



PROJET DE FIN D'ETUDES

En vue de l'obtention du diplôme

INGENIEUR D'ETAT

Filière : Génie informatique

Option : Génie Logiciel

Développement et Déploiement d'une solution E-commerce Sécurisée intégrée avec l'ERP Odoo

Réalisé par :

Mr. Hamza HADDAA

Soutenu le 24/06/2019, devant le Jury :

Pr. Fouzia MORADI	Professeur à L'ENSAH	Encadrante pédagogique
Mr. Ahmed EL HAMIDI	Ingénieur à Orange Bytes	Encadrant professionnel
Mr. Karim EL Moutaouakil	Professeur à L'ENSAH	Examinateur
Mr. Younes ABOULHANOUNE	Professeur à L'ENSAH	Examinateur

Dédicaces

En préambule à ce rapport je remercie ALLAH qui nous aide et nous donne la patience et le courage durant ces langues années d'études . Je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce travail ainsi qu'à la réussite de cette formidable année universitaire.

*Je tiens à remercier sincèrement Monsieurs, **Mustapha Hain, Aissam Jadli et Ahmed ELHAMIDI**, qui, en tant que Encadrants de projet, se sont toujours montrés à l'écoute et très disponible tout au long de la réalisation de ce projet, ainsi pour l'inspiration, l'aide et le temps qu'ils ont bien voulu me consacrer et sans qui ce projet n'aurait jamais vu le jour. Que ce travail soit l'expression de ma grande affection et un témoignage de mon attachement et de mon grand profond amour.*

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragée au cours de la réalisation de ce mémoire.

Résumé

Les magasins en ligne sont depuis des années, largement recommandés aux entreprises qui comptent sur la vente des produits, voire des services. Ce types de sites Web représentent un outil mondial permettant aux clients un accès à toutes les informations, produits et services à partir d'un portail unique en rapport avec leur activité.

Dans ce cadre, on s'est engagé dans une équipe de jeunes ingénieurs compétents pour la création du projet innovant *Joodperfumes.com*, qui peut être considéré le meilleur de son genre, au sein de l'entreprise **Orange Bytes**. Le projet consiste à la conception et à la réalisation d'une plateforme E-commerce de vente des parfums dans le monde arabe .

Au cours de cette expérience, une autre de mes missions était également la création et l'automatisation de l'infrastructure de Jood dans le cloud.

Dans ce contexte que vient s'inscrire notre **projet de fin d'études** pour l'obtention du **diplôme Ingenieur en génie Informatique à L'Ecole Nationale des Sciences Appliquées**.

Mots clefs : E-Commerce, Informatique, Programmation, Développement logiciel, Devops, Cloud

Abstract

Online stores have been for years widely recommended for companies that rely on the sale of products or services. These types of websites represent a global device that provides customers with a bridge between all information, products and services from a single portal related to their business.

In this context, we have committed ourselves to a team of young engineers competent in the innovative project *Joodperfumes.com*, which can be considered the best of its kind, within the company **Orange Bytes**. The project involves the conception and development of an e-commerce platform for sales of perfumes in the Arab world.

During this experiment, another of my missions was also the creation and automation of the Jood infrastructure in the cloud.

In this context that comes to register our **graduation project** to obtain the diploma of **engineer in Computer Engineering at the National School of Applied Sciences of Al-Hoceima**

Keywords : E-Commerce, Computer Science, Programming, Software Development, Devops, Cloud

Table des matières

Liste des figures	xiv
Liste des tableaux	xv
Introduction générale	1
1 Contexte général du projet	3
1.1 Introduction	3
1.2 Présentation de l'organisme d'accueil	3
1.2.1 Présentation d'Orange Bytes	3
1.2.2 Vision d'Orange Bytes	4
1.2.3 Services d'Orange Bytes	4
1.3 Présentation du projet	5
1.3.1 Le concept du E-Commerce	5
1.3.2 Problématique	6
1.3.3 Contexte du projet	7
1.4 Conduite et gestion du projet	7
1.4.1 Contraintes du projet	7
1.4.2 Communication entre les membres d'équipe avec l'outil Slack	8
1.4.3 Gestion des suivis et tâches avec la plateforme JIRA	9
1.4.4 Conduite du Projet	11
1.4.5 Planification du Projet	11
1.5 Conclusion	12
2 Etude fonctionnelle - technique	13
2.1 Introduction	13
2.2 Spécification des besoins	13
2.3 Analyse fonctionnelle	14
2.3.1 Identification des acteurs	14
2.3.2 Identification des différents modules (applications)	14
2.3.3 Diagrammes de cas d'utilisation	15
2.3.4 Diagrammes de séquence	17
2.3.5 Diagrammes de classes	20
2.4 Etude technique du projet	21
2.4.1 Modèles de conception adoptés	21
2.4.2 Architecture Global - Applications	22

2.5	Conclusion	23
3	Technologies Utilisées	24
3.1	Introduction	24
3.2	Outils d'infrastructure Cloud	24
3.3	Outils de développement -Backend	25
3.4	Outils de Tests	30
3.5	Outils de development Front-end	31
4	L'infrastructure cloud de joodperfumes	35
4.1	Introduction	35
4.2	Qu'est-ce que le cloud computing	35
4.2.1	Principaux avantages du cloud computing	35
4.2.2	Types de cloud computing	36
4.2.3	Types de services cloud : IaaS, PaaS, SaaS	36
4.3	le choix de AWS	37
4.4	Infrastructure en tant que code	38
4.5	Terraform - Concepts	38
4.5.1	Terraform - Fournisseur (Provider)	39
4.5.2	Terraform - Variables	40
4.5.3	Terraform - Modules	40
4.5.4	Terraform - Outputs	41
4.5.5	Terraform - Sources de données (Data Sources)	41
4.5.6	Terraform - Etat (State)	42
4.5.7	Terraform - CLI	43
4.6	Infrastructure de Jood	43
4.6.1	Jood-Services de calcul et de mise en réseau	43
4.6.2	Jood-Services de stockage et de livraison de contenu	45
4.6.3	Jood-Services de sécurité et d'identité	46
4.6.4	Jood-Services de Management	47
4.6.5	Jood-Outils de développement	47
4.6.6	La sécurisation de l'infrastructiure de Jood	48
4.6.7	L'architecture global	50
4.7	Conclusion	52
5	Jood - développement backend	53
5.1	Architecture backend du Jood	53
5.1.1	Répertoire Pytest_cache	54
5.1.2	Répertoire Project	54
5.1.3	Répertoire Templates	55
5.1.4	Répertoire apps	55
5.1.5	Répertoire Tests	56
5.1.6	Les fichiers Docker	56
5.1.7	Autre Fichiers	57
5.2	Interaction avec l'ERP Odoo	57
5.3	L'API de Jood	58
5.4	Conclusion	61
6	Jood - développement Front-end	62
6.1	Introduction	62
6.2	Organisation du projet	63
6.3	Workflow de JoodPerfumes	64
6.3.1	Interfaces d'utilisateur principale	64

6.3.2	Autres pages et composants	70
6.4	Conclusion	72
Conclusion		73
Liste des abréviations		75
Webographie		76

Table des figures

1.1	L'organigramme de la société d'accueil	4
1.2	les coordonnées de l'entreprise	5
1.3	Aperçu de Slack	8
1.4	Les messages directs	9
1.5	Workflow dans JIRA	10
1.6	Sprint Backlog <i>Joodperfumes.com</i> dans JIRA	10
1.7	Méthode agile Scrum	11
1.8	Diagramme de Gantt	12
2.1	Diagramme de cas d'utilisation d'un Client	16
2.2	Diagramme de cas d'utilisation de l'admin du plateforme	17
2.3	Diagramme de séquences : cas d'utilisation d'authentification	18
2.4	Diagramme de séquences : cas d'utilisation d'achat sur la plateforme	18
2.5	Diagramme de séquences : cas d'utilisation de paiement via KNET	19
2.6	Diagramme de classes du site	21
2.7	Architecture MVC	22
2.8	Design Pattern Active Record	22
2.9	Exemple d'architecture microservices	23
3.1	Pipeline d'exécution de Terraform	25
3.2	Aperçu de jood sur GitHub	27
3.3	Appel d'une Tache celery	28
3.4	Les principales fonctionnalités de ERP Odoo	29
4.1	Tendances du Cloud Computing sur Stack Overflow	38
4.2	HashiCorp Configuration Language	39
4.3	Creation de S3 bucket avec terraform	39
4.4	Résultat sur AWS	39
4.5	Exemples de fournisseurs Terraform	40
4.6	Variables avec Terraform	40
4.7	Modules avec Terraform	41
4.8	Outputs avec Terraform	41
4.9	Appel d'une Output avec Terraform	41
4.10	Data sources avec Terraform	42
4.11	Appel d'une source de donnees avec Terraform	42

4.12 état (state) avec Terraform	42
4.13 le stockage d'un etat avec Terraform	43
4.14 la gestion de conteneurs avec AWS Fargate	45
4.15 Security groupe du service Jood-core	46
4.16 Security groupe du Load Balancer	47
4.17 Processus de publication avec CodePipelin	47
4.18 Intégration-Déploiement Continue de Jood	48
4.19 Sessions SSH établies par l'intermédiaire d'un hôte Bastion	49
4.20 La configuration des hôtes VPN et Bastion dans jood	50
4.21 L'architecture cloud du Jood sur AWS	51
 5.1 Arhitecture du Jood-Core	53
5.2 Le Répertoire du Pytest_cahe	54
5.3 le répertoire d'arborescence du projet	54
5.4 Le répertoire de templates	55
5.5 Le répertoire des modules de projet	56
5.6 Le répertoire des tests	56
5.7 Le fichiers docker de projet	57
5.8 Connexion Jood-Core au serveur Jood-Odoo avec XRPC	57
5.9 Obtenir les modèles de Odoo	57
5.10 Creation d'un ulisateur sur Odoo	58
5.11 création d'un utilisateur sous Odoo avec celery	58
5.12 La création d'un ordre pour un utilisateur dans Odoo	58
5.13 Exemple d'un ProductType avec graphene-django	59
5.14 Exemple de ProductQuery avec graphene-django	59
5.15 Déclaration des requêtes et des mutations dans le schéma du Joodperfumes	60
5.16 L'ajout de graphiql au urlpattern de projet	60
5.17 La Vue de Graphiql api Jood	60
5.18 La requête products dans graphiql	61
5.19 Le résultat de la requête products dans graphiql	61
 6.1 Le référentiel du front-end sur github	62
6.2 La structure du Jood-Web	63
6.3 L'interface principale du Joodperfumes	65
6.4 Le choix d'un papier cadeau (Gift Wrapping)	66
6.5 L'icon de panier du plateforme Jood	66
6.6 Le contenu d'un panier dans Jood	67
6.7 Le choix de la méthode de paiment	68
6.8 Ajout d'une nouvelle adresse	69
6.9 Menu de gestion (d'adresses, mot de passe, profile)	69
6.10 Interface de gestion d'adresses	70
6.11 détails d'un produit	71
6.12 Modification des données d'un utilisateur	71
6.13 Interface pour changer un mot de passe	72

Liste des tableaux

2.1	Scénarios des cas d'utilisation d'un Client	15
2.2	Scénarios des cas d'utilisation de l'admin du plateforme	17
2.3	Identification des Classes	20
4.1	Les réseaux de l'infrastructure Jood	44
4.2	Les Instances EC2 de l'infrastructure Jood	44
4.3	Les Conteneurs Docker de l'infrastructure Jood	44
4.4	Jood-Elb	45
4.5	Les base de donnees RDS du Jood	46

Introduction générale

Ces dernières années ont été caractérisées par une évolution significative du commerce électronique. En effet, les entreprises ont commencé à faire d'énormes investissements dans la création de sites commerciaux, ces plateformes permettant aux clients de profiter d'une foire virtuelle disponible. Les produits sont mis à jour quotidiennement sans problèmes de distance, d'heures de travail et de disponibilité des transports.

Le marketing numérique est une discipline qui nécessite toujours l'écoute de son environnement. Avec le développement des moyens numériques et en particulier d'Internet, le marketing a été un grand changement. Aujourd'hui, Internet est un canal de vente, de communication et d'échange d'informations.

Une plate-forme de commerce électronique doit garantir aux clients la simplicité du processus d'achat, une gestion simple et efficace pour les administrateurs, permettant ainsi un bon positionnement par rapport aux concurrents.

Le projet **Joodperfumes**, a pour objectif de mettre en place une plateforme e-commerce de vente des parfums dans le monde arabe et spécialement dans les trois pays Koweit, Arabie saoudite et Émirats arabes unis avec un magasin qui se base au Koweit.

JoodPerfumes offre une zone d'administration complète offrant une gestion des clients, des factures, commandes et la gestion d'un système de questions / réponses.

La création d'une telle plate-forme fait l'objet de notre projet de fin d'études . Ce projet a été réalisé au sein de la société **Orange Bytes**. Cette dernière est une start-up koweïtienne innovante à fort potentiel de croissance et possède une filiale au Maroc, Casablanca . spécialisée dans les technologies de l'information et de la communication. Ce travail sera effectué en collaboration avec l'équipe de front-end, afin de garantir toutes les fonctionnalités d'un système de commerce électronique.

Dans un second temps, et voyant que je poursuis mes études dans le domaine de **BIG DATA**, mes autres missions étaient la création, l'administration et le déploiement de l'infrastructure de la plateforme dans le cloud. Ce fut une excellente occasion pour moi d'appliquer mes compétences acquises dans le cadre de ma carrière et de mon expérience pratique dans ce domaine.

Ce rapport est le fruit et l'illustration du travail effectué dans ce stage, décrit les différentes phases et les moyens que nous avons utilisés pour atteindre les objectifs en six chapitres.

Le premier chapitre envisage la description du contexte général du projet qui permettra de situer ce dernier et de bien comprendre la problématique tout en mettant en valeur les avantages du E-Commerce par rapport à la méthode traditionnelle.

Introduction générale

Dans **Le deuxième chapitre** on va aborder l'étude fonctionnelle du projet, qui vous donnera un aperçu sur les principales fonctionnalités de notre système. ainsi que l'étude technique qui décrit ce dernier en termes d'architecture.

Dans **Le troisième chapitre** qui présente les principales technologies et outils que nous utilisons avec Joodperfumes.

Puis on va passer au **quatrième chapitre** qui s'agit de l'architecture cloud du projet qui décrit ce dernier en termes d'infrastructure.

Nous décrivons dans **le cinquième chapitre** l'architecture et la mise en œuvre de la partie back-end du projet.

Enfin, **Le dernier chapitre** concerne la partie front-end qui décrit l'interaction avec les différentes interfaces des plates-formes.

Contexte général du projet

1.1 Introduction

Ce chapitre présente le contexte général du projet. Nous allons commencer tout d'abord par une présentation générale de l'organisme d'accueil. Nous aborderons par la suite la présentation du projet où nous décrirons les objectifs ciblés ainsi que le planning de travail adopté pour ce projet.

1.2 Présentation de l'organisme d'accueil

1.2.1 Présentation d'Orange Bytes

Orange Bytes est une start-up kuwaitienne innovante à fort potentiel de croissance qui a été fondée en 2010 pour se positionner sur le secteur de la technologie, et octroyer à ses utilisateurs des solutions de qualité et surtout adaptées à leurs besoins spécifiques, son siège social est situé à Dubai dans l'U.A.E et elle a une filiale au Maroc qui a été créée en 2014.

Parmi les avantages du travail avec une start-up telle qu'Orange Bytes, on site :

1. **L'agilité** : Les start-up sont des entreprises jeunes et souples. Elles peuvent avoir un business plan et une stratégie opérationnelle bien installés, mais rien ne les empêche d'en changer en cours de route pour s'adapter à un changement de marché.
2. **Bonnes perspectives d'évolution** : Lorsque une personne commence à travailler dans une start-up dès sa création, elle pourra voir la progression de l'entreprise et elle aura aussi la chance de grandir avec elle et de monter en grade plus rapidement.
3. **Un cadre de travail sympa et cool** : l'atmosphère d'une start-up est généralement très informelle et peu hiérarchique, l'employé sentira comme dans une famille. Aller voir un match de foot avec son patron sera considéré comme quelque chose de normal.
4. **La variété des tâches** : Les équipes sont de petites tailles et les projets nécessitent souvent plusieurs compétences. Alors les développeurs sont amenés à travailler avec d'autres métiers. C'est enrichissant, vu qu'il permet de s'ouvrir sur les diverses activités de l'entreprise et de découvrir ce que font les autres en tant que développeurs.
5. **L'environnement de travail** : L'environnement de travail est très dynamique. L'équipe est la plupart du temps jeune et pleine de ressources.

La figure ci-dessous montre l'organigramme de la société d'accueil :

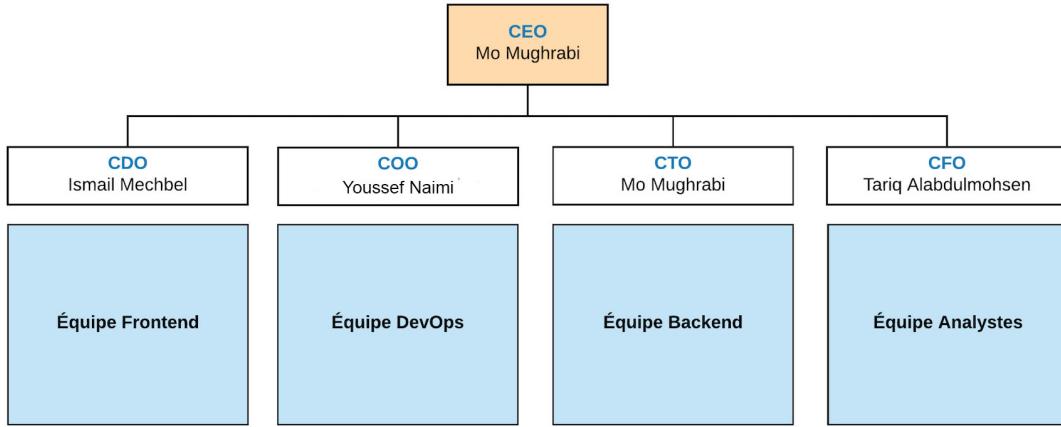


FIGURE 1.1: L'organigramme de la société d'accueil

1.2.2 Vision d'Orange Bytes

Les fondateurs avaient la volonté d'exploiter la technologie afin de fournir aux gens des solutions qui leur permettent d'exprimer leurs passions. Pour se faire, une attention particulière était procurée aux nouvelles technologies, aux processus d'automatisation, à l'innovation et à la recherche et développement.

1.2.3 Services d'Orange Bytes

De nos jours, le progrès technologique permet désormais aux entreprises de se concentrer plus sur l'innovation et la valeur ajoutée, et de démocratiser l'accès aux solutions développées.

Puisque l'univers digital est un espace où chacun peut contribuer et apporter, les outils et les technologies à disposition des masses (Emails, plateformes e-learning, plateformes de communication etc.) sont source de distraction, ne présentent plus un cadre de travail optimale, et ne suivent guère l'évolution économique et les nouveaux processus imposés sur les organisations.

Orange Bytes propose donc des solutions sous forme d'un cadre de travail global tenant compte de toutes les spécificités organisationnelles et opérationnelles, des solutions qui s'adaptent au changement et qui tiennent compte des besoins des clients en temps réel.

Cette façon de faire, impose à la société d'opérer dans un climat d'extrême incertitude, puisque les solutions qu'elle offre sont nouvelles, ce qui fait que leur adoption est incertaine. Parmi les solutions :

a- La stratégie de produit : Tous les produits excellents commencent par une stratégie claire axée sur le client et le marché. Alors l'équipe d'Orange Bytes veille à ce qu'elle travaille sur ce qui compte. Certaines de ses forces comprennent :

- Étude de marché.
- Analyse compétitive.
- Audit des analyses existantes.
- Exigences du projet.
- Recommandations d'ingénierie.
- Planification de la feuille de route.

b- UX & UI Design : Grâce à une stratégie axée sur l'utilisateur, ils comprennent les besoins réels de ses clients afin d'élaborer des solutions innovantes, des sites Web et des applications mobiles qui créent une expérience d'utilisateur intuitive et agréable. Certaines de ses forces comprennent :

- Architecture de l'information.
- Prototype d'application.
- Test d'utilisabilité.
- Guide de style visuel.

c- Développement Web & Mobile : Avec son efficacité et sa compétence de travailler avec un large éventail de technologies qui répondent le mieux aux besoins spécifiques des clients et leurs exigences. Certaines de ses forces comprennent :

- HTML5 & CSS 3
- Vanilla JavaScript, ReactJS & AngularJS
- Python, Django
- SQL (MySQL, PostgreSQL) & NoSQL (Elasticsearch, Redis, DynamoDB, Cassandra)

d- Cloud computing : Orange Bytes est en mission pour aider à tirer le maximum d'avantages du cloud. Elle offre la stratégie, la migration et l'automatisation du cloud. Certaines de ses forces comprennent :

- Amazon AWS, Google Cloud Plateforme et d'autres services cloud.
- Nginx, Apache & Kong API Gateway.
- Orchestration d'infrastructure (Kubernetes et Saltstack).
- Outil de gestion de la configuration (Ansible)
- Configuration et création de l'infrastructure grâce au code (Terraform).
- Découverte du service et configuration en direct (Consul).
- Conteneurs Docker pour construire une Architecture Microservices.

Siège social	Kuwait City	
Année de fondation	2013	
Type de compagnie	Privée	
Taille de l'entreprise	40 employés	
Site Web	http://www.abytes.com	

FIGURE 1.2: les coordonnées de l'entreprise

1.3 Présentation du projet

1.3.1 Le concept du E-Commerce

1.3.1.1 Généralités sur le commerce électronique

Selon Wikipedia, le e-commerce, ou commerce électronique, est « l'échange pécuniaire de biens, de services et d'informations par l'intermédiaire des réseaux informatiques, notamment Internet. »

En d'autres termes, il s'agit d'un commerce qui gère les paiements grâce à des moyens électroniques. Chaque commerce se dirige à un type de client spécifique. Cela nous permet de distinguer les types d'e-commerce suivants :

B2B : Entreprises dont les clients finaux sont d'autres entreprises ou d'autres organisations. Par exemple, une boutique de matériel de construction qui s'adresse à des architectes ou des décorateurs d'intérieurs.

