

My Booking service



Révolutionnez votre système de réservation hôtelière : flexibilité, extension facile et expérience client optimale.

Équipe composée de TANSEL Atilla, KIM Theodore, MEKOULOU Keysah et JOSEPH Dhierry

Sommaire

1. **Introduction**
2. **Répartition des développements entre les membres**
3. **Schéma d'architecture My Booking service**
4. **Schéma des microservices My Booking service**
5. **Schéma d'infrastructure My Booking service**
6. **Glossaire**

Introduction

Vous trouverez dans ce document les explications/justifications de nos choix que nous avons décidé de faire pour la mise en place du projet My Booking Service. Le besoin du client est le suivant : mettre en place une API de réservation hôtelière avec possibilités d'extensions.

Qui sommes-nous ?

Nous sommes une équipe composée de quatre développeurs, TANSEL Atilla, MEKOULOU Keysah, KIM Théodore et JOSEPH Dhierry.

Nous avons décidé d'adopter la méthodologie de travail Agile SCRUM pour notre projet car cette dernière nous permet d'obtenir des résultats rapides, de favoriser la collaboration, de gérer les changements de manière flexible et d'avoir une visibilité claire sur le projet.

Ainsi, nous avons décidé de nous organiser de la façon suivante afin de répondre au besoin du client dans les meilleures conditions.

Répartition des développements entre les membres

Projet : My Booking Services - Date de début : 13 Mars 2023

Membres de l'équipe de développement :

Théodore KIM - Développeur principal

Atilla TANSEL - Développeur back-end

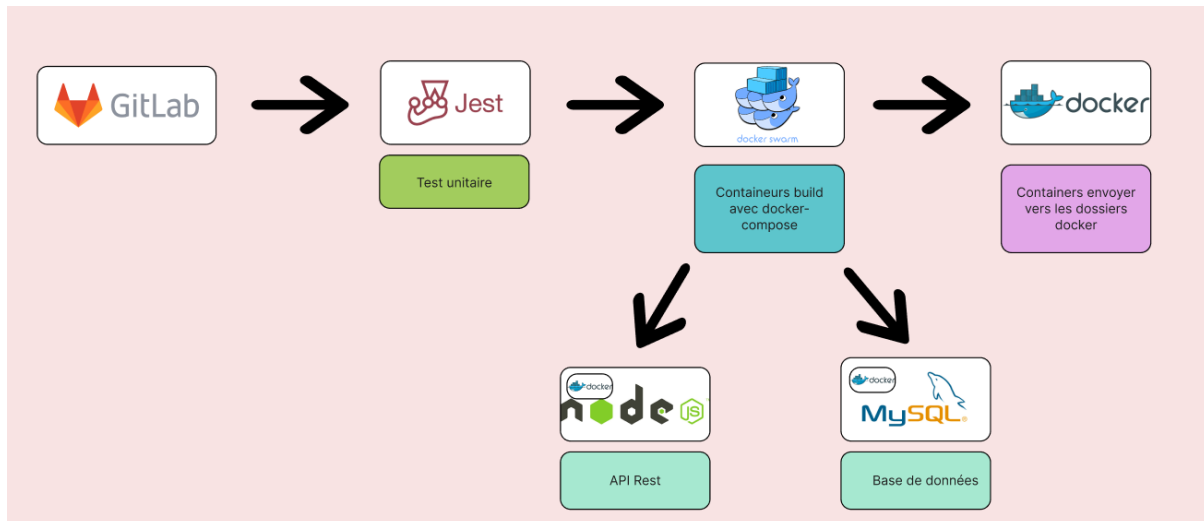
Dhierry JOSEPH - DevOPS

Keysah MEKOULOU - Scrum Master

Tâche	Responsable(s)	Échéance prévue
Mise en place des schémas d'architecture	Atilla TANSEL, Dhierry JOSEPH, Keysah MEKOULOU	5 avril 2023
Installation d'Ansible et mise en place des playbooks/rôles	Théodore KIM, Atilla TANSEL	10 avril 2023
Définir les objectifs de déploiement	Keysah MEKOULOU	15 avril 2023
Développement de la l'API	Théodore KIM, Atilla TANSEL	26 avril 2023
Mise en place de la documentation SWAGGER	Théodore KIM	5 mai 2023
Déploiement de la TICK Stack	Dhierry JOSEPH	10 mai 2023
Mise en place du système de déploiement et d'intégration continus	Théodore KIM, Atilla TANSEL, Dhierry JOSEPH	15 mai 2023
Solution de reverse proxy et de load balancing	Théodore KIM, Atilla TANSEL, Dhierry JOSEPH	21 mai 2023

Préparation pour la présentation au client	Atila TANSEL, Dhierry JOSEPH, Keysah MEKOULOU, Théodore KIM	28 mai 2023
--	---	-------------

Schéma d'architecture My Booking service



L'architecture commence par Ansible qui est utilisé pour l'intégration continue (CI), il est connecté à gitlab pour récupérer le code source à partir du référentiel GitLab. Jest est utilisé pour effectuer des tests fonctionnels pour vérifier l'API avant son déploiement. Si tous les tests réussissent, Docker Compose est utilisé pour créer des conteneurs pour l'API REST Node.js et la base de données mysql enfin Docker est utilisé pour déployer l'API et la base de données en tant que conteneur isolés.

