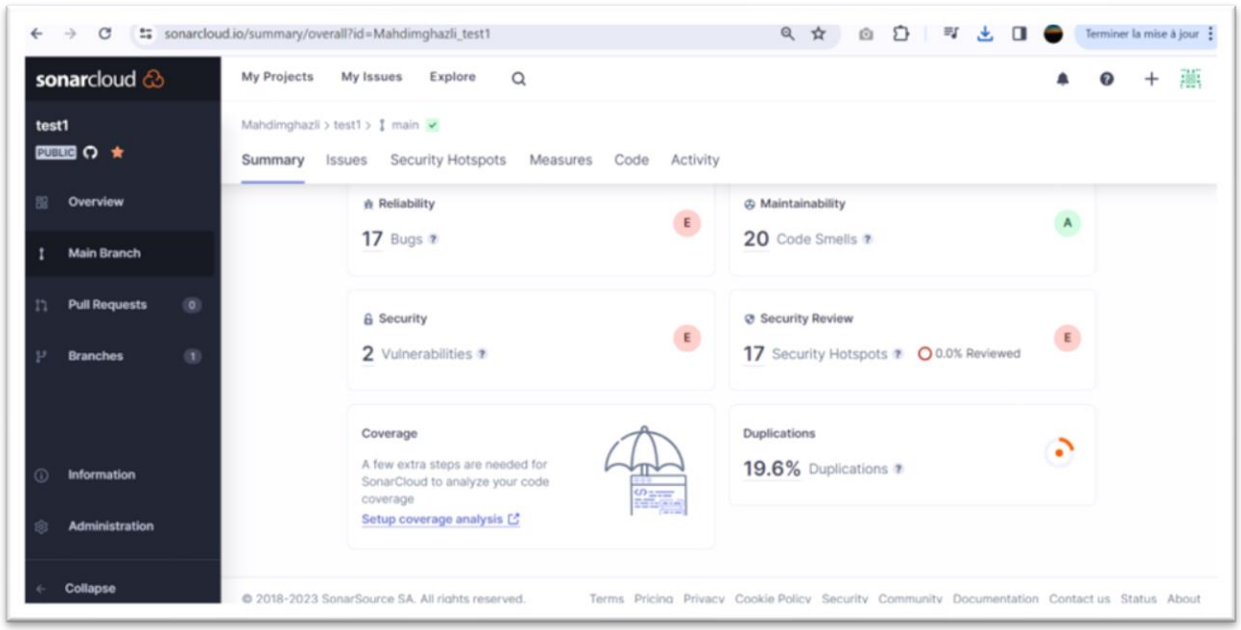

Configuration du déploiement
Le pipeline de build et de déploiement de l'action GitHub a été configuré.
il y a 5 minutes
Déploiement réussi
Le déploiement « Microsoft.Web-WebApp-Portal-43c623fa-b908 » sur le groupe de ressources « NIDAZI_FAROUKI_MGHAZLI » a réussi.
Accéder à la ressource
Épingler au tableau de bord
il y a 11 minutes