B2C : Entreprises qui vendent directement les produits ou services aux consommateurs finaux. C'est la forme la plus courante, et l'on dénombre une multitude d'exemples dans les domaines de la mode, de l'électronique, etc.

C2B : Portails sur lesquels les consommateurs proposent un produit ou un service que les entreprises peuvent acquérir. Il s'agit des portails d'emploi freelance classiques comme Elance, Odesk, Humaniance, Hopwork, etc.

C2C : Entreprises qui facilitent la vente de produits entre consommateurs. Les exemples les plus parlants sont eBay, Leboncoin ou tout portail de vente d'articles d'occasion entre particuliers. Ces formes de commerce électronique sont les plus courantes. Il en existe également d'autres types, comme le G2C (Government-to-Consumer), le C2G (Consumer-to-Government) ou le B2E (Business-to-Employer).

Comme l'on peut le voir, le commerce électronique regroupe des réalités qui vont au delà d'une simple achat dans une boutique.

1.3.1.2 Avantages et les limites d'un site e-commerce

Pourquoi les e-commerces sont-ils devenus si nombreux sur Internet en si peu de temps ?

Premièrement, car le e-commerce représente d'importants avantages par rapport au commerce traditionnel.

Avantages :

- Il facilite les achats, sans que l'acheteur n'ai besoin de se déplacer.
- Il permet de recueillir des données sur les clients, ce qui permettra de mettre en place des actions de marketing direct.
- Ce type de site permet aux clients de rechercher le meilleur prix.
- Il permet de développer l'interactivité entre l'entreprise et ses clients (sons, images, vidéos...)
- Pas d'horaires à l'inverse des boutiques traditionnelles ...

Inconvénients :

- Le manque de confiance dans les moyens de paiement et la peur de l'arnaque, portent préjudice aux sites de e-commerce.
- Le non contact avec le produit (et plus particulièrement en ce qui concerne le milieu du vêtement) peut être un frein à l'achat.
- pas de suivi par service après-vente en cas de problème

1.3.2 Problématique

Alfoudari Inc, spécialisée dans les parfums et les produits cosmétiques, est établie sur le marché koweïtien depuis plus de 30 ans. Elle possède une expérience dans le mélange, la personnalisation et la distribution de parfums qui séduit un large segment du marché koweïtien. La marque opère aujourd'hui avec 3 magasins physiques distribués au Koweït, une présence sur les médias sociaux pour le marketing et des commandes passées / reçues du public.

La marque est confrontée à une population de jeunes croissante qui vit à un rythme rapide, les magasins physiques ont considérablement chuté, ce qui donne lieu à une augmentation des services en

ligne dans le pays. De la restauration et la livraison de nourriture, épicerie, électronique et bien plus encore.

Solution proposée : Mettre en œuvre et expérimentation la digitalisation d'Alfoudari Inc, tirez parti du nom commercial et de la marque existants pour les commercialiser à un plus grand nombre d'utilisateurs. Établir des canaux de marketing en ligne, des procédures, un service client, des politiques de livraison et des canaux de marketing.

La phase initiale se concentrera sur la mise en place de l'activité de vente en ligne, des méthodes de travail, de la constitution de l'équipe chargée de la création de cette initiative et de la croissance et de la résilience des solutions introduites.

1.3.3 Contexte du projet

Le projet consiste au développement backend et de l'implémentation d'une infrastructure Cloud adaptée pour l'architecture spéciale de la plateforme e-commerce Joodperfumes qui est une plateforme d'achat en ligne qui vise à offrir des produits avec des normes de haute qualité et d'une manière plaisante et efficace, permettant aux clients de profiter d'une foire virtuelle disponible et sont mis à jour quotidiennement sans aucune contrainte, ce qui leur permettra de ne jamais manquer les nouveautés, comme une foire sans problèmes de distance géographique, d'horaire de travail et de disponibilité de transport. La plateforme offrira une grande variété de produits, exclusivement dans les trois pays Koweit, Arabie saoudite et Émirats arabes unis La plateforme *Joodperfumes.com* serait l'endroit d'achat en ligne avec une sécurité totale de tous les modes de paiements disponibles sur notre boutique

1.4 Conduite et gestion du projet

La gestion de projet est une démarche visant à organiser de bout en bout le bon déroulement d'un projet. Lorsque la gestion de projet porte sur un ensemble de projets concourants à un même objectif, on parle de gestion de programme.

En pratique, pour que le projet tourne vers l'objectif final, il doit être adaptable à des modifications fréquentes, mais maîtrisées et planifiées. Donc toute modification doit rester contrôlée. Et notamment, le projet doit rester dynamique et équilibré continuellement avec les contraintes techniques, de coût et de délai.

1.4.1 Contraintes du projet

Cette section recense les contraintes ou les restrictions particulières qui limitent le projet ou qui lui imposent des conditions, en particulier celles qui sont liées à la portée.

Les contraintes peuvent provenir de facteurs externes (sociaux, environnementaux, politiques, économiques et technologiques) ou de facteurs internes (ressources, expertise, exigences opérationnelles, exigences juridiques, installations, etc...). Il faut donc analyser l'environnement du projet afin de recenser les contraintes.

Deux catégories de contraintes ont été recensées :

1. Contraintes générales :
 - Respecter la date de livraison du projet.
 - Respecter les exigences du client d'une façon stricte.
2. Contraintes techniques :
 - Respecter les limitations de GraphQL API.
 - Protéger les données de l'utilisateur.
 - Offrir une plateforme e-commerce hautement disponible.

- Tous les composants critiques de l'infrastructure doivent être surveillés à l'aide des outils de « Monitoring ».
- La maintenance proactive de l'infrastructure.
- Respecter les exigences de sécurité d'une application web et de son infrastructure (Contrôle d'accès, Authentification et gestion de session, Chiffrement, Journalisation, VPN ...)
- Lire et analyser attentivement le code source d'une bibliothèque open source avant de l'utiliser.
- Rendre l'architecture du site extensible pour toute maintenance et évolution ultérieure.
- Toujours faire des tests unitaires d'un code avant de le déployer au production.
- Offrir une meilleure expérience aux utilisateurs finaux (temps de réponse logique, une navigation fluide ...)
- L'infrastructure doit être scalable pour supporter la monte en charge du trafic (« scalabilité »).

1.4.2 Communication entre les membres d'équipe avec l'outil Slack

Slack est un espace de travail numérique qui en interconnectant toutes Les ressources (humaines ou techniques), permet de faire des gains de productivité.

Pour comprendre le fonctionnement de Slack, il est utile de savoir comment les éléments s'imbriquent les uns dans les autres. La figure 1.3 illustre un exemple d'espace de travail Slack.

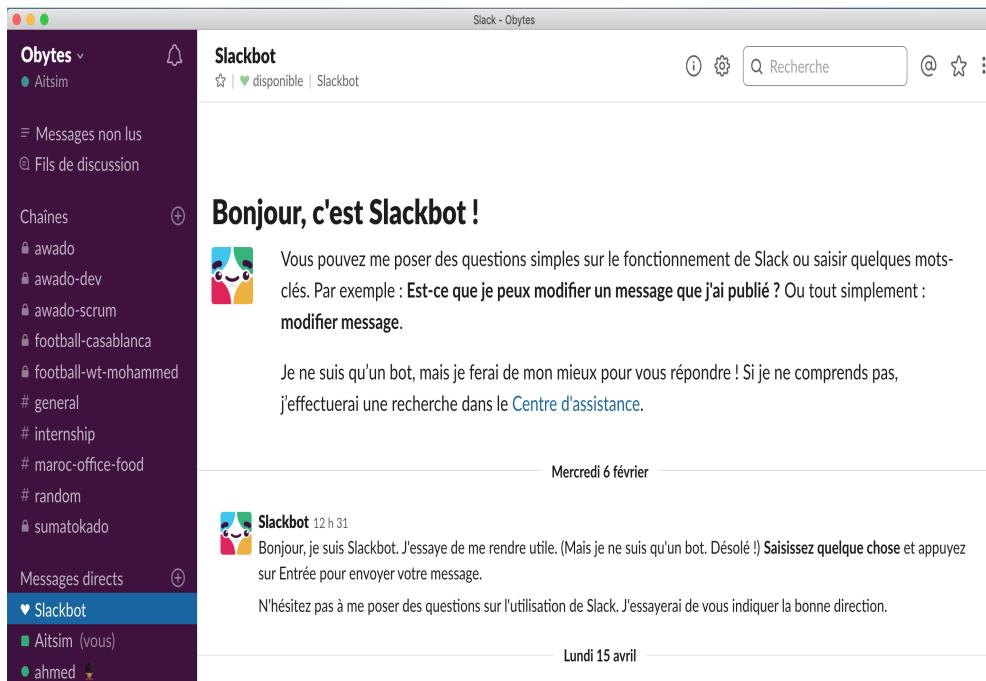


FIGURE 1.3: Aperçu de Slack

Bénéfices de Slack au sein de notre équipe

- Établir et structurer une liste de choses à faire (TODOs) Dans l'environnement Slack, on peut organiser nos tâches de nombreuses façons différentes, notamment en utilisant des posts, des épingle
- Organiser des réunions Qu'elles se déroulent en présentiel, au téléphone ou sur une chaîne, les réunions sont faciles à organiser avec Slack. Grâce aux appels audio et vidéo de Slack

- Les membres de l'équipe travaillent plus efficacement étant connectés entre eux. Il n'est pas toujours facile pour l'équipe de continuer à collaborer avec des personnes qui se trouvent dans un autre pays, dans une autre ville ou même dans un autre bureau ! Les chaînes et les messages directs de Slack nous a permis de communiquer avec les membres de l'équipe, où qu'ils se trouvent dans le monde ou dans l'entreprise. Comme parfois les idées ne peuvent prendre vie que lorsqu'elles sont échangées de vive voix, nous pouvions appeler les collaborateurs directement depuis Slack pour partager nos idées et points de vue.
- Réduire le bruit de fond. En intégrant Slack dans notre flux de travail, nous pouvions passer moins de temps à jongler entre les différents outils que nous avons l'habitude d'utiliser. Moins de distractions au cours de notre journée de travail, c'est plus d'attention et d'investissement pour les tâches importantes

Les messages directs sont parfaits pour de brèves conversations qui ne nécessitent pas la participation de tous les membres d'une chaîne.(figure 1.4),

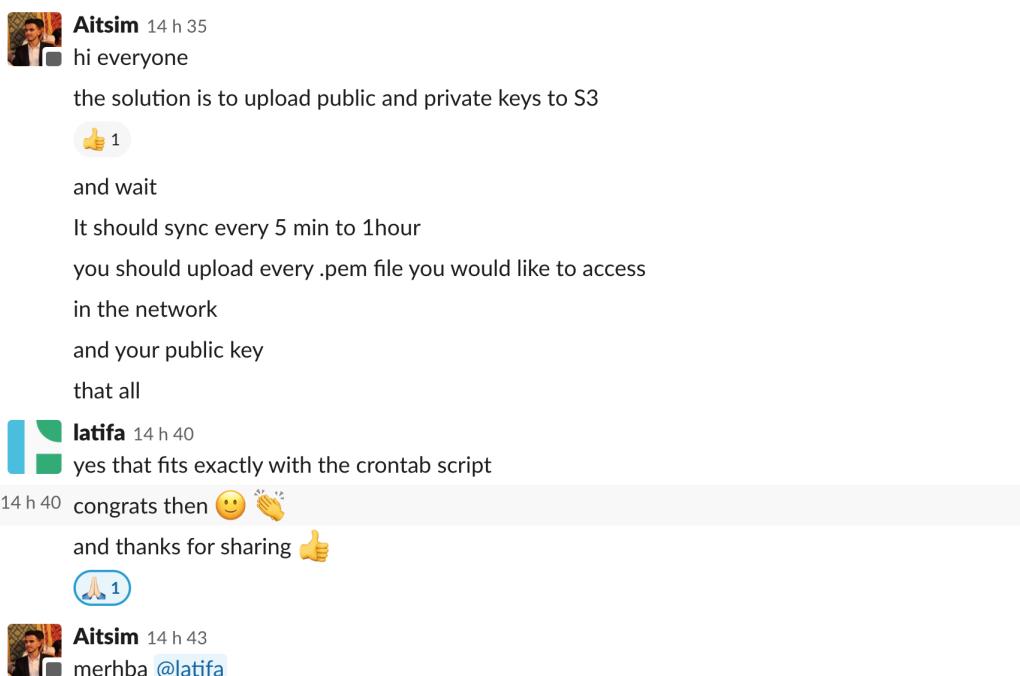


FIGURE 1.4: Les messages directs

1.4.3 Gestion des suivis et tâches avec la plateforme JIRA

JIRA est un système de suivi de bugs, un système de gestion des incidents, et un système de gestion de projets développé par Atlassian Software Systems.

Grâce aux fonctionnalités collaboratives de JIRA, les équipes travaillent en synergie focalisées sur leurs objectifs, sans perdre de temps ni d'énergie. De plus l'outil est paramétrable et permet d'obtenir une forme propre du projet.

Cet outil permet aussi de gérer le cycle de développement de projet de la conception jusqu'à la production et puis les incidents pendant l'exploitation du produit.

La figure 1.5 représente le Workflow dans JIRA, qui est un ensemble d'états et de transissions par lesquelles passe un issue durant son cycle de vie.

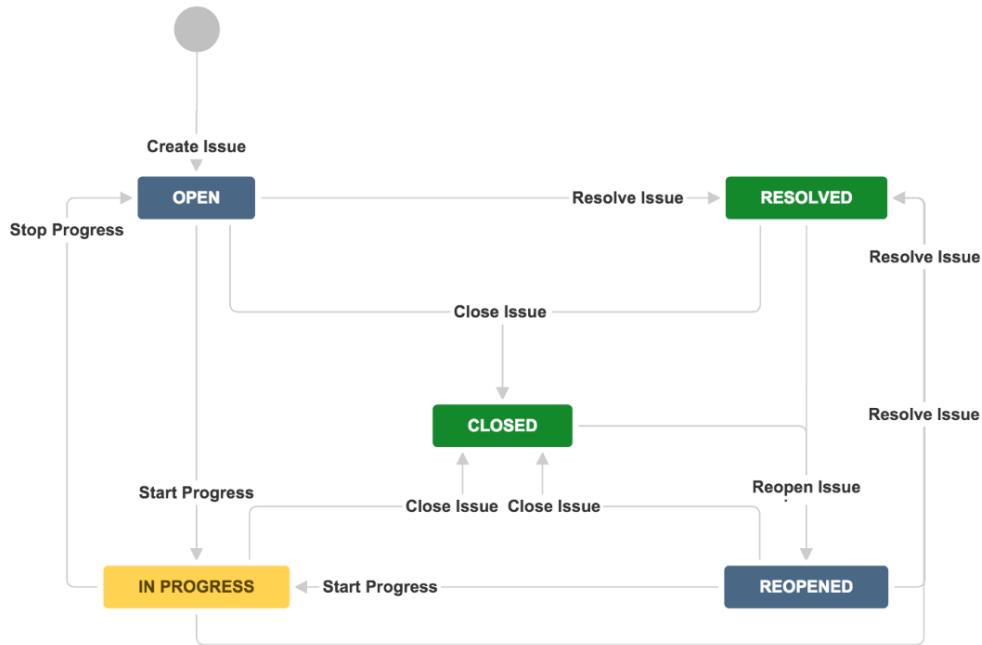


FIGURE 1.5: Workflow dans JIRA

Quelques références utilisant JIRA :

- Capgemini : Utilisé par les équipes de développement pour être plus agiles et pouvoir lancer de nouvelles fonctionnalités plus rapidement.
- FranceTV : utilisé par les équipes de contenu numérique pour partager des informations et offrir une visibilité à tous les niveaux, des développeurs aux scrum masters en passant par les éditeurs et les responsables techniques.

La figure 1.6 est un exemple de Sprint Backlog dans le projet *Joodperfumes.com* pour le suivi des "User story"

The screenshot shows the Jira Software interface for the 'JOD-DEV' project. The left sidebar includes links for Roadmap, Backlog, Board, Reports, Pages, Add item, and Project settings. The main area displays the 'JOD Sprint 2' backlog with four columns: BACKLOG (11 items), DOING (2 items), REVIEW (5 items), and DONE (3 items). Each item is a user story with a title, description, assignee, and status indicator (green checkmark for done, blue checkmark for doing). A modal at the top right says 'Hi, med aitsih. You're on the new Jira Software experience, which we're adding features to every day.' with 'Learn more' and 'Dismiss' buttons.

Column	Items
BACKLOG	11
DOING	2
REVIEW	5
DONE	3

FIGURE 1.6: Sprint Backlog *Joodperfumes.com* dans JIRA

1.4.4 Conduite du Projet

Souvent les projets n'atteignent pas leurs objectifs (dépassement de délais, surcoûts importants, qualité technique du produit insuffisante). Pour cela un processus de conduite de projet est nécessaire pour garantir une bonne gestion des moyens à disposition (temps, budget, information et ressources) afin de produire un système de bonne qualité.

Il semblait nécessaire de choisir une démarche pour assurer la bonne conduite de notre projet, Et c'est la méthode Scrum (figure 1.7) du groupe de méthodes AGILE qui a été adoptée. Le choix de cette démarche avait pour raison le découpage du projet en plusieurs itérations. A chaque itération une partie du projet aura été traitée et à la fin de l'itération une version autonome devra être livré en production.

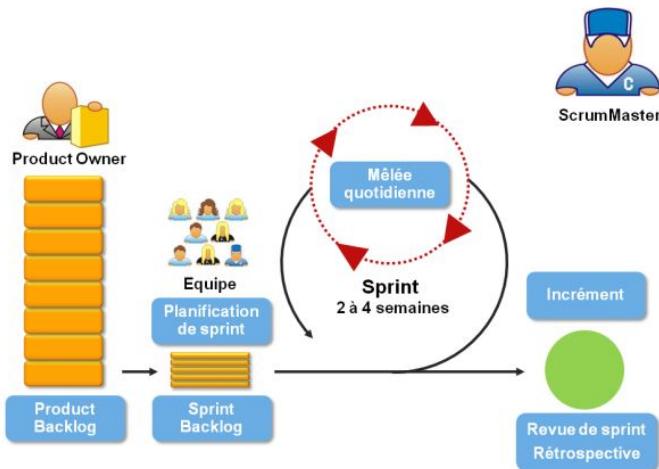


FIGURE 1.7: Méthode agile Scrum

En adoptant cette pratique, nous avons pu mener à bien maîtriser le projet, tenant compte de sa robustesse et de ses contraintes. Aussi, elle a garanti l'intégration progressive du projet en lui permettant d'être exposé au client tout au long du processus du développement. Ceci était d'une grande utilité pour réduire les risques et améliorer le produit tout au long du cycle de développement.

1.4.5 Planification du Projet

Pour atteindre l'objectif fixé, la planification est une phase indispensable. Sans cela, rien n'est maîtrisé, le chef de projet et son équipe, naviguerait à vue.

En effet, la planification a pour objectif d'organiser le déroulement des étapes du projet dans le temps. Une tâche fondamentale pour la maîtrise des délais.

Généralement pour planifier un projet, la première phase consiste à **le découper en plusieurs étapes**, d'en **estimer la durée**, d'identifier l'**enchaînement des étapes** (dont celles qui peuvent être conduites en parallèle), **affecter des ressources** (financières et humaines), et enfin **modéliser cette organisation sur un document opérationnel** partagé entre tous les acteurs concernés. La figure 1.8 résume, sous forme d'un diagramme de Gantt, le planning du projet que nous avons adopté durant la période de stage et ses différentes phases d'exécution.

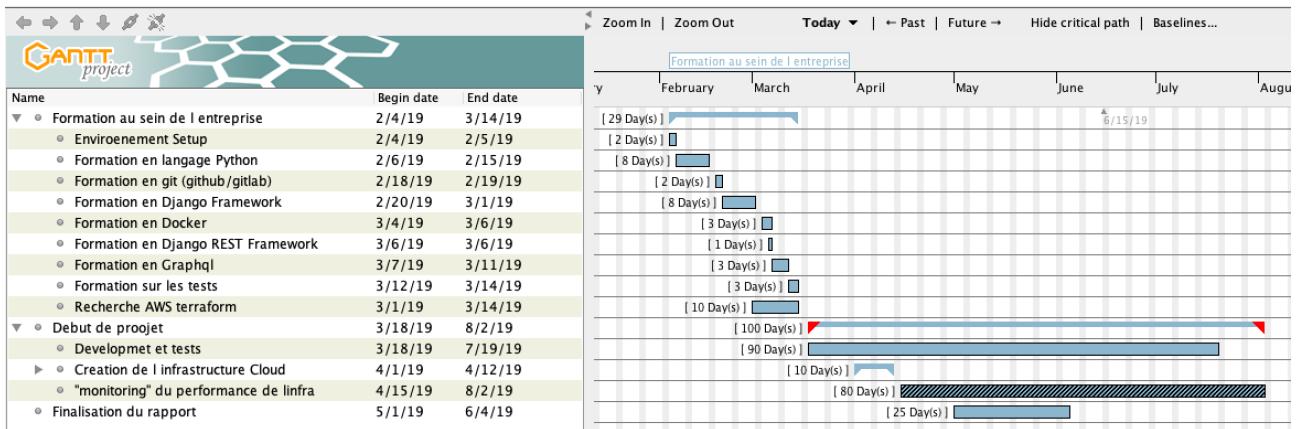


FIGURE 1.8: Diagramme de Gantt

1.5 Conclusion

Dans le présent chapitre, nous avons situé notre projet de fin d'études, en présentant d'une part la société Orange Bytes en termes d'activités et d'organisation, ainsi que les concepts clés de notre sujet. D'autre part nous avons présenté l'enchaînement et le déroulement de notre projet. La partie suivante présentera l'étude fonctionnelle de ce dernier.

Etude fonctionnelle - technique

2.1 Introduction

Ce chapitre est consacré à la description et l'étude fonctionnelle qui concerne l'exigence du projet, la problématique et les inconvénients imposés, nos solutions proposés en vue de résoudre cette problématique et d'atteindre les objectifs fixés.