Cette architecture suit un processus de développement et de test continu qui utilise plusieurs outils pour garantir la qualité de l'API avant son déploiement.

Pour la mise en place et la réalisation des tests unitaires à l'aide de Jest, JOSEPH Dhierry en sera chargé afin de s'assurer que l'API fonctionne correctement avant son déploiement. Cette API sera développée par TANSEL Atila et KIM Théodore qui sera leur tâche commune en NodeJS. Enfin MEKOULOU Keysah, quant à lui, sera chargé de la dockerisation du projet pour garantir sa portabilité et faciliter sa distribution une fois que l'API aura été développée et testée avec succès.

Pourquoi ces technologies ?

En utilisant Ansible, Docker et GitLab ensemble, il est possible de créer un environnement de développement et de déploiement cohérent et fiable. Ansible permet de gérer la configuration et le déploiement de l'infrastructure, Docker permet de créer des images d'application portables et GitLab permet de gérer le processus de déploiement de l'application de manière automatisée.

Docker Swarm permet de déployer et de gérer efficacement des applications conteneurisées sur un cluster de noeuds Docker, il simplifie également le déploiement des applications distribuées

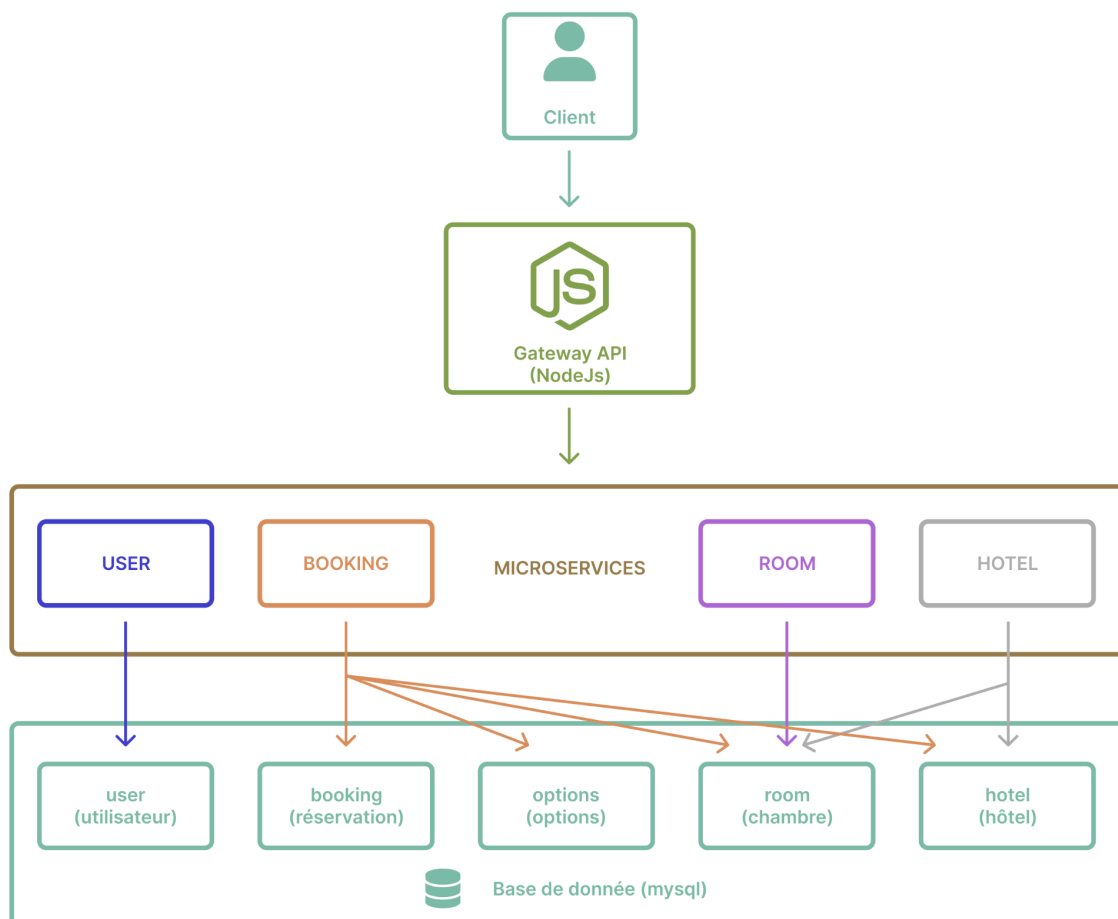
en fournissant une interface unifiée pour la gestion du cluster et s'intègre parfaitement à l'écosystème Docker ce qui nous permettra de bénéficier de toutes les fonctionnalités et des outils existants de Docker.