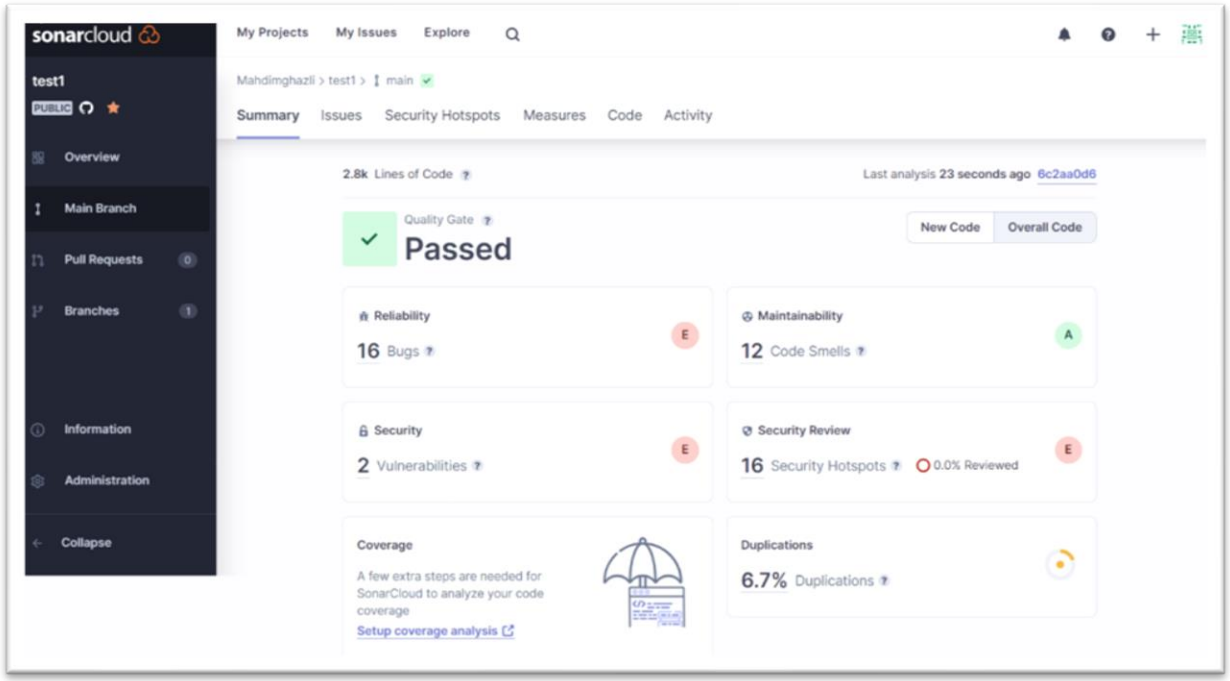


5. Intégration de SonarQube:

L'objectif principal de l'intégration de SonarQube dans notre projet est d'améliorer la qualité du code en identifiant et en corrigeant rapidement les erreurs, les vulnérabilités, et en assurant une conformité aux normes de codage. La qualité du code est cruciale pour garantir la stabilité de l'application, réduire les coûts de maintenance et améliorer la collaboration au sein de l'équipe de développement, contribuant ainsi au succès global du projet.




sonarcloud.io/summary/overall?id=Mahdimghazli_test1
Terminer la mise à jour
sonarcloud
My Projects
My Issues
Explore
test1
PUBLIC
Mahdimghazli > test1 > main
Summary
Issues
Security Hotspots
Measures
Code
Activity
Overview
Main Branch
Pull Requests
Branches
Information
Administration
Collapse
Reliability
17 Bugs
E
Maintainability
20 Code Smells
A
Security
2 Vulnerabilities
E
Security Review
17 Security Hotspots
0.0% Reviewed
E
Coverage
A few extra steps are needed for SonarCloud to analyze your code coverage
Setup coverage analysis
Duplications
19.6% Duplications
© 2018-2023 SonarSource SA. All rights reserved.
Terms
Pricing
Privacy
Cookie Policy
Security
Community
Documentation
Contact us
Status
About


sonarcloud
My Projects
My Issues
Explore
test1
PUBLIC
Mahdimghazli > test1 > main
Summary
Issues
Security Hotspots
Measures
Code
Activity
Overview
Main Branch
Pull Requests
Branches
Information
Administration
Collapse
2.8k Lines of Code
Last analysis 23 seconds ago 6c2aa0d6
Quality Gate
Passed
New Code
Overall Code
Reliability
16 Bugs
E
Maintainability
12 Code Smells
A
Security
2 Vulnerabilities
E
Security Review
16 Security Hotspots
0.0% Reviewed
E
Coverage
A few extra steps are needed for SonarCloud to analyze your code coverage
Setup coverage analysis
Duplications
6.7% Duplications

6.Conclusion

Résumé des Accomplissements :

- ❖ **SonarQube :**
- ✓ Identification proactive et correction des erreurs de code.
- ✓ Amélioration de la conformité aux normes de codage.
- ✓ Surveillance continue de la qualité du code à l'aide de métriques et de tableaux de bord.
- ❖ **Conteneurisation avec Docker :**
- ✓ Création d'environnements de développement cohérents et portables.
- ✓ Simplification du déploiement avec des conteneurs légers et autonomes.
- ✓ Amélioration de l'efficacité grâce à une gestion simplifiée des dépendances.
- ❖ **Déploiement Automatique avec Azure Cloud :**
- ✓ Automatisation complète du cycle de vie du développement.
- ✓ Utilisation d'Azure App Service pour un déploiement rapide et une gestion simplifiée de l'infrastructure.

Perspectives Futures :

- ❖ **Sécurité et Conformité :**
- ✓ Renforcer les mesures de sécurité pour garantir une protection maximale des déploiements.
- ✓ Veiller à la conformité aux normes de sécurité et de confidentialité lors des déploiements sur Azure.