Après l'élaboration de l'étude fonctionnelle, une étude technique devrait être menée. Elle a pour objectif principal de faire une capture de tous les besoins techniques liés au projet et de proposer une nouvelle architecture du système offrant une meilleure structuration de ses couches. C'est l'objectif de la deuxième partie de ce chapitre.

2.2 Spécification des besoins

Le projet consiste à la conception et le développement d'une boutique en ligne Joodperfumes ainsi que l'implémentation d'une infrastructure Cloud adaptée pour l'architecture spéciale de la plateforme ecommerce Jood qui est basée sur l'architecture des Services , qui peut être mise en place pour la commercialisation des parfumes dans monde arabe destinés principalement aux clients dans trois pays principales (Koweit,Arabie saoudite,Émirats arabes unis). Il est essentiel pour nous de préciser à ce stade que mon projet de fin d'étude tiendra compte de toutes ces contraintes tout en essayant présenter les solutions nécessaires dans le respect des règles d'un magasin virtuel telles que la simplicité de navigation entre les pages, la bonne ergonomie et la sécurité des données confidentielles des clients ainsi que de leurs paiements.

L'exposition des produits Notre site doit avoir une vitrine virtuelle à travers laquelle le client peut consulter une grande variété de produits. Il sera donc essentiel de présenter les prix et les caractéristiques techniques de chaque produit pour faciliter la sélection du produit à acheter.

L'inscription du client Jusque-là, le client est toujours anonyme mais pour pouvoir passer une étape plus rigoureuse, il doit être enregistré, cela n'est fait que pour la première commande mais après, notre client peut authentifier avec votre adresse e-mail et votre mot de passe pour passer d'autres commandes.

Ajout des produits au panier Après avoir choisi un produit, le client doit mentionner la quantité qui est automatiquement ajoutée à son panier avec le prix unitaire et le prix total.

Mode de livraison le client doit remplir soigneusement un formulaire contenant les informations nécessaires telles que :

- - Le nom du destinataire qui peut être le même client ou une autre personne.
- - l'adresse de livraison précise (pays, ville,)
- - numéro de téléphone

La confirmation de la commande Jusqu'à cette étape, nous avons un client, une commande et une adresse de livraison. Le chemin est maintenant plus claire, la commande ne passera pas seulement après validation de toutes les informations affichées dans l'interface mais bien après la phase de paiement.

La Sécurité Notre solution doit respecter en particulier la confidentialité des données personnelles des clients qui reste l'une des contraintes les plus importantes des sites Web.

2.3 Analyse fonctionnelle

L'étude ou l'analyse fonctionnelle est un élément indispensable pour la bonne réalisation d'un projet, puisqu'elle a pour but de recenser et valoriser toutes les fonctionnalités inhérentes au système étudié. Cela permettra d'une part de cerner exhaustivement les caractéristiques intrinsèques du système, d'autre part, de choisir adéquatement les mécanismes et techniques pour sa réalisation et sa mise en œuvre.

Cette étude fonctionnelle, identifie dans un premier temps les acteurs qui interviennent à l'élaboration de l'ensemble des processus du système, puis étudier dans un deuxième temps les différents diagrammes UML.

2.3.1 Identification des acteurs

Les acteurs qui interviennent dans le système représentent toute personne ayant accès à l'application, cette dernière peut être accessible depuis n'importe quel dispositif connecté à internet et doté d'un navigateur Web, nous pouvons donc distinguer deux types d'acteurs :

- **Administrateur** : C'est un super utilisateur possédant toutes les permissions (Ajout, modification et suppression) sur tous les modèles sans exceptions via l'interface d'administration.
- **Client** : Un acteur qui possède un compte sur le site, cet utilisateur peut accéder à toutes les fonctionnalités offertes par la plateforme et peut gérer les objets qui lui sont attribués (Profils, Notifications, etc.).
- **Internaute** : Un visiteur du site qui n'a pas de compte et donc par conséquent n'accède qu'à des fonctionnalités limitées (liste des produits et des catégories, détails d'un produit etc.).

2.3.2 Identification des différents modules (applications)

Cette architecture nous a permis de découper le projet en plusieurs services indépendants. Après une étude préalable détaillée, il s'est avéré qu'il serait optimal d'adopter la décomposition suivante :

- **Cart module** : qui a pour le but de gérer les articles et produits commander dans un panier
- **Notify module** : qui est une application de notification qui gère les fils d'actualités des autres modules pour un utilisateur donné. Il reçoit également et règle les flux de messages publiés, éléments d'information envoyés par d'autres services selon différents types, etc.
- **Odoo module** : le module représente plusieurs méthodes facilitant les interactions avec le ERP Odoo

- **Order module** : qui représente l'essence de la plateforme, c'est là où réside la gestion des commandes ,les modes et es adresses de livraison .
- **Payment module** : qui est chargé de la gestion de la facturation et les paiements des utilisateurs.
- **Product module** : gère les produits, emballages cadeau, catégorie et dépôt dans les différentes zones .
- **Profile module** : pour gérer les profils, l'inscription et l'authentification des utilisateurs.

2.3.3 Diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation décrivent, sous forme d'action et de réaction, le comportement d'un système du point de vue d'un utilisateur. Ils permettent de définir le fonctionnement et les limites du système ainsi que la description des interactions entre l'acteur et le système et donc lui permet d'atteindre son objectif en l'utilisant.

Client : Le tableau 2.1 détaille tous les cas d'utilisation d'un Client qui sont présentés dans le diagramme de cas d'utilisation dans la figure 2.1.

Cas d'utilisation	Opérations	But
Authentification	Login	Identifier l'utilisateur de l'application
	Logout	
Gérer un compte	Ajouter un compte	Pour avoir plus d'informations sur le profil d'un utilisateur qu'il peut les modifier
	Modifier un compte	
Créer une commande	Ajouter une adresse de livraison	Achat en ligne
	Annuler la commande	
	Valider la commande	
	Paiement de commande	
Gérer un panier	Ajouter un produit au panier	Gestion de panier
	modifier la quantité	
	Vider le panier	

TABLE 2.1: Scénarios des cas d'utilisation d'un Client

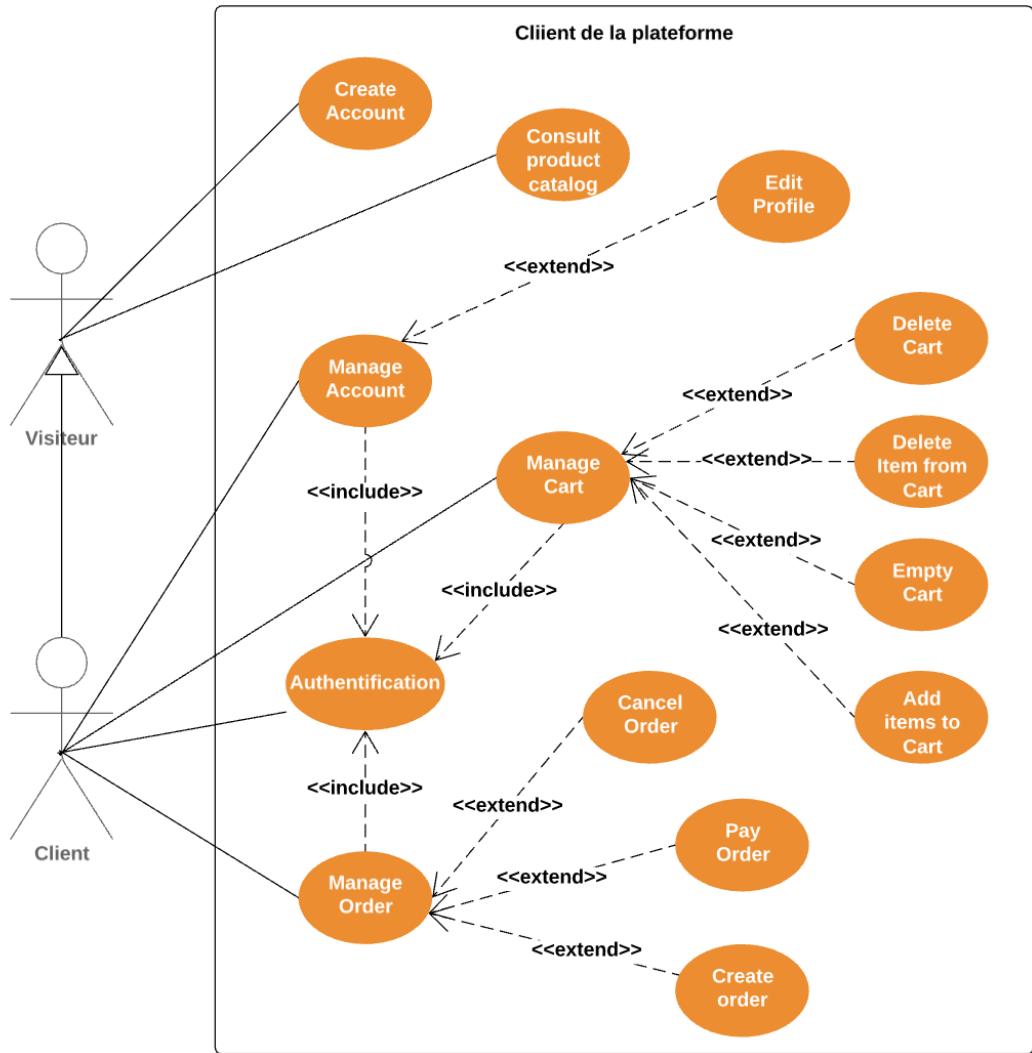


FIGURE 2.1: Diagramme de cas d'utilisation d'un Client

Admin : Le tableau 2.2 détaille tous les cas d'utilisation d'un admin du boutique qui sont présentés dans le diagramme de cas d'utilisation dans la figure 2.2.

Cas d'utilisation	Opérations	But
gérer les utilisateurs	Supprimer un compte client	—
Créer le catalogue	Ajouter une categorie	Assurer la mise à jour du catalogue
	Suprimer une catégorie	
Gérer les produits	Ajouter un produit	Gestion des produits
	Modifier un produit	
	Afficher la liste des produits	
	Suprimer un produit	
Gérer les Commandes	Suprimer une commande	Gestion des commandes
	Modifier l'état d'une commande	

TABLE 2.2: Scénarios des cas d'utilisation de l'admin du plateforme



FIGURE 2.2: Diagramme de cas d'utilisation de l'admin du plateforme

2.3.4 Diagrammes de séquence

Les diagrammes de séquences peuvent servir à illustrer un cas d'utilisation, par la suite nous détaillerons les principaux cas d'utilisation, à savoir l'authentification sur la plateforme, l'achat d'un produit et finalement l'interaction avec la méthode du paiement.

Authentification sur la plateforme (sign in) : La figure 2.3 représente le diagramme de séquences détaillé du cas d'utilisation d'authentification.

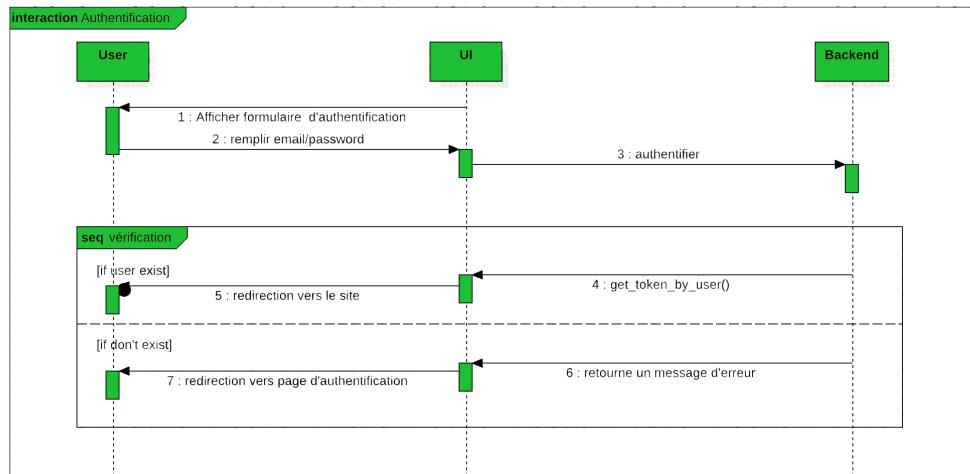


FIGURE 2.3: Diagramme de séquences : cas d'utilisation d'authentification

Achat sur la plateforme : La figure 2.4 représente le diagramme de séquences détaillé du cas d'utilisation d'achat sur Joodperfumes .

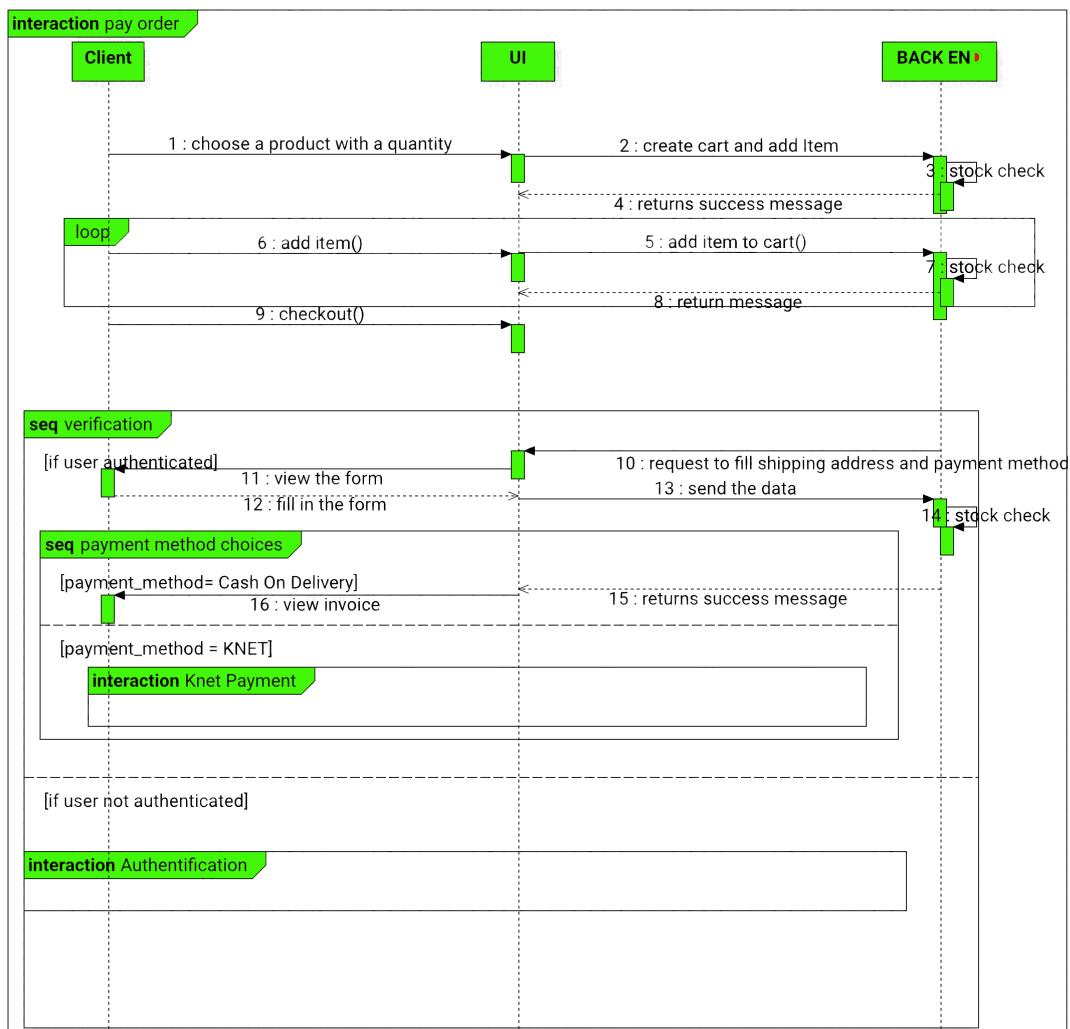


FIGURE 2.4: Diagramme de séquences : cas d'utilisation d'achat sur la plateforme

Interaction avec la méthode KNET : La figure 2.5 représente le diagramme de séquences détaillé du cas d'utilisation de paiement avec KNET.

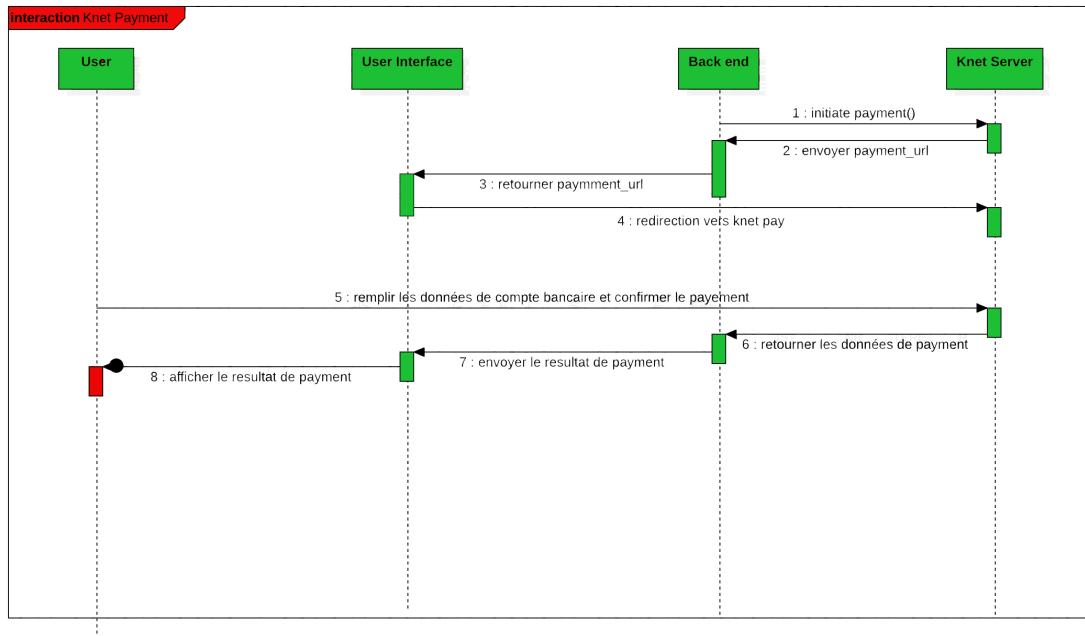


FIGURE 2.5: Diagramme de séquences : cas d'utilisation de paiement via KNET

2.3.5 Diagrammes de classes

A cette étape il faut mettre à jour les principales abstractions du système sous forme d'objets et de classes ainsi que les différentes relations entre celles-ci.

Pour une raison de clarté, nous présenterons les différents dans un diagramme de classe global, avec la description de chacune des classes sous forme de tableaux.

La figure 2.6 représente le diagramme de classes de plateforme, qui sont détaillées dans le tableau 2.3

Classe	Description
User	Contient les informations d'un profil utilisateur inscrit sur la plateforme.
Product	Regroupe l'ensemble des informations propres à un produit.
Cart	Le panier d'un utilisateur contient l'ensemble des produits qu'il aura sélectionnés préalablement.
CartItem	Un article du panier désigne par un produit et la quantité de demandé
Order	C'est panier prêt à payer.
OrderItem	Les élément de panier qui compose un ordre .
ShippingAddress	L'adresse de distribution à un consommateur.
PaymentMethod	C'est le méthode de paiement pour un ordre généralement il ya deux méthodes Cash On Delivery et Payment Online .
GiftWrapping	C 'est la classe qui représente l'emballage Cadeau pour un produit
Category	C'est la classe qui représente les catégories existants des produits.
Warehouse	C'est la classe qui représente les entrepôts où existe les produits.
Site	C'est la classe qui représente les pays .
GiftWrappingSite	C'est la classe qui représente le caractéristique de l'emballage cadeau dans un pays (principalement le prix).

TABLE 2.3: Identification des Classes

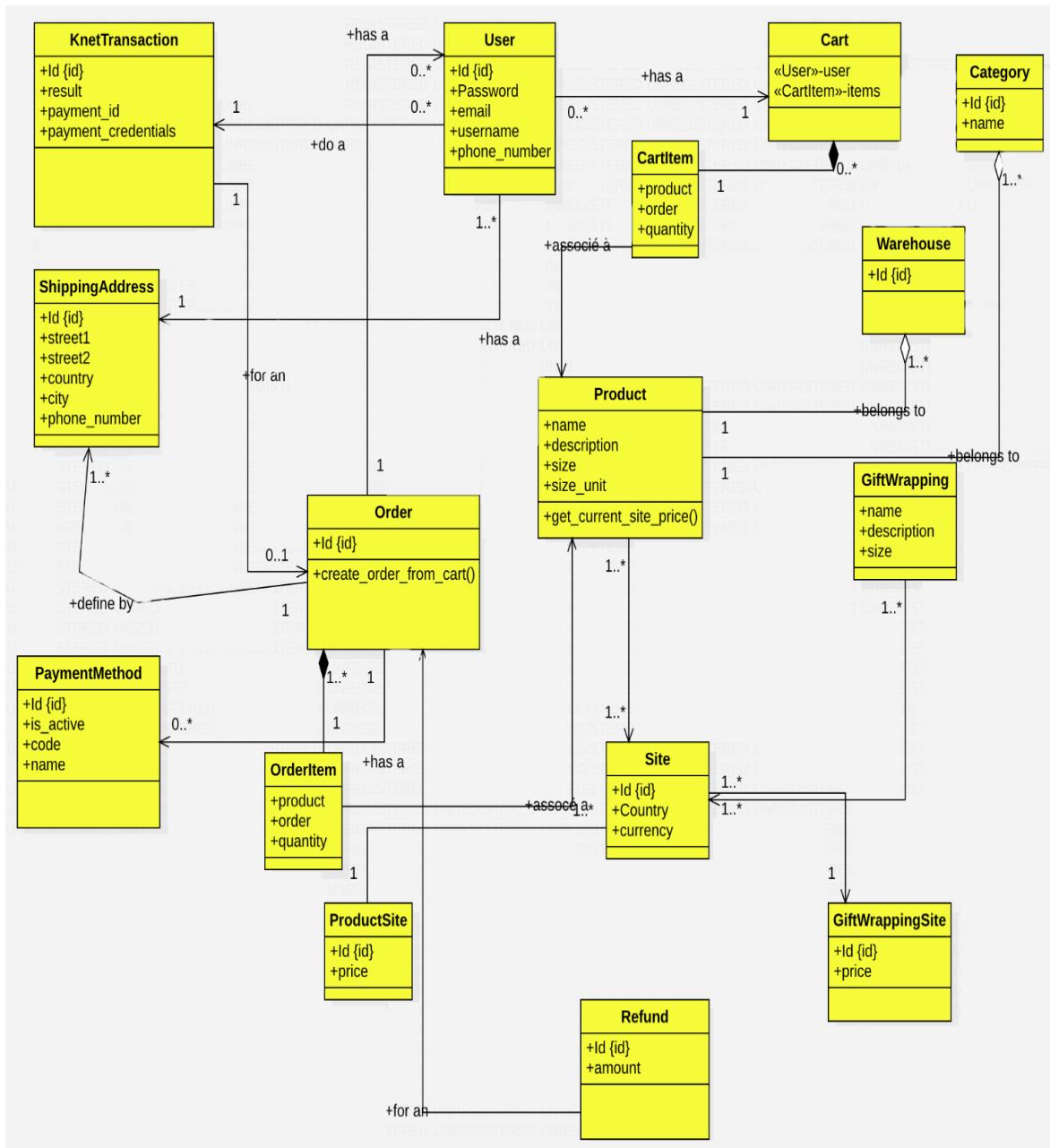


FIGURE 2.6: Diagramme de classes du site

2.4 Etude technique du projet

Cette partie traite les besoins en termes de logiciel. Elle commence d'abord par présenter les modèles de conception utilisés puis aboutit à une présentation de l'architecture logique adoptée.