La performance de Node.js permet de gérer facilement les requêtes simultanées de nombreux utilisateurs lors des réservations simultanées. Les requêtes multiples peuvent être traitées de manière asynchrone, ce qui permet une latence minimale et une meilleure expérience utilisateur pour les clients effectuant des réservations.

Le choix de Jest est pour les tests unitaires API en raison de ses fonctionnalités permettant de simuler des requêtes XHR et d'intercepter des requêtes AJAX, ainsi que pour sa familiarité avec les équipes de développement qui l'utilisent déjà pour les tests end-to-end.

Pour la base de données, MySQL est très évolutif, il peut facilement gérer des volumes de données importants, est très flexible et peut être utilisé pour stocker différents types de données.

Schéma des microservices My Booking service

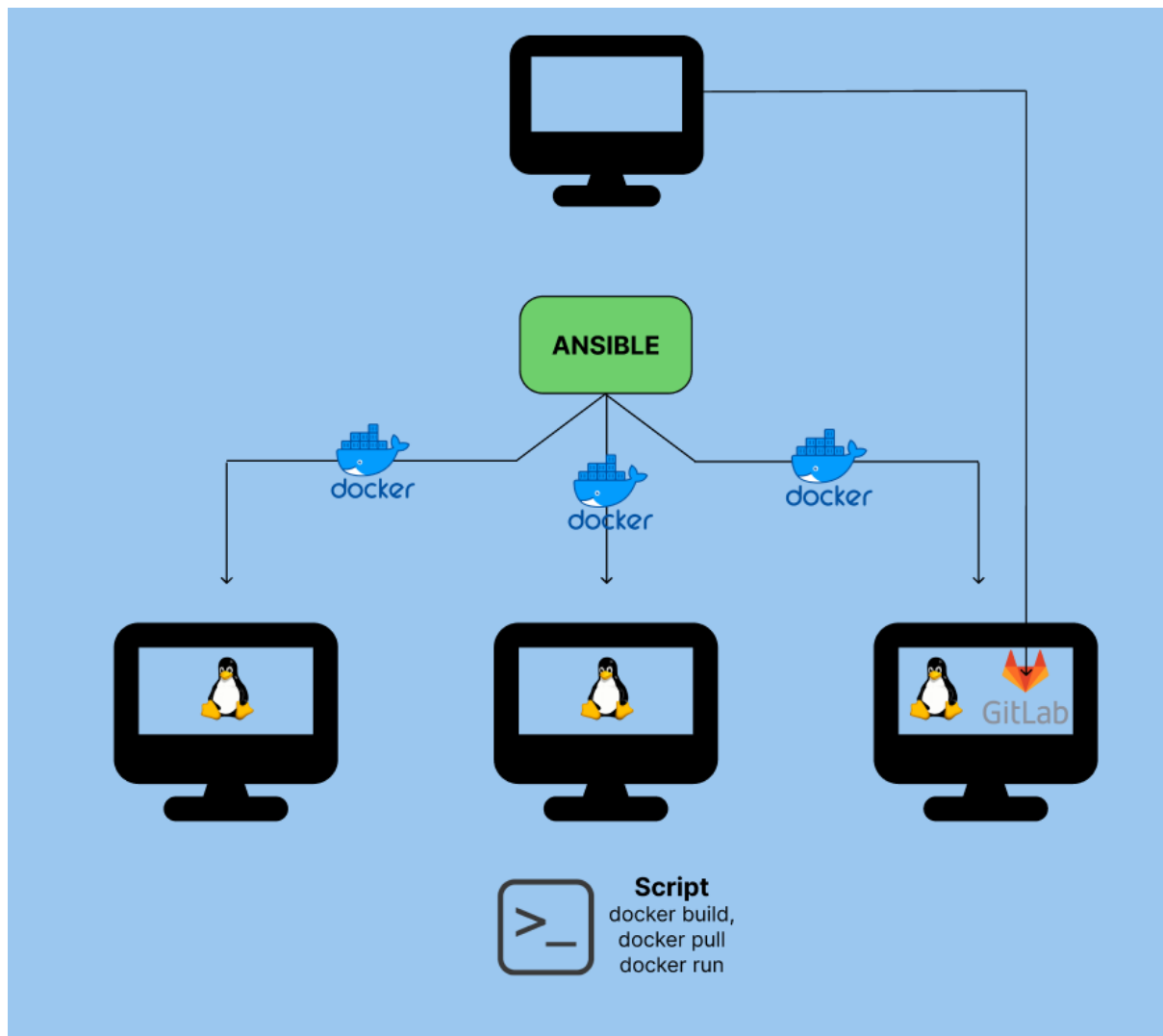


Explication du schéma :