2.4.1 Modèles de conception adoptés

Un modèle de conception « Design Pattern » décrit une meilleure pratique ou une solution prouvée qui permet de résoudre un problème commun ayant tendance à se reproduire, en mettant l'accent sur le contexte et sur les conséquences et impacts de la solution.

Certains modèles de conception sont utilisés dans ce projet, notamment, le modèle MVC (Model View Controller) et Active Record.

1. Le pattern M.V.C :

Le Modèle Vue Contrôleur est un modèle de conception logicielle dont le principe est de séparer les données, la présentation et les traitements. Ainsi l'application se trouve décomposée en trois couches essentielles :

- Le Modèle : Représente la partie de l'application qui exécute la logique métier. Cela signifie qu'elle est le responsable de récupérer les données, de les convertir selon des concepts chargés de sens pour votre application, tels que le traitement, la validation, l'association et beaucoup d'autres tâches concernant la manipulation des données.
- La Vue : La Vue retourne une présentation des données venant du model. Etant séparée par les Objets « Model », elle est responsable de l'utilisation des informations dont elle dispose pour produire une interface de présentation de l'application.
- Le Contrôleur : Gère les requêtes des utilisateurs. Elle est responsable de retourner une réponse avec l'aide mutuelle des couches Model et Vue.

La figure 2.8 schématise le pattern MVC.

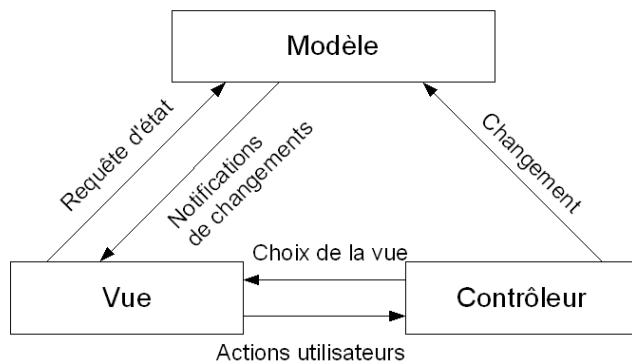


FIGURE 2.7: Architecture MVC

2. Le modèle Active Record :

Le design pattern Active Record est une approche pour faire l'abstraction entre le support de stockage et les modèles sans l'intervention d'autres classes de manipulation. Elle permet donc l'écriture et la lecture des données depuis les objets.

La figure 2.8 schématise et illustre ce patron de conception.

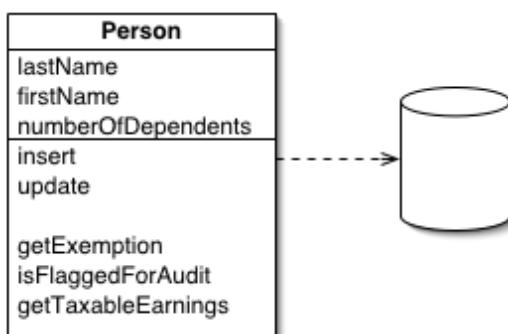


FIGURE 2.8: Design Pattern Active Record

2.4.2 Architecture Global - Applications

Joodperfumes est un projet qui est décomposé en plusieurs applications/services faiblement couplés, souvent spécialisés dans une seule tâche. Les processus indépendants communiquent les uns avec les autres dans un seul projet.

Pour s'attaquer au problème de la complexité on a adopté cette architecture de modèles. Cette architecture nous a permis de décomposer ce qui pourrait être une application monolithique, en plusieurs services, chacun indépendant de l'autre .

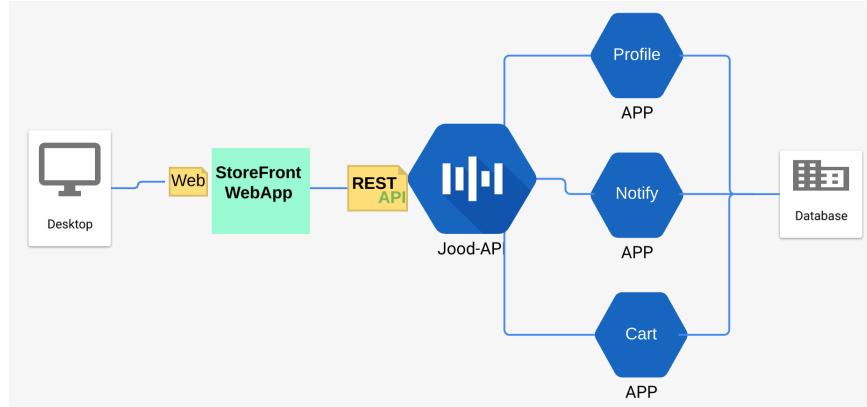


FIGURE 2.9: Exemple d'architecture microservices

2.5 Conclusion

Cette étape été nécessaire pour la structuration de la couche métier, Ainsi nous avons raffiné les diagrammes de classe d'analyse, définit des modules et des composants.

Ensuite, nous avons traité l'étude technique du projet, au cours de laquelle nous avons détaillé les caractéristiques techniques de la solution escomptée qui sera utilisé dans la réalisation et mise en oeuvre qui fera l'objet du chapitre suivant.

Technologies Utilisées

3.1 Introduction

Dans ce chapitre, je présenterai les principales technologies que nous utilisons avec Joodperfumes. Les technologies sont divisées par leur portée. Certaines sont spécifiques aux opérations et l'autre au développement, aux tests, au développement frontal et à la surveillance.. Le choix de l'environnement technique du travail est une étape primordiale avant d'entamer la phase de l'implémentation. Cette partie décrit l'ensemble des outils technologiques utilisés lors de développement du système.

3.2 Outils d'infrastructure Cloud

AWS : Amazon Web Services (AWS) est une division du groupe américain de commerce électronique Amazon.com, spécialisée dans les services de cloud computing à la demande pour les entreprises et particuliers .

AWS propose plus de 90 services, comprenant le calcul, le stockage, le réseau, la base de données, l'analyse de données, des services applicatifs, du déploiement, de la gestion de système, de la gestion d'applications mobiles, des outils pour les développeurs et pour l'internet des objets.



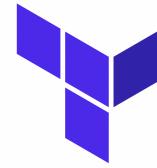
Cloudflare : CloudFlare est une technologie d'optimisation qui utilise plusieurs techniques pour vous permettre d'optimiser les performances de votre site sans disposer de toute la gestion et de l'administration nécessaires à son accomplissement. Il réduit le temps de chargement de votre site Web, augmente sa sécurité et son efficacité et améliore ses performances globales.



Terraform : Terraform est un outil créé en 2014 par HashiCorp, la société qui a créé d'autres outils : Consul, Vagrant, Vault, Atlas, Packer et Nomad. C'est un outil open source, écrit en Go, avec une communauté active de plus de 1200 contributeurs, plus de 15 000 stars sur Github, et qui repose sur une architecture basée sur les plugins.

Terraform (TF) est un outil qui permet de construire, modifier et versionner une infrastructure.

Contrairement à ce que l'on peut lire sur Internet, la technologie n'est pas « plateforme agnostique », mais elle permet d'utiliser plusieurs providers dans un même template de configuration. Il existe en effet des plugins pour des providers de Cloud, des services d'hébergement, des SCM... Nous le verrons un peu plus tard dans ce chapitre.



Que fait l'outil ?

- Il assure la création et la cohérence d'infrastructure.
- Il permet d'appliquer des modifications incrémentales.
- On peut détruire des ressources si besoin.
- On peut prévisualiser les modifications avant de les appliquer.

Terraform permet de créer automatiquement son infrastructure, comme des machines virtuelles, des lambdas Amazon Web Services (AWS)... le tout, simplement, en trois étapes :

La figure 4.3 schématisise le Pipeline d'exécution de Terraform.

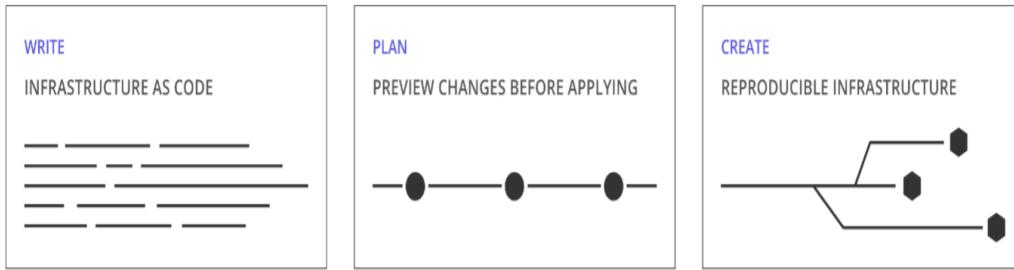


FIGURE 3.1: Pipeline d'exécution de Terraform

1. Write : écriture de la définition de ses ressources dans des fichiers au format *.tf;
2. Plan : un plan des ressources à créer/modifier supprimer est affiché, avant tout changement ;
3. Create : l'infrastructure voulue est mise en place, et reproduitible dans tous les environnements souhaités

Tunnelblick Tunnelblick est une interface utilisateur graphique gratuite et open source pour OpenVPN® sous macOS. Il permet de contrôler facilement les connexions client et/ou serveur OpenVPN.



3.3 Outils de développement -Backend

Le développement d'applications web et de sites Internet présente des points communs. D'où l'utilisation des Framework qui vient avec un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes d'une partie d'un logiciel (architecture). Notre projet se base totalement sur des technologies open-source.

Langage Python Python est un langage de programmation orienté Object similaire au Perl et Ruby. Sa force majeure est qu'il propose des milliers de modules qui vous feront gagner un temps précieux durant le développement de vos applications.

Python est un langage interprété, c'est à dire qu'il n'est pas compilé. L'interpréteur Python exécute directement des instructions écrites dans un fichier source contenant du texte. Bien-sûr des séries d'optimisations permettent de garder une version entre le binaire et la source pour ne pas réinterpréter tout le texte à chaque exécution.



Les applications Python sont donc directement cross-platform puisqu'il suffit d'installer l'interpréteur Python sur un des très nombreux systèmes supportés pour que l'application puisse tourner.

Django Framework Django est un framework web Python de haut niveau qui encourage le développement rapide et propre. Gratuit et open source, Django vous permet d'éviter de réinventer la roue grâce à toutes les librairies disponibles en Python, mais aussi de tout ce que ce framework offre dès son installation.



Django a été créé en 2005 et a beaucoup évolué depuis. Il dispose d'une bonne maturité, et d'une bonne stabilité grâce à son système de branche test, et stable. La documentation est claire quant aux versions et branches, ce qui est souvent oublié dans les documentations de logiciels informatiques. Les développeurs veulent souvent garder la documentation au niveau de la dernière version.

SGBD : PostgreSQL PostgreSQL est un système de gestion de bases de données relationnel robuste et puissant, aux fonctionnalités riches et avancées, capable de manipuler en toute fiabilité de gros volumes de données, même dans des situations critiques.



Il est extrêmement respectueux des standards, se conformant au plus près à la norme ANSI-SQL 2008. Il supporte plus d'une douzaine de langages de programmation, dont Java, Perl, Python, Ruby, Tcl, C / C ++. Son propre PL / pgSQL est similaire à PL / SQL d'Oracle.

Open-source, il est développé par une communauté mondiale regroupant des milliers de développeurs et plusieurs dizaines d'entreprises.

Docker Docker est une plate-forme ouverte pour le développement, l'embarquement (shipping) et l'exécution d'applications. Docker permet de séparer les applications de l'infrastructure afin qu'on puisse livrer rapidement un logiciel. Avec Docker, on peut gérer l'infrastructure de la même façon qu'on géré ses applications. En profitant des méthodologies de Docker pour l'embarquement, le test et le déploiement rapide du code, on peut réduire considérablement le délai entre l'écriture du code et son exécution en production.



Compose Compose est un outil permettant de définir et d'exécuter des applications Docker à conteneurs multiples. Avec Compose, nous pouvons utiliser un fichier YAML pour configurer les services de notre application. Ensuite, avec une seule commande, nous pouvons créer et démarrer tous les services de notre configuration. JOOD-CORE fonctionne avec Docker Compose dans les processus de développement et de test .



Git & GitHub Github est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git, amène chaque contributeur à télécharger les sources du projet et à proposer ensuite ses modifications à l'équipe du projet. Le site offre aussi un logiciel de suivi de problèmes (issues tracking system). GitHub propose aussi l'intégration d'un grand nombre de services externes, tels que l'intégration continue (continuous integration), la gestion de versions, badges, chat basés sur les projets, etc.



La figure 3.2 représente l'organisation Jood (de notre projet) sur GitHub.

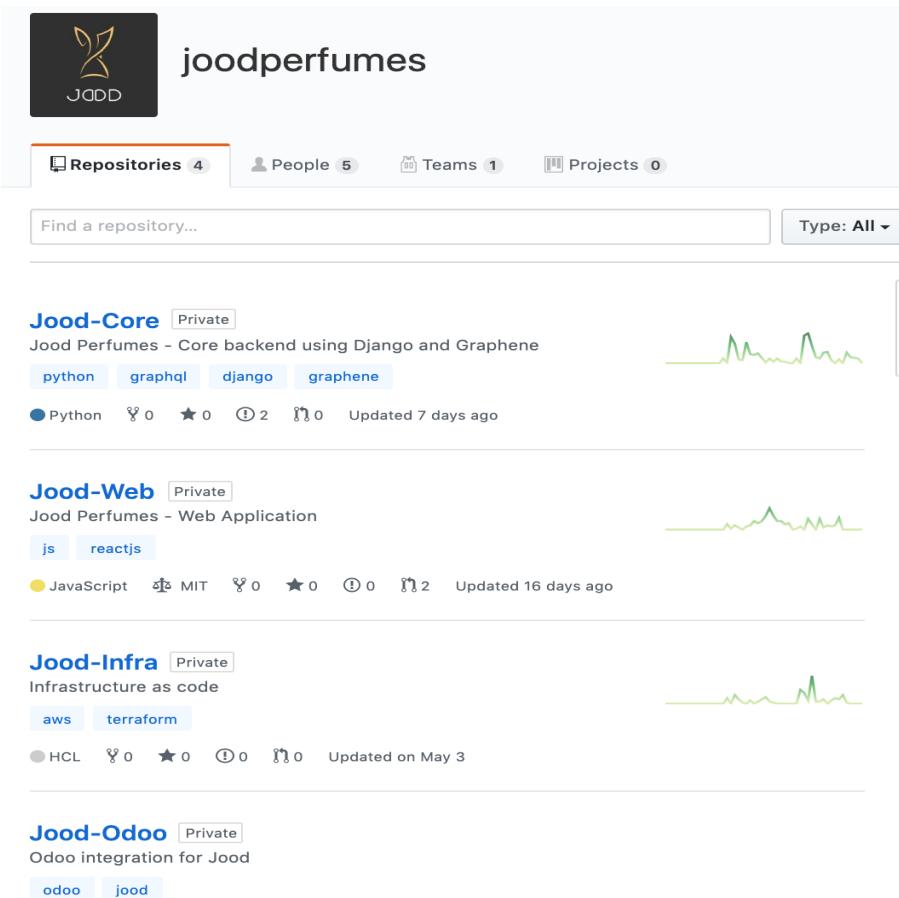


FIGURE 3.2: Aperçu de jood sur GitHub

GraphQL-Graphene GraphQL est un langage de requêtes développé en interne par Facebook en 2012 jusqu'à sa publication en 2015². Il propose une alternative aux API REST¹ dont le stockage est éventuellement distribué. Il propose au client de formuler la structure de données dans la requête, tandis que cette même structure est renvoyée par le serveur. Fortement typé, ce langage évite les problèmes de retour de données insuffisants (under-fetching) ou surnuméraires (over-fetching).



GraphQL a été implémenté pour de nombreux langages, par exemple Graphene³ pour Python, Apollo⁴ et Relay⁵ pour Javascript.

Il est utilisé par certaines bases de données orientées graphe comme ArangoDB en tant que langage de requête⁶.

Graphene Graphene-Python est une bibliothèque permettant de créer facilement des API GraphQL en Python. Son objectif principal est de fournir une API simple mais évolutive facilitant la vie des développeurs.

Celery est une file d'attente de tâches / travaux asynchrone basée sur le transfert de messages à un courtier, tel que RabbitMQ . Il est axé sur le fonctionnement en temps réel, mais prend également en charge la planification. Les unités d'exécution, appelées tâches, sont exécutées simultanément sur un ou plusieurs serveurs de travail utilisant le multitraitemet, Eventlet ou gevent. Les tâches peuvent s'exécuter de manière asynchrone (en arrière-plan) ou synchrone (attendre d'être prêt).



Celery est un système Python Task-Queue qui gère la répartition des tâches sur les travailleurs via des threads ou des nœuds de réseau. Cela facilite la gestion des tâches asynchrones.

RabbitMQ RabbitMQ est une solution de messagerie orientée messages, ou solution Message-Oriented Middleware (MOM). Un middleware est un logiciel tiers permettant de créer un réseau d'échange d'informations entre des applications. La technique d'échange d'informations utilisée par RabbitMQ est l'échange de messages.



Le terme MQ, Message Queueing, présent dans RabbitMQ souligne cette technique et par ailleurs la principale fonctionnalité du produit. Il représente la connexion d'applications par le biais de messages routés grâce à un broker. On peut comparer le broker à La Poste, c'est-à-dire qu'il reçoit un message d'une application et le délivre à une autre. Nous utilisons RabbitMQ dans l'infrastructure JOOD parce qu'elle répond à ces exigences d'application modernes. Nous l'utilisons comme intermédiaire entre Celery et **Jood-Core** .

Jood-Core doit simplement envoyer des messages à RabbitMQ, et les employés de Celery les afficheront et planifieront l'exécution des tâches.

La figure 3.3 illustre un exemple d'Appel d'une Tache par l'application executer avec Celery et RabbitMQ.

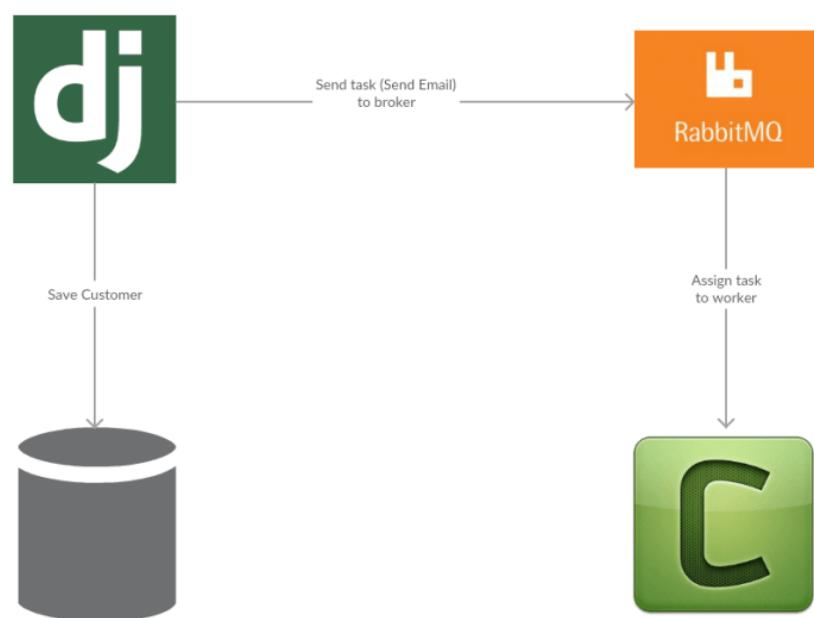


FIGURE 3.3: Appel d'une Tache celery

XML-RPC XML-RPC est un protocole RPC (remote procedure call), une spécification simple et une nsemble de codes qui permettent de s'exécuter dans des environnements différents defaire des appels de méthodes à travers un réseau.XML-RPC permet d'appeler une fonction sur un serveur distant de n'importe quel système(Windows, Mac OS X, GNU / Linux) et avec n'importe quel langage de programmation. Le serveur est lui-même sur n'importe quel système et est programmé dans n'importe quel langage.Cela permet de fournir un service Web utilisable par tout le monde sans restriction de système ou de langage.Les processus d'invocation à distance ont utilisé le protocole HTTP pour le transfert de données et la norme XML pour la structuration des données.XML-RPC est conçu pour permettre aux structures de données d'être transmis, exécutées et renvoyées très facilement.XML-RPC est une alternative aux services web WS- *, ne SOAP.

On a utilisé XML-RPC dans JOOD-CORE pour communiquer avec JOOD-ODOO le serveur de JOOD dédié à OODOO

Odoo Odoo, anciennement OpenERP, est une société de logiciels open source fondée en 2004 qui propose une suite complète de modules de gestion d'entreprise entièrement intégrés.

Odoo est le programme de gestion d'entreprise le plus évolutif et le plus installé au monde avec des applications qui répondent à tous les besoins d'une entreprise, de la gestion de la relation client à la création de sites Web, en passant par le commerce électronique. commerce, par la production, la gestion des stocks, la comptabilité, les ERP, etc., parfaitement intégrés. C'est la première fois qu'un éditeur de logiciel atteint un tel niveau de fonctionnalité.

La figure 3.4 représente les principales fonctionnalités de ERP Odoo



FIGURE 3.4: Les principales fonctionnalités de ERP Odoo

Pourquoi Odoo ?

- Solution ERP Open Source : le code source est disponible à tout moment, votre ERP n'est donc pas une boîte noire inaccessible ;
- il s'agit d'un ERP flexible : une mise en oeuvre progressive en fonction de l'évolution de vos besoins est donc tout à fait possible ;
- La solution ERP est très facilement interopérable avec tout logiciel et / ou service tiers (notamment via le connecteur générique développé par Camptocamp) ;

- S'appuie sur un éditeur solide de plus de 600 employés, une communauté dynamique et structurée (offrant plus de 20 000 modules de communauté) et un réseau mondial de plus de 1 000 partenaires certifiés.

KNETPAY est la solution de passerelle KNET Payment Gateway pour les marchands koweïtiens et les sociétés de services disposant de capacités de commerce électronique. Le service prendra en charge les transactions par carte bancaire de marque KNETco sur le Web. KNET présentera les cartes de crédit à l'avenir. Le service KNET Payment Gateway (PG) fournira une facilité de paiement pour les services marchands ou les biens. Les principales caractéristiques du service KNETPG sont les suivantes : oPrise en charge des cartes de débit de marque KNETco sur une page de paiement sécurisée hébergée sur KNET oAutorisation en ligne des transactions de paiement via une intégration de plug-in marchand-gateway



IDE : PYCharm On attend principalement d'un IDE qu'il regroupe dans une même interface les fonctionnalités indispensables à la réalisation et au suivi du projet tout au long de son cycle de vie. On y retrouve donc bien évidemment un éditeur performant simplifiant l'écriture de code, mais aussi idéalement l'intégration d'un débogueur, l'interfaçage avec un outil de gestion de versions, l'interfaçage avec un outil de gestion de tâches, la prise en charge des tests unitaires ou encore des outils de déploiement. PyCharm intègre tout cela de façon intuitive et efficace. Développé par l'équipe de JetBrains, c'est une équipe qui met dans ses produits tout leur savoir-faire et leur passion pour le développement informatique. Tout en étant très orienté professionnel.



3.4 Outils de Tests

Tester les applications est devenu un ensemble de compétences standard requis par tout développeur compétent. La communauté Python accepte les tests et même la bibliothèque standard Python possède de bons outils intégrés pour prendre en charge les tests. Dans le grand écosystème Python, il existe de nombreux outils de test.

Les tests ne font pas exception. Pour les monolithes, il existe des tests unitaires, des tests décomposants et des tests d'intégration. Les limites sont claires, la façon d'écrire des tests est également claire.

PYTEST – UNIT TESTING PyTest est un outil de test Python robuste qui peut être utilisé pour tous les types et niveaux de tests logiciels. PyTest peut être utilisé par les équipes de développement, les équipes d'assurance qualité, les groupes de test indépendants, les utilisateurs du logiciel TDD et les projets open source. En fait, des projets partout sur Internet sont passés de test unitaire, notamment Mozilla et Dropbox. Pourquoi ? Parce que PyTest offre des fonctionnalités puissantes telles que la réécriture « assert », un modèle de plug-inters et un modèle de montage puissant et simple, inégalé dans aucun autre framework de test. Dans **OBytes**, chaque développeur est à la fois un testeur et un développeur. Lors de la création d'une nouvelle fonctionnalité pour Jood, nous nous assurons que les tests sont écrits en premier, puis nous développons la fonctionnalité.



VCRPY – SERVICE MOCKING VCR.py simplifie et accélère les tests des requêtes HTTP. La première fois que nous exécutons du code contenu dans un gestionnaire de contexte VCR.py ou une fonction décorée, VCR.py enregistre toutes les interactions HTTP ayant lieu par le biais des bibliothèques. Il prend en charge, sérialise et écrit dans un fichier plat (au format yaml par défaut). Ce fichier plat s'appelle une cassette. Lorsque le code correspondant est exécuté à nouveau, VCR.py lit les requêtes et les réponses sérialisées à partir du fichier de cassette susmentionné, intercepte les requêtes

HTTP qu'il reconnaît à partir de l'exécution du test d'origine et renvoie les réponses correspondant à ces requêtes. Cela signifie que les demandes ne généreront pas de trafic HTTP, ce qui présente plusieurs avantages, notamment :

- La possibilité de travailler hors ligne
- Tests complètement déterministes
- Augmentation de la vitesse d'exécution du test

Si le module que nous testons modifie constamment son API, il suffit de supprimer nos fichiers de cassettes existants et de réexécuter nos tests. VCR.py détectera l'absence de fichier de cassette et enregistrera à nouveau toutes les interactions HTTP, qui les actualisera pour correspondre à la nouvelle API. Nous utilisons vcr.py pour simuler des appels http à une externe API.

FAKER – DATA GENERATION Lors de l'écriture de tests unitaires, nous rencontrons une situation dans laquelle nous devons générer des données de test ou utiliser des données factices dans nos tests. Si nous avons déjà des données quelque part dans une base de données, une solution que nous pourrions utiliser consiste à générer un vidding de ces données et à les utiliser dans nos tests (par exemple, les fixtures). Cependant, nous pourrions également utiliser un paquet tel que faker pour générer très facilement de fausses données lorsque nous en aurons besoin. Le seul but de Faker est de créer de fausses données semi-aléatoires. Faker peut créer de faux noms, adresses, agents utilisateurs de navigateur, domaines, paragraphes et bien plus encore.

CODECOV – CODE COVERAGE CODECOV décrit le degré de test du code source d'un programme. Nous utilisons CodeCov pour mesurer la couverture de code de notre application . Il surveille le projet , notant quelles parties du code ont été exécutées, puis analyse la source pour identifier le code qui aurait pu être exécuté mais ne l'a pas été. La mesure de la couverture est généralement utilisée pour évaluer l'efficacité des tests. Il peut indiquer quelles parties du code sont soumises à des tests et les quelles ne le sont pas. JOOD Core code doit avoir une couverture de 98% pour être fusionné à la branche principale.

CONFTEST – Definir des Fixtures Utiliser PyTest est génial et le support pour les montages de test est assez impressionnant. Cependant, afin de partager nos appareils sur l'ensemble du module, PyTest suggère de définir tous les appareils dans un seul fichier conftest.py. Ce n'est pas pratique si nous avons un grand nombre de fixtures. Pour une meilleure organisation et une meilleure lisibilité, nous préférerions définir nos fixtures sur plusieurs fichiers bien nommés .

3.5 Outils de development Front-end

React aussi appelé (React.js ou ReactJS) est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état. React est une bibliothèque qui ne gère que l'interface de l'application, considéré comme la vue dans le modèle MVC. Elle peut ainsi être utilisée avec une autre bibliothèque ou un framework MVC comme AngularJS. La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité.



La bibliothèque est utilisée notamment par Netflix (pour gagner 50 % de performance), Yahoo, Airbnb, Sony, Atlassian ainsi que par les équipes de Facebook, appliquant le dogfooding sur le réseau social éponyme, Instagram ou encore WhatsApp. À la fin de 2015, WordPress.com annonce Calypso, une interface pour les administrateurs WordPress, développée en JavaScript avec node.js et React

GatsbyJS



Les sites Web statiques sont des sites dont le contenu est fixe. Techniquement, il est composé d'une simple liste de fichiers HTML, qui affiche les mêmes informations quel que soit le visiteur.

Contrairement aux sites dynamiques, il ne requiert ni programmation back-end ni base de données. Pour faciliter leur création, de nombreux générateurs de sites statiques sont disponibles : Jekyll, Hugo, Hexo, etc. La plupart du temps, le contenu est géré via des fichiers (idéalement en format Markdown) statiques ou via une API de contenu. Ensuite, le générateur requête le contenu, l'injecte dans les templates définis par le développeur et génère un ensemble de fichiers HTML, CSS et JavaScript.

Les Progressive Web Apps (PWA) sont des applications web, reposant fortement sur l'utilisation de JavaScript, qui se veulent être fiables, rapides et engageantes. Étant donné qu'elles rendent la navigation plus rapide et qu'elles offrent une expérience utilisateur bien meilleure, les PWA sont devenues la manière par défaut de construire des interfaces web. Pour cette raison, de nombreux frameworks front-end ont fait leur apparition au cours des dernières années : Angular, React, et plus récemment, Vue.

Gatsby : quand les sites statiques rencontrent les Progressive Web Apps. Les sites Web statiques et PWA ont tous les deux de solides avantages, ce qui nous incite à trouver un moyen de les utiliser tous les deux dans le même projet. Heureusement, quelques outils ont réussi à faire le pont entre les deux, notamment celui dont on entend le plus parler depuis quelques mois : **Gatsby**

Gatsby est un générateur de site statique (SSG) basé sur React. Il s'agit d'un projet open Source (GitHub) né en mai 2015 auquel ont contribué plus de 1350 développeurs en 7000 commits à ce jour. Début 2018 Kyle Mathews crée Gatsby Inc pour soutenir le développement du projet.

ECMAScript 6 / JavaScript

ECMAScript ECMAScript 2015 est la version actuelle pour le langage ECMAScript. C'est le standard qui définit l'implémentation de JavaScript, notamment pour SpiderMonkey, le moteur utilisé par Firefox et les autres applications Mozilla. Cette version est appelée avec le nom de code « ES.next », « Harmony » ou « ECMAScript 6 ». Les brouillons de spécification (aussi appelés drafts) se trouvent sur le wiki officiel d'ECMA. Le premier draft basé sur ECMAScript 5.1, a été publié le 12 juillet 2011 sous le nom « ES.next ». À partir d'août 2014, ECMAScript 2015 n'a plus accepté de nouvelles fonctionnalités et a été stabilisé. Il a été officiellement approuvé et fut publié le 17 juin 2015 par l'assemblée générale ECMA.



JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs, avec l'utilisation (par exemple) de Node.js.



C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe. Le langage supporte le paradigme objet, impératif et fonctionnel. JavaScript est le langage possédant le plus large écosystème grâce à son gestionnaire de dépendances npm, avec environ 500 000 paquets en août 2017.

NodeJS est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge. Elle utilise la machine virtuelle V8 et implémenté sous licence MIT les spécifications CommonJS. Parmi les modules natifs de Node.js, on retrouve http qui permet le développement de serveur



HTTP. Il est donc possible de se passer de serveurs web tels que Nginx ou Apache lors du déploiement de sites et d'applications web développés avec Node.js. Concrètement, Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur. Node.js est utilisé notamment comme plateforme de serveur Web, elle est utilisée par Groupon, Vivaldi, SAP, LinkedIn, Microsoft, Yahoo!, Walmart, Rakuten, Sage et PayPal.

Flux est l'architecture applicative utilisée par Facebook pour la construction d'applications Web côté client. Il complète les composants de vue composables de React en utilisant un flux de données unidirectionnel. Il s'agit plus d'un modèle que d'un cadre formel et vous pouvez commencer à utiliser Flux immédiatement sans beaucoup de nouveau code.

- **Les actions** sont des objets simples avec une propriété de type et des données. Par exemple, une action peut être type : IncrementCount, données_local : delta : 1
- **Le Stores** contiennent l'état et la logique de l'application. La meilleure abstraction est de penser aux stores comme à la gestion d'un domaine particulier de l'application
- **Le Dispatcher** agit comme un hub central. Le répartiteur traite les actions (par exemple, les interactions utilisateur) et invoque les callbacks que les stores ont enregistrés.
- **Les vues** sont très courantes dans la plupart des modèles d'interface graphique MVC. Ils écoutent les changements dans les Stores et se rendent de manière appropriée. Les vues peuvent également ajouter de nouvelles actions au dispatcher, par exemple lors d'interactions entre utilisateurs.

Redux est un conteneur du “state” pour les applications JavaScript.



Il vous aide à écrire des applications qui se comportent de manière cohérente, fonctionnent dans différents environnements (client, serveur et natif) et sont faciles à tester. En plus de cela, il fournit une grande expérience de développement, telle que l'édition de code en temps réel combinée à un débogueur temporel. Redux peut être utilisé avec React ou avec toute autre bibliothèque de vues. Il est minuscule (2 Ko, dépendances comprises), mais dispose d'un vaste écosystème d'addons disponibles.

JSX React peut s'écrire avec la “véritable syntaxe” JavaScript mais cela est moins facile à écrire et encore moins à lire. JSX apporte un véritable souplesse dans l'écriture, dans la lecture et la compréhension. Babel propose un preset pour JSX ce qui permet de transpiler le JSX en code JavaScript classique et exécutable par le navigateur.



JSX est une extension de syntaxe de type XML à ECMAScript sans sémantique définie. Il n'est pas destiné à être implémenté par les moteurs ou les navigateurs. Ce n'est pas une proposition d'intégrer JSX dans la spécification ECMAScript elle-même. Il est destiné à être utilisé par divers préprocesseurs (transpilers) pour transformer ces jetons en ECMAScript standard.

pourquoi JSX ? React comprend le fait que la logique de rendu est intrinsèquement couplée à une autre logique d'interface utilisateur : mode de gestion des événements, évolution de l'état dans le temps et mode de préparation des données pour l'affichage.

Au lieu de séparer artificiellement les technologies en plaçant les balises et la logique dans des fichiers séparés, React dissocie les problèmes avec des unités faiblement couplées appelées «composants»

qui contiennent les deux. React ne nécessite pas l'utilisation de JSX, mais la plupart des gens le trouvent utile comme aide visuelle lorsqu'on travaille avec une interface utilisateur dans le code JavaScript. Cela permet également à React d'afficher des messages d'erreur et d'avertissement plus utiles.

L'infrastructure cloud de joodpefumes

4.1 Introduction

Dans ce chapitre, je présenterai le Workflow d'infrastructure et d'exploitation de JoodPerfumes, la première section sera consacrée à une introduction sur le cloud computing. Dans la deuxième partie j'expliquerai les technologies utilisées pour la gérer et l'organisation des réseaux, services, clusters, aux fournisseurs de cloud que nous utilisons et à la sécurisation de l'infrastructure de Joodperfumes. Enfin, je traiterai de la dernière section l'architecture cloud de Joodperfumes .

4.2 Qu'est-ce que le cloud computing

Pour simplifier, le **cloud** computing est la fourniture de services informatiques (notamment des serveurs, du stockage, des bases de données, la gestion réseau, des logiciels, des outils d'analyse, l'intelligence artificielle) via Internet le **cloud** dans le but d'offrir une innovation plus rapide, des ressources flexibles et des économies d'échelle. En règle générale, vous payez uniquement les services cloud que vous utilisez (réduisant ainsi vos coûts d'exploitation), gérez votre infrastructure plus efficacement et adaptez l'échelle des services en fonction des besoins de votre entreprise.

4.2.1 Principaux avantages du cloud computing

Le cloud computing est radicalement différent de l'approche traditionnelle que les entreprises adoptent en matière de ressources informatiques. Voici sept raisons courantes pour lesquelles les organisations optent pour des services de cloud computing :

- **Coût** : Le cloud computing élimine la nécessité d'investir dans du matériel et des logiciels, et de configurer et de gérer des centres de données sur site : racks de serveurs, alimentation électrique permanente pour l'alimentation et le refroidissement, experts informatiques pour la gestion de l'infrastructure. La facture est vite salée.
- **Vitesse** : La plupart des services de cloud computing sont fournis en libre-service et à la demande. D'énormes ressources de calcul peuvent donc être mises en œuvre en quelques minutes et en quelques clics, offrant ainsi aux entreprises un haut niveau de flexibilité et les dégageant de la pression liée à la planification de la capacité.
- **Mise à l'échelle mondiale** : La mise à l'échelle élastique est un des avantages des services de cloud computing. En termes de cloud, cela veut dire qu'il est possible de mettre en œuvre

la quantité nécessaire de ressources informatiques, par exemple plus ou moins de puissance de calcul, de stockage ou de bande passante, au moment où elles sont nécessaires, là où elles sont nécessaires.

- **Productivité** : Les centres de données sur site nécessitent en général la manipulation de matériel, la mise à jour des logiciels et d'autres corvées informatiques qui prennent beaucoup de temps. Le cloud computing supprime la plupart de ces tâches et les équipes informatiques peuvent donc passer plus de temps à travailler à la concrétisation des objectifs de l'entreprise.
- **Performances** : Les plus grands services de cloud computing s'exécutent sur un réseau de centres de données sécurisés, dont le matériel est régulièrement mis à niveau pour assurer des performances rapides et efficaces. Ceci offre plusieurs avantages par rapport à un centre de données classique, y compris un temps de latence réseau réduit pour les applications et de plus grandes économies d'échelle.
- **Fiabilité** : Le cloud computing simplifie la sauvegarde des données, la récupération d'urgence et la continuité des activités. Il rend ces activités moins coûteuses, car les données peuvent être mises en miroir sur plusieurs sites redondants au sein du réseau du fournisseur.
- **Sécurité** : De nombreux fournisseurs de cloud offrent un vaste éventail de stratégies, technologies et contrôles qui renforcent globalement votre situation de sécurité, contribuant ainsi à protéger vos données, vos applications et votre infrastructure contre des menaces potentielles.

4.2.2 Types de cloud computing

Tous les clouds ne sont pas identiques et aucun type de cloud computing ne convient à tout le monde. Plusieurs modèles, types et services différents ont évolué pour vous aider à trouver la solution adaptée à vos besoins.

Vous devez commencer par déterminer le type de déploiement cloud ou d'architecture de cloud computing sur lequel vos services cloud seront implémentés. Il existe trois modes de déploiement de services cloud : le cloud public, le cloud privé et le cloud hybride.

Cloud public : Public clouds are owned and operated by a third-party cloud service providers, which deliver their computing resources, like servers and storage, over the Internet. Microsoft Azure is an example of a public cloud. With a public cloud, all hardware, software, and other supporting infrastructure is owned and managed by the cloud provider. You access these services and manage your account using a web browser.

Cloud privé : Le cloud privé est l'ensemble des ressources de cloud computing utilisées de façon exclusive par une entreprise ou une organisation. Le cloud privé peut se trouver physiquement dans le centre de données local de l'entreprise. Certaines entreprises paient également des fournisseurs de services pour qu'ils hébergent leur cloud privé. Le cloud privé est un cloud dans lequel les services et l'infrastructure se trouvent sur un réseau privé.

Cloud hybride : Le cloud hybride regroupe des clouds publics et privés, liés par une technologie leur permettant de partager des données et des applications. En permettant que les données et applications se déplacent entre des clouds privé et public, un cloud hybride offre à votre entreprise une plus grande flexibilité, davantage d'options de déploiement et une optimisation de votre infrastructure, de votre sécurité et de votre conformité existantes.

4.2.3 Types de services cloud : IaaS, PaaS, SaaS

La plupart des services de cloud computing peuvent être classés en quatre grandes catégories : IaaS (infrastructure as a service), PaaS (platform as a service), serverless et SaaS (software as a service). On les appelle parfois « pile » de cloud computing, car elles s'empilent les unes sur les autres. Si vous

savez en quoi elles consistent et en quoi elles sont différentes, vous pourrez plus facilement atteindre vos objectifs.

Infrastructure as a service (IaaS) : La catégorie la plus basique des services de cloud computing. Avec l'IaaS, vous louez une infrastructure informatique (serveurs, machines virtuelles, stockage, réseaux, systèmes d'exploitation) auprès d'un fournisseur de services cloud, avec un paiement à l'utilisation

Platform as a service (PaaS) : L'expression plateforme en tant que service (PaaS, Platform-as-a-Service) qualifie les services de cloud computing qui offrent un environnement à la demande pour développer, tester, fournir et gérer des applications logicielles. PaaS est conçu pour permettre aux développeurs de créer rapidement des applications web ou mobiles sans avoir à se préoccuper de la configuration ou de la gestion de l'infrastructure de serveurs, de stockage, de réseau et de bases de données nécessaire au développement.

Software as a service (SaaS) : Le logiciel en tant que service (SaaS, Software-as-a-Service) est une méthode de diffusion d'applications logicielles via Internet, à la demande et en général sur abonnement. Avec le SaaS, les fournisseurs de services cloud hébergent et gèrent les applications logicielles et l'infrastructure sous-jacente, et gèrent la maintenance, par exemple la mise à niveau des logiciels et l'application des correctifs de sécurité. Les utilisateurs se connectent à l'application via Internet, en général par l'intermédiaire d'un navigateur web sur leur téléphone, leur tablette ou leur PC.

4.3 le choix de AWS

Amazon Web Services (AWS) et Google Cloud sont des plates-formes pour fournir des services de cloud computing aux entreprises et aux particuliers. Nous avons choisi AWS parce qu'il est plus efficace que Google Cloud dans plusieurs services que nous allons maintenant mentionner :

La première catégorie concerne la façon dont leurs machines virtuelles (instances) sont traitées dans la technologie Google Cloud Virtual Machine est KVM, tandis que la technologie derrière les machines virtuelles AWS EC2 est Xen.

Stockage / Disque : le type de stockage et les disques utilisés par un fournisseur de cloud jouent un rôle très important car ils ont un impact direct sur les performances, tels que le débit attendu (IO), le nombre maximal d'E / S par volume / instance et possibilité d'augmenter la capacité sur de courtes périodes. Lorsque vous comparez Google et AWS, il existe deux principaux types d'options de stockage : le stockage en masse et le stockage d'objets.