- ❖ **Client**: C'est l'élément qui interagit avec l'ensemble du système. Il peut s'agir d'une application web, d'une application mobile ou d'une autre forme de point d'interaction utilisateur. Le client communique avec l'infrastructure via l'API Gateway.

- ❖ API Gateway en Node.js: L'API Gateway agit comme un pont entre le client et les divers microservices de l'architecture. Elle accepte les requêtes du client et les achemine vers le micro-service approprié.
- ❖ Microservices:
 - Microservice Utilisateur:
Ce service est dédié à la gestion complète des utilisateurs. Il traite toutes les informations associées aux utilisateurs, mais il permet également d'effectuer des opérations telles que la création et la suppression d'utilisateurs. Toutes les données pertinentes sont stockées dans une base de données dédiée, nommée "user".
 - Microservice Réservation:
Ce service est chargé de la gestion des réservations. Il permet de créer de nouvelles réservations, mais aussi de gérer les réservations existantes. Il s'appuie sur sa base de données "booking" pour stocker et récupérer les informations pertinentes. De plus, pour obtenir des détails supplémentaires nécessaires à la gestion des réservations, ce service a accès aux bases de données "option", "room" et "hotel".
 - Microservice Room:
Ce service est dédié à la gestion de toutes les informations et fonctionnalités relatives aux chambres. Il permet d'effectuer des opérations telles que l'ajout de nouvelles chambres, la mise à jour des informations sur les chambres existantes et la vérification de la disponibilité des chambres. Pour réaliser ces tâches, le micro-service Room utilise sa propre base de données "room", qui lui permet de stocker et de manipuler les informations pertinentes concernant les chambres.
 - Microservice Hôtel:
Ce service gère les données relatives aux hôtels. Pour cela, il a accès à sa propre base de données "hotel" et "room".

Schéma d'infrastructure My Booking service



Glossaire

Ansible: Permet de déployer, configurer et gérer des serveurs, des réseaux et d'autres ressources informatiques de manière cohérente et reproductible

API: Ensemble de règles et de protocoles qui permettent à différentes applications informatiques de communiquer et d'échanger des données entre elles

Asynchrone: Mode de fonctionnement où les opérations peuvent se dérouler indépendamment les unes des autres, sans attendre la fin de chaque opération avant de passer à la suivante

Cluster: Groupe de plusieurs ordinateurs ou serveurs interconnectés qui travaillent ensemble comme une seule entité cohérente

Conteneur: Unité d'exécution autonome qui encapsule une application et ses dépendances, offrant une isolation, une portabilité et une facilité de gestion accrues par rapport aux méthodes traditionnelles de déploiement d'applications

Docker: Plateforme open source qui permet de créer, de distribuer et d'exécuter des applications dans des conteneurs

Gitlab: Plateforme de développement logiciel complète, basée sur Git, qui offre des fonctionnalités de gestion du code source, de collaboration et de livraison continue. Elle facilite le travail collaboratif, l'intégration continue et le déploiement des projets logiciels

Intégration continue (CI): Pratique de développement logiciel qui consiste à intégrer régulièrement et automatiquement les modifications du code source dans un référentiel partagé. L'objectif principal de l'intégration continue est de détecter rapidement les éventuels problèmes d'intégration et de qualité du code

Jest: Framework de test JavaScript open source utilisé pour le développement d'applications. Il est spécialement conçu pour les tests unitaires, les tests d'intégration et les tests de snapshot

Microservices: Approche d'architecture logicielle où une application est construite à partir de services indépendants, autonomes et communiquant entre eux. Cela permet une plus grande flexibilité, évolutivité et résilience dans le développement et le déploiement des applications

Mysql: Système de gestion de base de données relationnelle open source, utilisé pour stocker, organiser et récupérer des données structurées

Requêtes AJAX: Requêtes asynchrones effectuées depuis une page web vers un serveur, permettant d'échanger des données en arrière-plan sans recharger la page entière. Elles améliorent l'interactivité et la réactivité des applications web

Requêtes XHR: Permettent d'échanger des données avec un serveur sans recharger la page entière, facilitant ainsi le chargement dynamique de contenu et l'interaction en temps réel avec le serveur

Tests unitaires: Tests automatisés qui vérifient le bon fonctionnement des unités individuelles d'un programme