La capacité réseau disponible sur les instances de Google Compute Engine fonctionne légèrement différemment, car elle dépend du nombre de processeurs de vos machines virtuelles. Chaque noyau est soumis à une limite de 2 Gbits / seconde (Gbps) pour des performances optimales. Chaque noyau supplémentaire augmente le plafond du réseau, jusqu'à un maximum théorique de 16 Gbps pour chaque machine virtuelle.

Les instances Amazon EC2 ont une bande passante maximale de 25 Gbps, mais cela ne s'applique qu'aux instances plus grandes. Nombre maximal d'instances standard à 10 Gbps / seconde.

la figure suivante montre les Tendances du Cloud Computing sur Stack Overflow dans lesquels ils analysent des modèles basés sur le % de questions d'un mois.

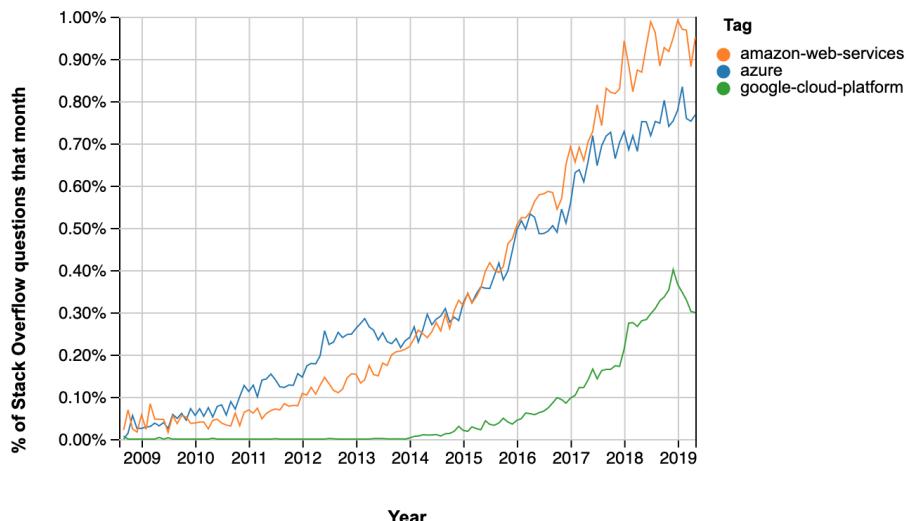


FIGURE 4.1: Tendances du Cloud Computing sur Stack Overflow

Nous avons aussi opté pour AWS car dans l'entreprise ils ont déjà travaillé avec google cloud mais malheureusement ils ont eu beaucoup de problèmes soit avec les serveurs qui tombent en panne soit avec la performance qui est faible

4.4 Infrastructure en tant que code

Parfois appelée "infrastructure programmable," l'infrastructure en tant que code **IaC** aborde la configuration d'infrastructure exactement de la même manière que la programmation de logiciel. En effet, elle commence par estomper les limites entre la rédaction d'applications et la création des environnements où celles-ci sont exécutées. Les applications peuvent contenir des scripts qui créent et orchestrent leur propres machines virtuelles (VM). C'est une partie fondamentale du **Cloud Computing** et essentielle aux **DevOps**.

L'infrastructure en tant que code permet une gestion programmable de machines virtuelles, ce qui élimine le besoin en configuration manuelle et en mises à jour des composantes matérielles. Cela rend cette infrastructure hautement "élastique", c'est-à-dire reproductible et évolutive. Un opérateur peut déployer et gérer 1 machine ou 1 000 machines avec le même ensemble de codes. La vitesse, l'économie de coûts et la réduction des risques sont les conséquences directes de l'infrastructure en tant que code.

L'infrastructure en tant que code supporte **IaaS** en permettant l'accès à des machines virtuelles et à des outils de gestion s'appuyant sur des logiciels conviviaux, regroupés et vendus au sein d'un service.

Actuellement, toutes les ressources d'infrastructure sont gérées à l'aide de terraform, le code est stocké dans un référentiel GitHub.

4.5 Terraform - Concepts

Dans cette section, nous allons présenter les principaux concepts qui composent Terraform. Les fichiers de configuration s'écrivent en HCL (HashiCorp Configuration Language). Le principe est d'écrire des ressources.

Les ressources peuvent être écrites en JSON également, mais il est recommandé de les écrire en HCL.

La figure 4.2 représente le composants d'une ressource avec terraform :



FIGURE 4.2: HashiCorp Configuration Language

L'exemple ci-dessous montre que, en quelques lignes de code, on va pouvoir déployer/créer un bucket S3 sur AWS :

```
resource "aws_s3_bucket" "default" {
  bucket = "jood-terraform"
  acl    = "private"

  versioning {
    enabled = true
  }
}
```

FIGURE 4.3: Création de S3 bucket avec terraform

La figure 4.4 représente le Résultat Terraform code sur AWS :

The screenshot shows the AWS S3 console interface. The search bar contains "terr". There is one bucket listed:

Bucket name	Access	Region
jood-terraform	Objects can be public	US East (N. Virginia)

FIGURE 4.4: Résultat sur AWS

4.5.1 Terraform - Fournisseur (Provider)

Un fournisseur (provider) est responsable du cycle de vie/du CRUD (create, read, update, delete) d'une ressource : sa création, sa lecture, sa mise à jour et sa suppression.

Plus de 125 fournisseurs (providers) sont actuellement supportés :

- AWS, GCP, Azure, OpenStack... ;
- Heroku, OVH, 1&1... ;
- Consul, Chef, Vault... ;
- Consul, Chef, Vault... ;

- GitLab, BitBucket, GitHub... ;
- MySQL, PostgreSQL... ;
- mais encore RabbitMQ, DNSimple, CloudFlare... .

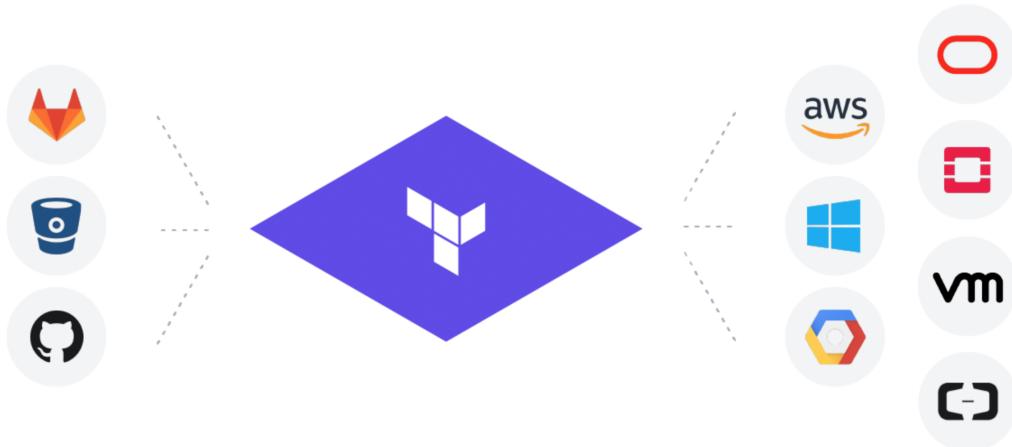


FIGURE 4.5: Exemples de fournisseurs Terraform

4.5.2 Terraform - Variables

Afin de faire du code propre et qui se réutilise, il est recommandé d'initialiser des variables et de les utiliser dans les autres fichiers .tf.

```
variable "domain" {
  default = "joodperfumes.com"
}
variable "email" {}
variable "token" {}
```

FIGURE 4.6: Variables avec Terraform

4.5.3 Terraform - Modules

Les modules sont utilisés pour créer des composants réutilisables, améliorer l'organisation et traiter les éléments de l'infrastructure comme une boîte noire.

C'est un groupe de ressources qui prennent en entrée des paramètres et retournent en sortie des outputs. Dans le même fichier, fichier.tf, dans lequel vous avez défini votre fournisseur, vous pouvez ensuite définir vos modules :

```

module "vpn" {
    source = "../../modules/aws/ec2/vpn"

    prefix      = "${local.prefix}"
    common_tags = "${local.common_tags}"
    vpn_domain  = "aws-vpn.jood.com"

    s3_logging      = "${var.s3_logging}"
    private_subnet_ids = "${var.private_subnet_ids}"
    public_subnet_ids = "${var.public_subnet_ids}"
    vpc_id          = "${var.vpc_id}"
    cidr_block      = "${var.cidr_block}"
}

```

FIGURE 4.7: Modules avec Terraform

4.5.4 Terraform - Outputs

Les modules peuvent produire des outputs que l'on pourra utiliser dans d'autres ressources.

```

output "alb_dns" {
    value = "${module.rest_api.dns_name}"
}

# ecs
output "ecs" {
    value = {
        name = "${aws_ecs_cluster.jood_ecs.name}"
        arn  = "${aws_ecs_cluster.jood_ecs.arn}"
    }
}

```

FIGURE 4.8: Outputs avec Terraform

La figure 4.11 représente l'utilisation d'une output dans une autre ressource avec terraform :

```

resource "cloudflare_record" "qa-api" {
    domain  = "${var.domain}"
    name    = "qa-api.${var.domain}"
    type    = "CNAME"
    value   = "${data.terraform_remote_state.qa_state.alb_dns}"
    proxied = true
}

```

FIGURE 4.9: Appel d'une Output avec Terraform

4.5.5 Terraform - Sources de données (Data Sources)

Les sources de données (Data Sources) servent à récupérer une donnée/ressource existante dans votre infrastructure.

Utiliser une source de données est une bonne pratique. En effet, il est bien de privilégier son utilisation plutôt que de coder en dur des variables ou autre donnée qui peut être récupérée !

```

data "aws_acm_certificate" "ssl_certificate" {
  domain = "*..com"
}

```

FIGURE 4.10: Data sources avec Terraform

Et ensuite vous pouvez appeler cette source de données simplement dans vos fichiers .tf comme ceci :

```

resource "aws_lb_listener" "ssl_listener" {
  load_balancer_arn = "${aws_alb.default.id}"
  port              = "443"
  protocol          = "HTTPS"
  ssl_policy        = "ELBSecurityPolicy-2015-05"
  certificate_arn   = "${data.aws_acm_certificate.ssl_certificate.arn}"
}

```

FIGURE 4.11: Appel d'une source de données avec Terraform

4.5.6 Terraform - Etat (State)

Un état (state) est un snapshot de votre infrastructure depuis la dernière fois que vous avez exécuté la commande `terraform apply`.

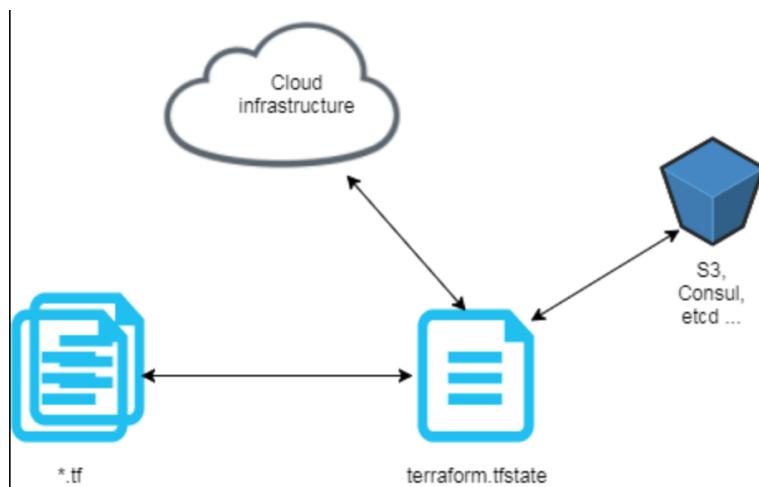


FIGURE 4.12: état (state) avec Terraform

Terraform utilise un stockage local pour créer les plans et effectuer les changements sur votre infrastructure, mais il est possible de stocker cet état dans une solution cloud.

Pour configurer le stockage de cet état de manière distante, il suffit de définir un backend.

```

  terraform {
    backend "s3" {
      bucket      = "jood-terraform"
      key         = "aws/us-east1/qa/terraform.tfstate"
      region      = "us-east-1"
      shared_credentials_file = "~/.aws/credentials"
      profile     = "jood"
    }

    required_version = "= 0.11.13"
  }

```

FIGURE 4.13: le stockage d'un etat avec Terraform

4.5.7 Terraform - CLI

Terraform a une CLI (Command-Line Interface) facile à utiliser et composée de plusieurs commandes : nous allons en voir quelques-unes, pas toutes, uniquement celles que j'utilise le plus au quotidien.

- **terraform init** : La commande init va initialiser votre répertoire de travail qui contient vos fichiers de configuration au format .tf.
C'est la première commande à exécuter pour une nouvelle configuration, ou après avoir fait un checkout d'une configuration existante, depuis votre dépôt git par exemple.
La commande init va :
 - télécharger et installer les fournisseurs (providers) ;
 - initialiser le backend (si défini) ;
 - télécharger et installer les modules (si définis).
- **terraform plan** : La commande plan permet de créer un plan d'exécution. Terraform va déterminer quelles actions il doit entreprendre afin d'avoir les ressources listées dans les fichiers de configuration par rapport à ce qui est actuellement en place sur l'environnement/le fournisseur cible.
Cette commande n'effectue concrètement rien sur votre infrastructure.
- **terraform apply** : La commande apply, comme son nom l'indique, permet d'appliquer les changements à effectuer sur l'infrastructure. C'est cette commande qui va créer nos ressources.

4.6 Infrastructure de Jood

L'infrastructure Joodperfumes.com (ou Jood) est entièrement gérée par Dockers. Chaque composant est censé fonctionner dans un environnement Docker et géré par AWS ECS Fargate . L'infrastructure est actuellement hébergée sur Aamzone web service (AWS) alors que le domaine DNS utilise Cloudflare. Les détails sur l'automatisation et les configurations d'infrastructure sont suivis dans Jood-infra, fonctionnant actuellement dans une zone unique avec la configuration en place pour s'étendre à d'autres zones et / ou fournisseurs de cloud

4.6.1 Jood-Services de calcul et de mise en réseau

- **Amazon VPC** : Amazon Virtual Private Cloud nous permet de lancer et ressembler les ressources de Jood sur un réseau virtuel sur l'infrastructure du jood on a deux réseaux :
 - admin vpc : Ce réseau virtuel rassemble les composants communs entre la zone de production et la zone de staging tel que bastion host et le vpn
 - staging vpc : Ce réseau contient les principaux éléments de jood (Docker conteneurs , Instances EC2) ainsi que les outils pour les gerer.

Réseaux	Cidr proposé
qa-jood-useast1-vpc	10.241.0.0/16
adm-jood-useast1-vpc	10.241.0.0/16

TABLE 4.1: Les réseaux de l'infrastructure Jood

- **Amazon EC2** : Amazon Elastic Compute Cloud est un service Web qui fournit une capacité de calcul sécurisée et redimensionnable dans le cloud. il est conçu pour faciliter l'accès aux ressources de cloud computing à l'échelle du Web les instances ec2 dans joodperfumes

Instances	Role	Réseau
Odoo	Odoo est le système ERP utilisé avec notre boutique	adm-jood-useast1-vpc
RabbitMq	RabbitMQ est le courtier de messages utilisé par Jood.	qa-jood-useast1-vpc
VPN	Créer des connexions sécurisées avec les réseaux adm et qa (staging) basées sur le cryptage du protocole OpenVPN	adm-jood-useast1-vpc
Bastion	Instance qui fournit un point d'entrée et/ou de sortie unique vers toutes les instances de jood infra par ssh	qa-jood-useast1-vpc

TABLE 4.2: Les Instances EC2 de l'infrastructure Jood

- **Amazon ECS** : Amazon Elastic Container Service est un service de gestion de conteneurs hautement évolutif et rapide, qui facilite l'exécution, l'arrêt et la gestion des conteneurs Docker sur un cluster d'instances Amazon EC2. Joodperfumes utilise ECS Fargate qui est un moteur de calcul conçu pour Amazon ECS qui nous permet d'exécuter les conteneurs sans avoir à gérer des serveurs ou clusters.

Conteneur	Role	Réseau
Jood-Core	API de joodperfumes	qa-jood-useast1-vpc
Celery	Celery est une file d'attente de tâches / travaux basée sur la transmission de messages distribuée exécutées en arrière-plan	qa-jood-useast1-vpc
Flower	Flower est un outil Web pour surveiller et administrer les Clusters de Celery.	qa-jood-useast1-vpc

TABLE 4.3: Les Conteneurs Docker de l'infrastructure Jood

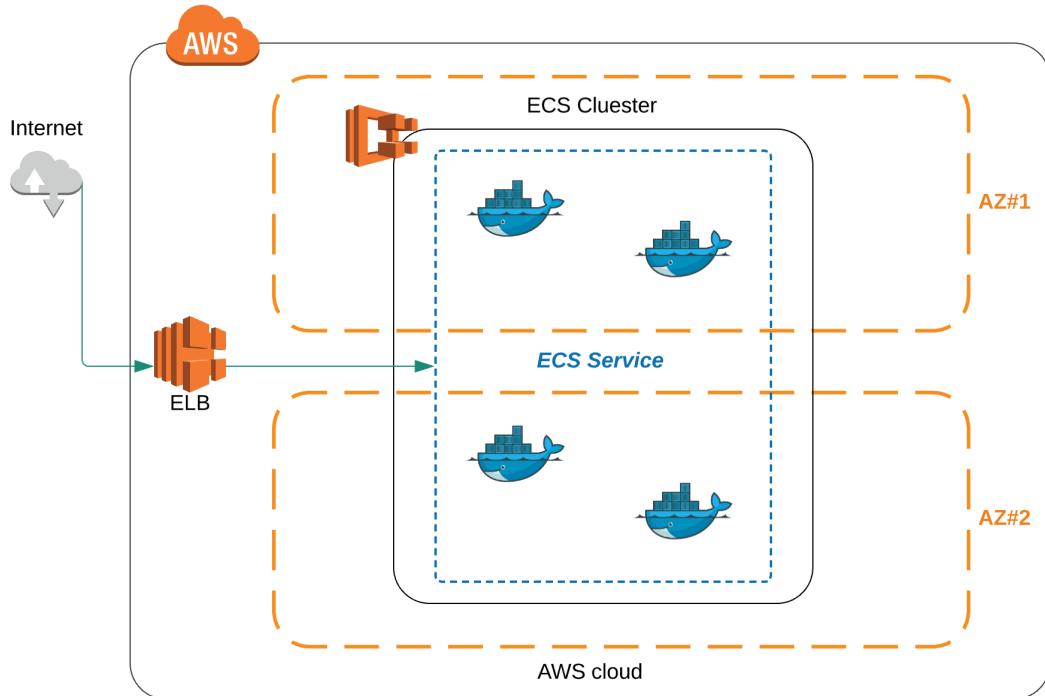


FIGURE 4.14: la gestion de conteneurs avec AWS Fargate

- **Elastic Load Balancing** : (ELB) répartit automatiquement le trafic entrant sur nos instances Ec2 (rabbitmq) et les conteneurs (flower,jood-core, Celery) dans la zone de staging. Elb sert aussi a détecter nos cibles non saines et cesser d'y acheminer du trafic et répartir la charge sur les autres cibles saines.

ELB	Type
adm-jood-useast1-bastion-elb	privée
qa-jood-useast1-core-api-elb	public
qa-jood-useast1-flower-elb	privée
qa-jood-useast1-rabbitmq-elb	privée

TABLE 4.4: Jood-Elb

- **Amazon ECR** : Amazon Elastic Container Registry est un registre de conteneurs Docker entièrement géré par AWS , il permet de stocker, gérer et déployer facilement nos images Docker (conteneurs) a partir de github repository. c'est une version publiable du code source sur aws .

4.6.2 Jood-Services de stockage et de livraison de contenu

- **Amazon Relational Database Service** : Le service de base de données Amazon Relational (Amazon RDS) est un service Web qui facilite la configuration, l'utilisation et la mise à l'échelle d'une base de données relationnelle dans le cloud. Jood infra contient 2 instances RDS l'une pour gerer la base de données de plateforme et l'autre pour la base de données de l'ERP Odoo

Base de données	Type
qa-jood-useast1-core	PostgreSQL
qa-jood-useast1-odoo	PostgreSQL

TABLE 4.5: Les bases de données RDS du Jood

— **Amazon S3** : Amazon Simple Storage Service est un service de stockage dans le cloud sur internet. Vous pouvez utiliser Amazon S3 pour stocker et récupérer toute quantité de données à tout moment, où que vous soyez sur le Web. Le service s3 utilisé dans jood infra pour :

- Charger des objets (photos, vidéos de produits) ;
- Stocker des fichiers statiques (html, css, js) ;
- Le dernier état (snapshot) de l'infrastructure ; (instances, vpc, sécurité ...)
- Charger les (logs, settings, cache) ...

— **Amazon CloudFront** : Amazon CloudFront accélère la distribution du contenu Web statique et dynamique, tel que les fichiers .html, .css, .php, images et fichiers multimédias.

Lorsque les utilisateurs demandent le contenu de notre boutique, CloudFront le diffuse via un réseau mondial d'emplacements périphériques offrant une faible latence et des performances élevées.

4.6.3 Jood-Services de sécurité et d'identité

— **AWS Identity and Access Management** : AWS (IAM) est un service Web permettant de contrôler en toute sécurité l'accès aux services AWS. Avec IAM, nous pouvons gérer de manière centralisée les utilisateurs, les informations d'identification de sécurité telles que les clés d'accès et les autorisations qui contrôlent les ressources AWS auxquelles les utilisateurs et les applications peuvent accéder.

— **AWS Security Group** : AWS security group agit en tant que pare-feu virtuel pour que une instance contrôle le trafic entrant et sortant. Lorsque vous lancez une instance dans un VPC, nous pouvons affecter jusqu'à cinq groupes de sécurité à l'instance. Les groupes de sécurité agissent au niveau de l'instance et non du sous-réseau.

La figure 4.15. présente la sg (security group) du service jood-core qui autorise tout le trafic provenant du load balancer.

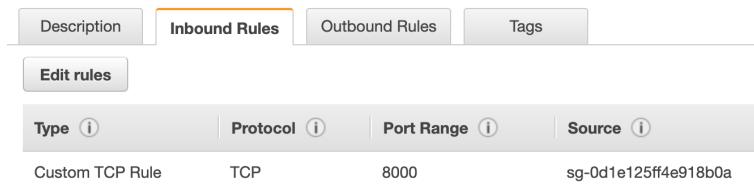


FIGURE 4.15: Security group du service Jood-core

La figure 4.16. présente la sg du load balancer qui autorise tout le trafic provenant d'Internet via le port 80 (http) et 443 (https).

Security Group: sg-0d1e125ff4e918b0a			
Description	Inbound Rules	Outbound Rules	Tags
Edit rules			
Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	8000	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0

FIGURE 4.16: Security groupe du Load Balancer

4.6.4 Jood-Services de Management

- **AWS Auto Scaling** : Avec AWS Auto Scaling, nous pouvons configurer rapidement une mise à l'échelle dynamique et prédictive pour nos ressources AWS évolutives et pour mettre à l'échelle les instances EC2 et les conteneur docker.
- **Amazon CloudWatch** : Amazon CloudWatch fournit une solution de surveillance fiable, évolutive et flexible que nous avons utiliser pour collecter et suivre des métriques, qui sont des variables que nous pouvons mesurer pour nos ressources et nos applications.

4.6.5 Jood-Outils de développement

- **AWS CodeBuild** : AWS CodeBuild est un service qui compile le code source, exécute des tests unitaires et produit des artefacts prêts à être déployés.
- **AWS CodePipeline** : AWS CodePipeline est un service de distribution continue qui nous permet de modéliser, de visualiser et d'automatiser les étapes nécessaires à la publication de notre plateforme.
- **AWS CodeDeploy** : AWS CodeDeploy est un service de déploiement qui nous permet d'automatiser le déploiement sur les instances et de mettre à jour les repositories en fonction des besoins.

Le diagramme 4.17. montre un exemple de processus de publication utilisant CodePipeline.

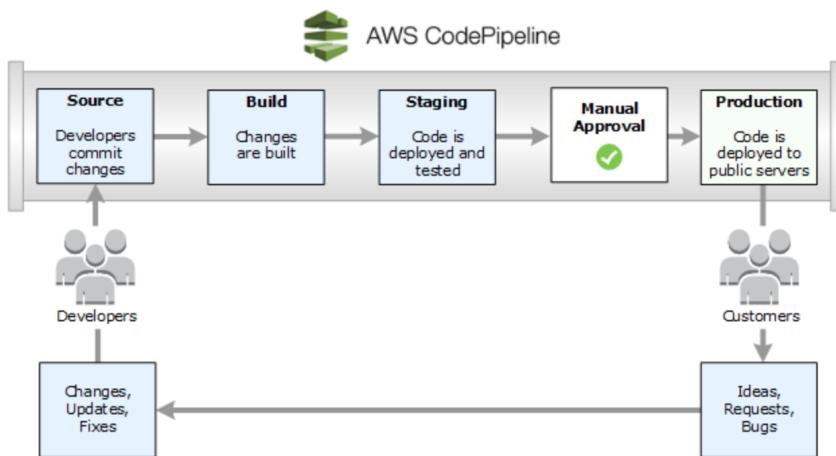


FIGURE 4.17: Processus de publication avec CodePipeline

Cycle de déploiement du Jood Les cycles de déploiement dans Jood-infra sont structurés comme suit :

- Chaque validation des modifications apportées avec commit dans GitHub déclenche un Trigger de terraform par AWS CodePipeline Webhooks ,ce dernière faire appel au CodeBuild pour la compilation du code source et l'exécution des tests unitaire .
- Chaque fusion à la branche master déclenchera un flux de travail avec 2 plans de travail, s'appliquera à l'environement d'administration , S'appliquera aussi a la production .

La figure 4.18 représente le Cycle de déploiement du Jood :

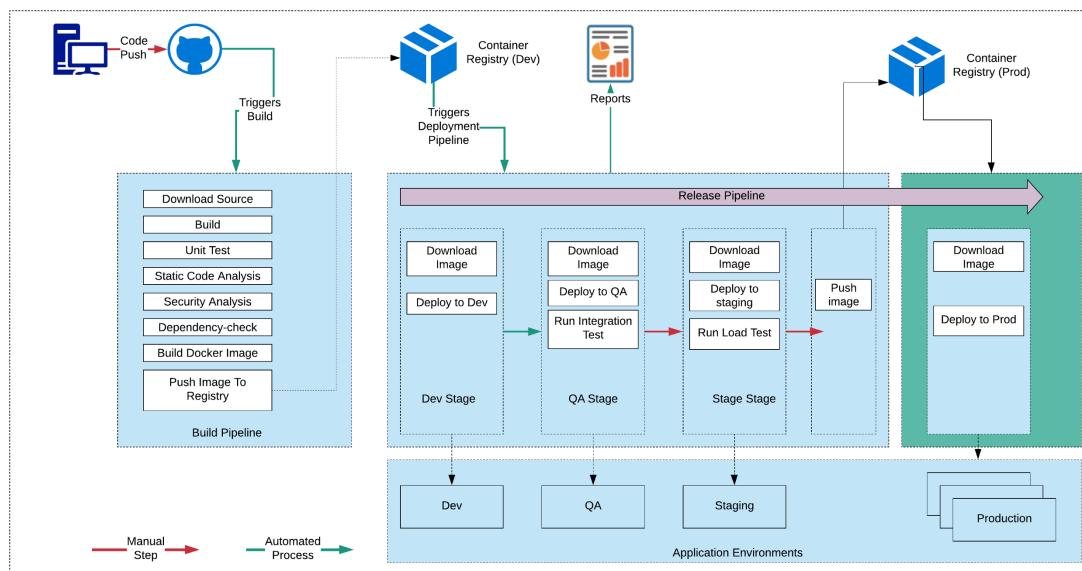


FIGURE 4.18: Intégration-Déploiement Continue de Jood

4.6.6 La sécurisation de l'infrastructiure de Jood

Problématique Toute personne sur Internet semble recommander d'exécuter vos instances EC2 dans une boîte aux lettres fermée, d'installer une instance NAT puis de configurer une DMZ avec au moins un serveur bastion. Brisez la vie privée et capturez les sessions des utilisateurs avec des scripts faits maison !

Solution proposé Il est pratique courante d'utiliser les boîtes de saut SSH et le groupe de sécurité pour restreindre l'accès SSH aux instances dans AWS. Cette méthode fonctionne très bien,

parmi les choix sur aws est le bastion hôte (instance)

Un hôte bastion est un serveur dont le but est de fournir un accès à un réseau privé à partir d'un réseau externe, tel qu'Internet. En raison de son exposition à une attaque potentielle, un hôte de bastion doit minimiser le risque de pénétration. Par exemple, vous pouvez utiliser un hôte bastion pour atténuer le risque d'autoriser les connexions SSH d'un réseau externe à des instances Linux lancées dans vos réseaux privés depuis votre vpc.

L'architecture de la solution Dans cette section, je présente l'architecture de cette solution sur notre projet et explique comment nous avons configuré l'hôte bastion pour enregistrer les sessions SSH. Comme nous l'avons déjà expliqué dans ce chapitre, Amazon VPC vous permet de lancer des ressources AWS sur un réseau privé virtuel que vous avez défini. L'hôte du bastion fonctionne sur une instance Amazon EC2 qui est généralement située dans un sous-réseau public de votre commande par correspondance Amazon. Les instances Linux sont situées dans un sous-réseau qui n'est pas accessible

au public, et elles sont configurées avec un groupe de sécurité qui permet l'accès SSH depuis le groupe de sécurité attaché à l'instance EC2 sous-jacente qui exécute l'hôte bastion. Les utilisateurs de l'hôte Bastion se connectent à l'hôte Bastion pour se connecter aux instances Linux, comme le montre le schéma suivant 4.19

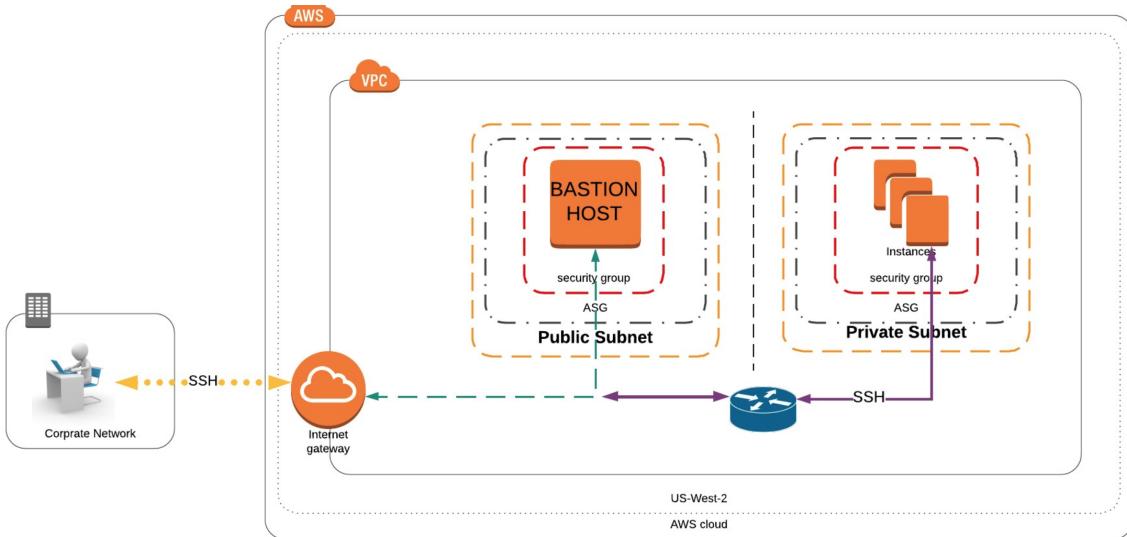


FIGURE 4.19: Sessions SSH établies par l'intermédiaire d'un hôte Bastion

Bastion host dans Jood infra est une instance qui se trouve dans l'admin vpc. Une fois que la connectivité à distance a été établie avec l'hôte bastion, il agit alors comme un serveur 'jump', vous permettant d'utiliser SSH pour vous connecter à d'autres instances (dans des sous-réseaux privés/publics) plus profondes dans notre adm et qa VPC qui sont liées avec une connexion de peering VPC entre les deux. .le bastion sert essentiellement de pont vers d'autres instances du réseau. Toutes les instances ont le port 22 restreint pour être ouvert uniquement à l'instance du bastion.

Afin d'améliorer cette solution, il est parfois très utile de pouvoir accéder aux instances directement et de manière sécurisée. Pour ce faire, nous avons utilisé OpenVPN combiné avec quelques astuces AWS pour créer un serveur bastion VPN résilient. et pour mettre tous les services (conteneurs docker, instances...) derrière le vpn

Qu'est-ce qu'un VPN ? Pensez à un réseau privé virtuel comme une couche sous votre connexion réseau. En connectant votre appareil au VPN, les données sont envoyées et reçues en privé via une adresse IP différente. La connexion à un VPN offre une multitude d'avantages, tels que la fonctionnalité, la sécurité (cryptage des données). En termes très simples, un hôte VPN est le seul hôte ou ordinateur autorisé à accéder au réseau public. Il s'agit généralement d'une puissante boîte de serveur avec une sécurité réseau élevée parce que c'est le seul hôte qui est autorisé l'accès public (donc il dirige notre instance bastion vers un réseau privé) Les administrateurs système peuvent utiliser cette machine pour se connecter au réseau jood dans le cloud

Cela signifie que vous ne pouvez pas ssh aux instances du Jood sans passer par l'hôte bastion et vous ne pouvez pas y accéder si nous sommes pas connecté à l'hôte VPN. comme le montre le schéma suivant : figure 4.20

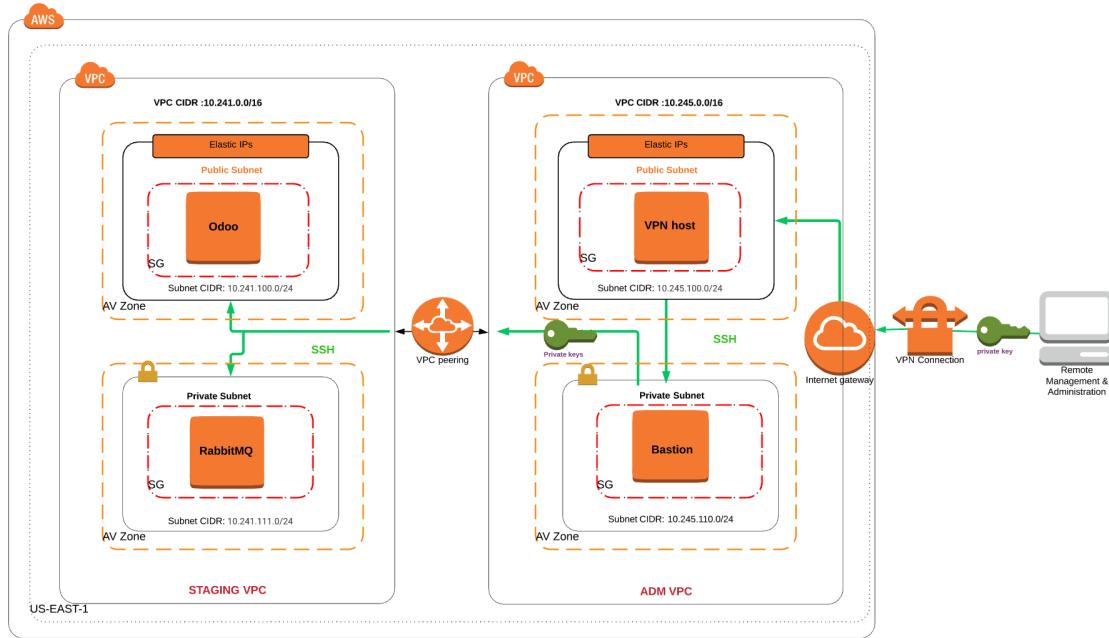


FIGURE 4.20: La configuration des hôtes VPN et Bastion dans jood

4.6.7 L'architecture global

La figure 4.21 représente l'architecture cloud du Jood sur AWS :

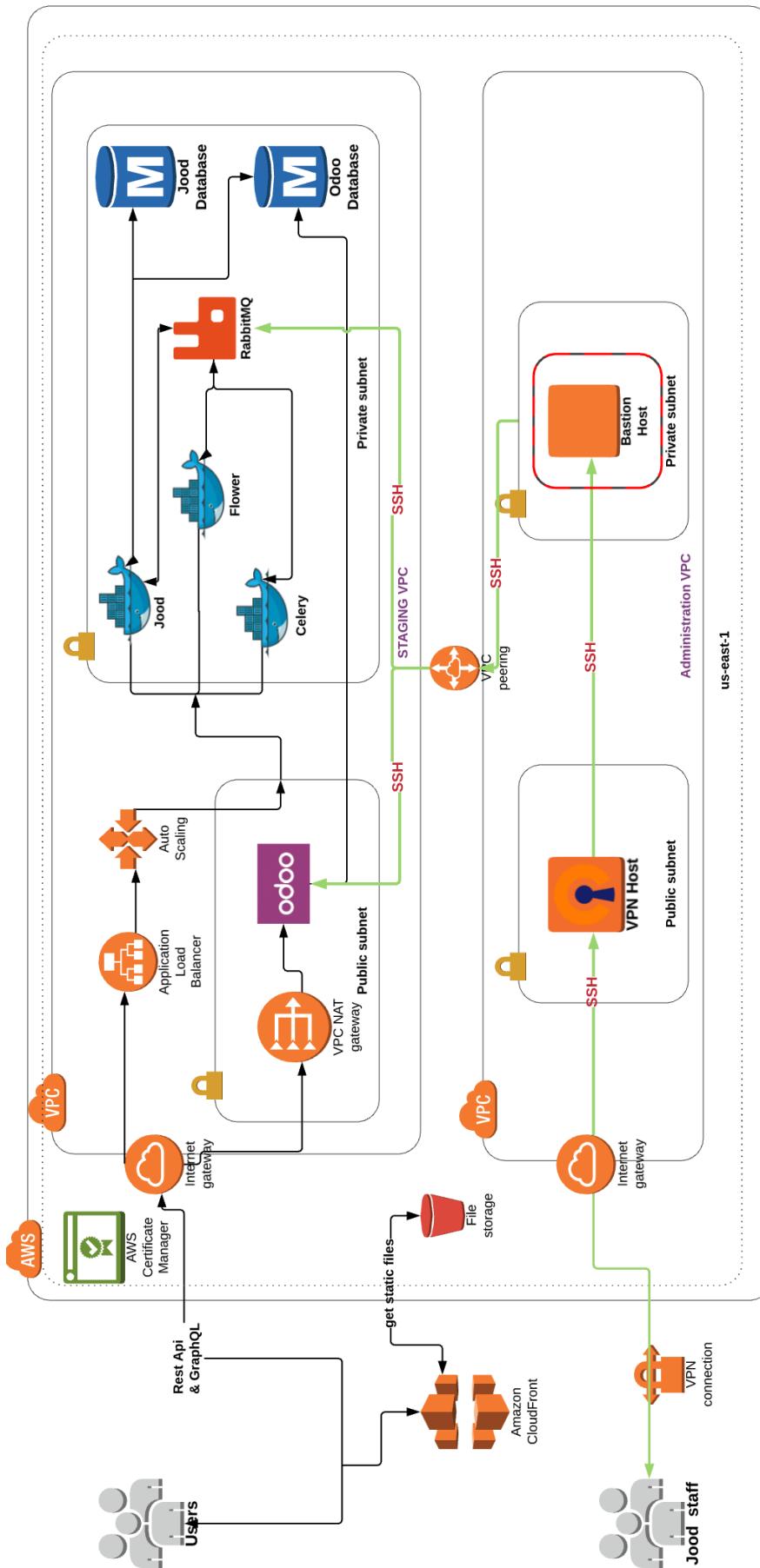


FIGURE 4.21: L'architecture cloud du Jood sur AWS

4.7 Conclusion

Durant ce chapitre, nous avons traité la conception et l'implémentation de l'environnement Cloud pour les différents services du Joodperfumes

Jood - développement backend

les référentiels du Joodperfumes sont séparés en 3 catégories : référentiels de micro-modules backend, référentiels d'opérations et référentiels d'interface utilisateur. Dans ce chapitre, nous expliquerons l'architecture backend du projet Joodperfumes. D'abord nous commencerons par détailler la structure des repertoire sous Jood-core ,l'utilisation des différents fichiers ensuite nous présenterons quelques outils utilisés pour l'administration de l'application JOOD-ODOO et leur objectif ainsi que quelques captures de notre projet.

5.1 Architecture backend du Jood

JOOD a différentes applications, chacune ayant une responsabilité unique mais formant ensemble une seule et même plateforme solide et cohérente.

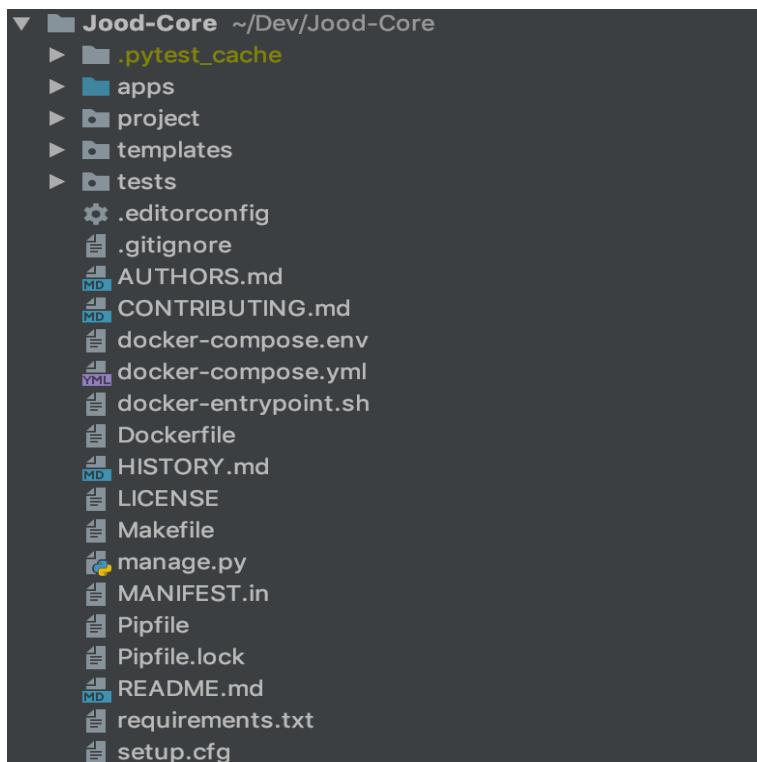


FIGURE 5.1: Arhitecture du Jood-Core

5.1.1 Répertoire Pytest_cache

pytest-cache fait deux choses :

- un mécanisme par lequel d'autres plugins peuvent obtenir / définir des valeurs via config.cache.get|set. Ceci est utilisé par pytest-pep8 et pytest-flakes par exemple pour stocker l'heure de la dernière vérification afin d'éviter de revérifier les fichiers tout le temps.
- sauvegarder les échecs de test afin que vous puissiez exécuter -lf pour n'exécuter que les derniers échecs et -ff pour exécuter les derniers échecs en premier, puis le reste des tests

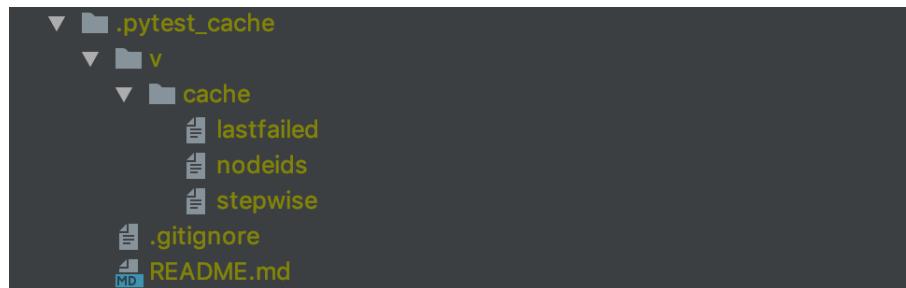


FIGURE 5.2: Le Répertoire du Pytest_cahe

5.1.2 Répertoire Project

- Celery.py contient la configuration du service CELERY TASK
- S3_storage.py fichier de configuration pour le système de stockage S3
- Production.py contient la configuration spécifiée en production (paramètres spécifiques à l'environnement, caches)
- Local.py contient la configuration dédié à l'environnement local (Réglages spécifiques à l'environnement, caches)
- Views.pyc'est fichier des contrôleurs de l'application, Il contient la fonction qui nous retourne la page Index.html
- Urls.py c'est le fichier des routes , le lien entre les contrôleurs (views) et les vues (templates)
- Wsgi.py spécifie Comment déployer avec WSGI (Web Server Gateway Interface)

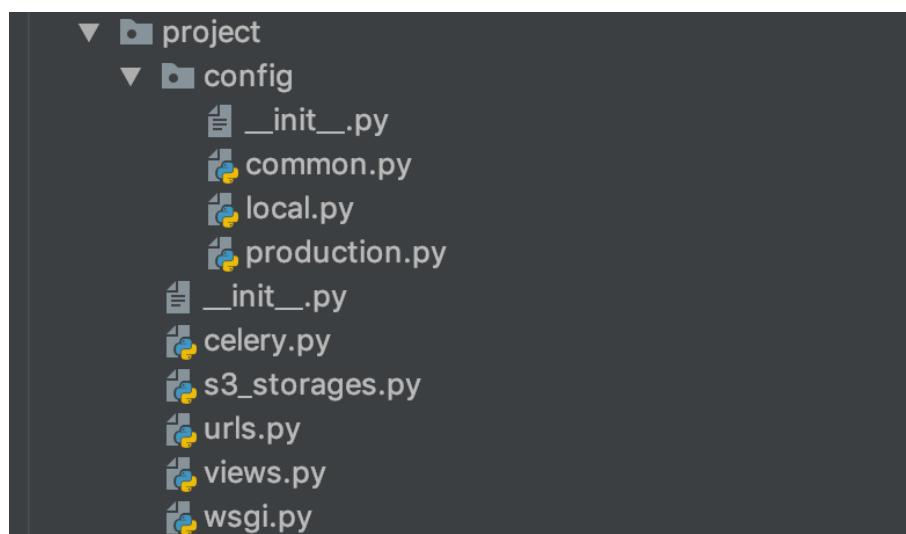


FIGURE 5.3: le répertoire d'arborescence du projet

5.1.3 Répertoire Templates

C'est le répertoire des Vues de backend, il contient seulement index.html page de démarrage de l'application vue que les templates et leurs traitement se présentent dans Jood-Web

5.1.4 Répertoire apps

Il contient les répertoires de l'objectif principale du projet, distribués aux modules suivants appelés applications. Le terme application décrit un paquet Python qui fournit un certain ensemble de fonctionnalités. Les applications peuvent être réutilisées dans différents projets



FIGURE 5.4: Le répertoire de templates

- **Cart** C'est le module qui gère les paniers et ces articles, contient les classes Cart, CartItemet et les services Web graphql
- **Profile** c'est le module qui contient les classes User, et Site, il a la même forme de Django-profile, il contient les contrôleurs d'enregistrement et d'authentification et les services Web Graphql
Généralement, un utilisateur peut s'authentifier avec un email / mot de passe
Pour enregistrer un utilisateur avec un email et un mot de passe, nous faisons un POST pour / utilisateurs / en incluant ces informations : Email, mot de passe, prénom, nom, numéro de téléphone
- **Product** C'est le module qui gère le stock et la catégorie de produits, ainsi que les dépôts de ces produits, et fournit aux utilisateurs toutes les informations sur les produits.
c'est le module qui contient les classes Product, Site, Category, Warehouse, GiftWrapping et GiftWrappingSite, ainsi que les services Web graphql.
- **Order** Ce module gère le cycle de vie d'une commande, de la création à la facturation.
le module qui contient les classes Order, OrderItem et Shipping Address et les services Web Graphql, ainsi qu'un service Web Rest pour initialiser le paiement d'une commande.
Il gère également la facturation après le parment d'une commande
- **graphql_core** c'est le répertoire qui contient la configuration générale de graphql ainsi que les fonctions utilitaires de graphql (fonction de décodage et d'encodage, ...) ainsi que les classes mères de mutations ...
- **graphql_utils** contient le schéma de toutes les requêtes graphql dans l'application
Il s'agit d'un contrat conclu entre e front-end et le back-end. Par conséquent, son maintien au centre permet aux deux parties du développement d'évoluer sans sortir du cahier des charges. Cela facilite également la parallélisation du travail, car le client peut dès le départ avoir une connaissance complète de l'API, à l'aide d'un simple service de simulation, qui peut ensuite être facilement remplacé par ce dernier. serveur.
- **Notify** c'est le module qui gère les notifications envoyé aux utilisateurs après un paiement d'un ordre ou d'une inscription au site et aussi les promotions
Le service devrait :
 - Autoriser l'utilisateur à définir ses préférences de notification.
 - Autoriser l'utilisateur à définir sa méthode de notification préférée
 - Fournir une interface configurable aux modèles de courrier électronique et aux fournisseurs.
 - Gérez le fil de nouvelles des utilisateurs en fonction des canaux de notification et des canaux pub / subs

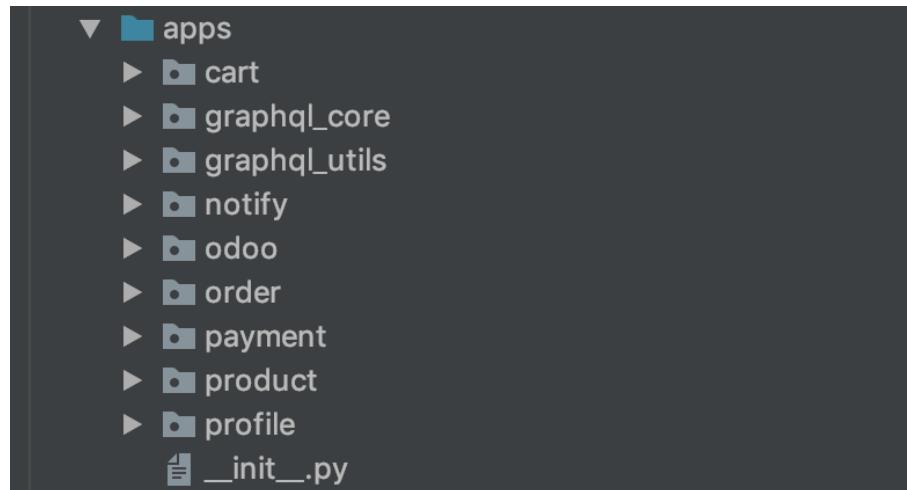


FIGURE 5.5: Le répertoire des modules de projet

5.1.5 Répertoire Tests

C'est le répertoire qui contient les tests de chaque module dans le répertoire apps.

Ceci est le répertoire qui contient les tests pour chaque module du répertoire apps. Dans notre projet, nous testons par module et répartis en 4 types de tests :

- **Test Queries** sert à tester les requêtes de recherche dans la base de données en utilisant GraphQL.
- **Test API** ce sont les tests API externes utilisés dans le module comme knet payment, odoo item
- **Test de Mutation** ce sont les tests des demandes de création et de modification, et la suppression dans la base de données en utilisant GraphQL

Pour démarrer les tests, tapez simplement la commande "make test" pour lancer tous les tests.

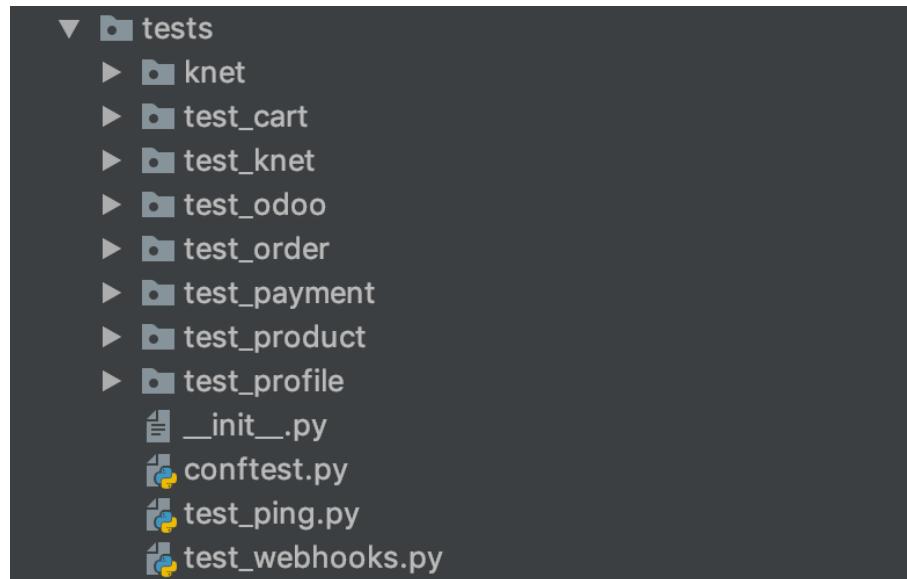


FIGURE 5.6: Le répertoire des tests

5.1.6 Les fichiers Docker

Il y a 4 fichiers docker et docker dans notre application :

- **Dockerfile** : c'est le fichier qui définit les dépendances nécessaires à la construction de l'image JOOD-CORE

- **Docker-compose.env** : contient les variables d'environnement telles que DATABASE_URL
- **Docker-compose.yaml** : contient les services docker utilisé par l'image JOOD-CORE
- **Docker-entrypoint.sh** : contient les commandes nécessaires à exécuter lors du démarrage de l'application
- **Makefile** : contient des commandes spécifiques pour notre Conteneur Docker , telles que make tests ...

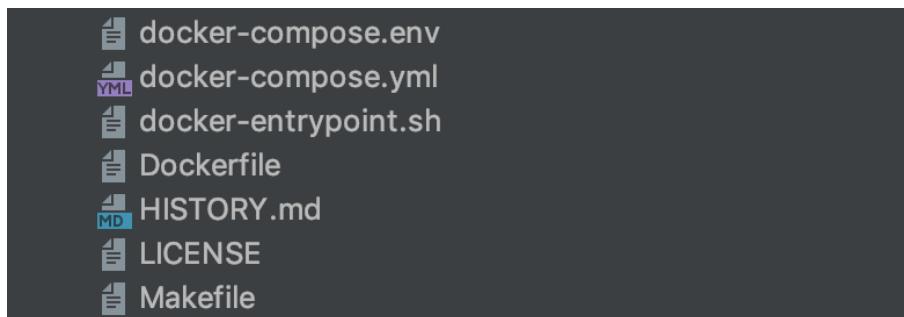


FIGURE 5.7: Le fichiers docker de projet

5.1.7 Autre Fichiers

- **manage.py** :fichier principale de django c'est le point d'entrée de projet
- **Pipfile** : contient les versions des package ,docker utilise ce fichier pour construire l'image et installer toutes les dépendances de l'image
- **Pipfile.lock** :contient les information de chaque dépendance du projet comme le lien de téléchargement), requirements.txt (le fichier qui contient les exigences du projet

5.2 Interaction avec l'ERP Odoo

Pour une meilleure gestion des stocks, des ventes, des ressources humaines et des clients, la société a décidé d'utiliser l'ERP Odoo et de la relier avec la plate-forme JOOD afin de synchroniser les données des commandes, les produits et les clients après chaque opération de création, de modifier ou de supprimer la base de données du site à l'aide du XML. Protocole-RPC pour connecter Jood-Core à Jood-Odoo et CELERY afin que les tâches XML-RPC soient synchronisées avec les opérations JOOD-CORE et doivent se trouver à l'arrière-plan du site dans l'objective de ne pas ralentir le fonctionnement, c'est-à-dire que les erreurs de ces tâches n'influencent pas sur le fonctionnement normal du site.

les figures au dessous montrent quelques fonctions implémentées sur odoo

```
def auth():
    common = client.ServerProxy('{0}:{1}/xmlrpc/2/common'.format(settings.ODOO_URL, settings.ODOO_XMLRPC_PORT))
    uid = common.authenticate(settings.ODOO_DATABASE, settings.ODOO_USER, settings.ODOO_PASSWORD, {})
    return uid
```

FIGURE 5.8: Connexion Jood-Core au serveur Jood-Odoo avec XRPC

```
def odoo_models():
    models = client.ServerProxy('{0}:{1}/xmlrpc/2/object'.format(settings.ODOO_URL, settings.ODOO_XMLRPC_PORT))
    return models
```

FIGURE 5.9: Obtenir les modèles de Odoo

```
def create_odoo_user(user):
    if not user.odoo_id:
        uid = auth()
        models = odoo_models()
        id = models.execute_kw(settings.ODOO_DATABASE, uid, settings.ODOO_PASSWORD, 'res.partner', 'create', [
            {'name': user.get_full_name,
             'email': user.email,
             'phone': str(user.phone_number),
            }])
        user.odoo_id = id
        user.save()
        return id
    return user.odoo_id
```

FIGURE 5.10: Creation d'un ulisateur sur Odoo

```
@app.task
def task_create_user_odoo(user_id):
    try:
        user = User.objects.get(pk=user_id)
        return create_odoo_user(user)
    except User.DoesNotExist:
        return None
```

FIGURE 5.11: création d'un utilisateur sous Odoo avec celery

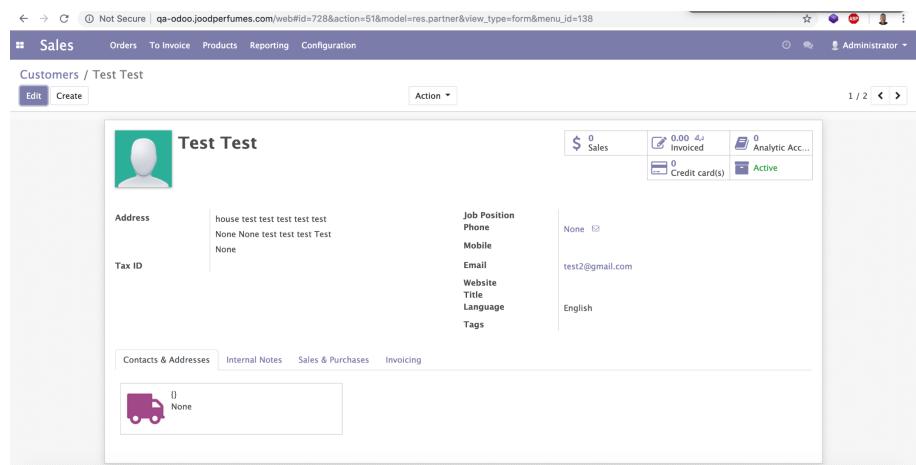


FIGURE 5.12: La création d'un ordre pour un utilisateur dans Odoo

5.3 L'API de Jood

GRAPHQL API : Pour servir API graphql dans notre projet, nous avons suivi les étapes suivantes pour chaque classe : On prend la class Product comme exemple

Création de types et de schémas Dans GraphQL, un Type est un objet pouvant contenir plusieurs champs. Chaque champ est calculé via des résolveurs, qui renvoie une valeur. Une collection de types s'appelle un schéma. Chaque schéma a un type spécial appelé requête pour obtenir des données du serveur et une mutation pour envoyer des données au serveur.

```

class ProductType(DjangoObjectType):
    primary_image = graphene.String()
    images = graphene.List(graphene.String)
    gift_wrappings = graphene.List('product.graphql.types.GiftWrappingType')
    price = graphene.String()
    country = graphene.String()
    currency = graphene.String()
    site_key = graphene.String()
    prices = graphene.List(ProductSiteType)

    class Meta:
        exclude_fields = ('published', 'odoo_id')
        model = Product
        interfaces = (relay.Node,)

    def resolve_primary_image(self, info):
        photo = self.photos.filter(is_primary=True).first()
        if photo:
            return photo.image.url

    def resolve_images(self, info):
        return [photo.image.url for photo in self.photos.all()]

    def resolve_gift_wrappings(self, info):
        return self.gift_wrappings.all()

    def resolve_price(self, info):
        return self.current_site_price

    def resolve_country(self, info):
        return self.current_site_country

    def resolve_currency(self, info):
        return self.current_site_currency

    def resolve_site_key(self, info):
        return self.current_site_key

    def resolve_prices(self, info):
        return self.product_prices.all()

    @classmethod
    def get_node(cls, info, id):
        try:
            return cls._meta.model.objects.get(pk=id, published=True)
        except cls._meta.model.DoesNotExist:
            return None

```

FIGURE 5.13: Exemple d'un ProductType avec graphene-django

Dans l'extrait de code ci-dessus, ProductType a été créé à l'aide de DjangoObjectType - un type personnalisé disponible dans Graphene Django.

```

from graphene_django.filter import DjangoFilterConnectionField
from graphene import relay

from .filters import ProductFilterSet, GiftWrappingFilterSet
from .types import CategoryType, ProductType, GiftWrappingType

class ProductQuery(object):
    products = DjangoFilterConnectionField(ProductType, filterset_class=ProductFilterSet)

```

FIGURE 5.14: Exemple de ProductQuery avec graphene-django

Dans l'extrait de code ci-dessus, la requête de products spécial a été créée avec un résolveur pour les liens de champ, qui renvoie tous les liens

```

class Mutations(CartMutation, OrderMutation, ProfileMutations, NotifyMutation, graphene.ObjectType):
    pass

class Query(PaymentMethodQuery, CartQuery, OrderQuery, ProductQuery, GiftWrappingQuery, ProfileQuery, SiteQuery,
            NotificationQuery, graphene.ObjectType):
    pass

schema = graphene.Schema(query=Query, mutation=Mutations)

```

Expected type 'Optional[GraphQLObjectType]', got 'Type[Mutations]' instead more... (#F1)

FIGURE 5.15: Déclaration des requêtes et des mutations dans le schéma du Joodperfumes

GraphiQL est un IDE graphique intégré au navigateur. En d'autres termes, un terrain de jeu. Notez que vous devez désactiver la protection Django CSRF.

```

urlpatterns = [
    url(r'^ping/?', lambda request: HttpResponse('pong'), name='ping'),
    url(r'^admin/', admin.site.urls),
    url(r'^graphiql', csrf_exempt(FileUploadGraphQLView.as_view(graphiql=True)), name='graphiql'),
    path('', csrf_exempt(views.my_view), name='sparkpost_webhook'),
    url(r'^payment/', include('apps.payment.knet.urls', namespace='payment')),
    url(r'^order/', include('apps.order.urls', namespace='order'))
]

```

FIGURE 5.16: L'ajout de graphiql au urlpattern de projet

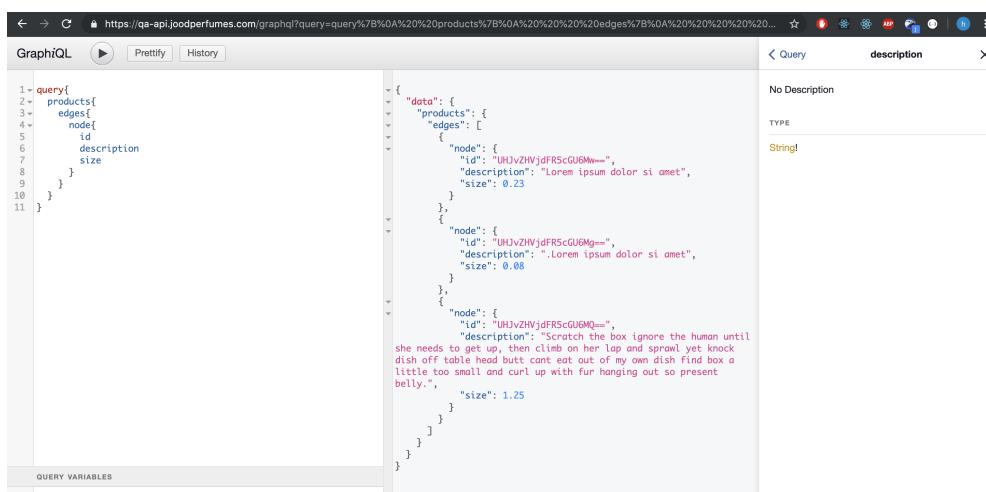


FIGURE 5.17: La Vue de Graphiql api Jood

Sur la droite se trouve l'explorateur de documentation, des informations sur notre schéma créées automatiquement par introspection. Sur la gauche, pour créer votre requête, tapez ce qui suit :

```

query{
  products{
    edges{
      node{
        id
        description
        size
      }
    }
  }
}
  
```

FIGURE 5.18: La requête products dans graphiql

Vous devriez voir un lien de réponse ceci :

```

{
  "data": {
    "products": {
      "edges": [
        {
          "node": {
            "id": "UHJvZHVjdFR5cGU6Mw==",
            "description": "Lorem ipsum dolor si amet",
            "size": 0.23
          }
        },
        {
          "node": {
            "id": "UHJvZHVjdFR5cGU6Mg==",
            "description": ".Lorem ipsum dolor si amet",
            "size": 0.08
          }
        },
        {
          "node": {
            "id": "UHJvZHVjdFR5cGU6MQ==",
            "description": "Scratch the box ignore the human until
she needs to get up, then climb on her lap and sprawl yet knock
dish off table head butt cant eat out of my own dish find box a
little too small and curl up with fur hanging out so present
belly.",
            "size": 1.25
          }
        }
      ]
    }
  }
}
  
```

FIGURE 5.19: Le résultat de la requête products dans graphiql

5.4 Conclusion

Dans ce chapitre, on a pu présenter le fonctionnement de la côté backend, qui a servir une API graphql et aussi api Rest , consommable par le Front end de l'application, de tout en mettant en place les différents outils techniques et logicielles qui nous a permis de concevoir les différents modules de projet.

6

Jood - développement Front-end

6.1 Introduction

Nous présenterons dans cette section un aperçu général sur le fonctionnement de la plateforme en utilisant quelques captures d'écran. Toutefois, il faut mentionner que la plateforme est encore en version Beta et que plusieurs de ses aspects ne sont pas encore implementés dans l'UI, soit parce qu'il sont encore en phase de tests dans le Back-end ou bien qu'ils n'ont pas une priorité élevée...

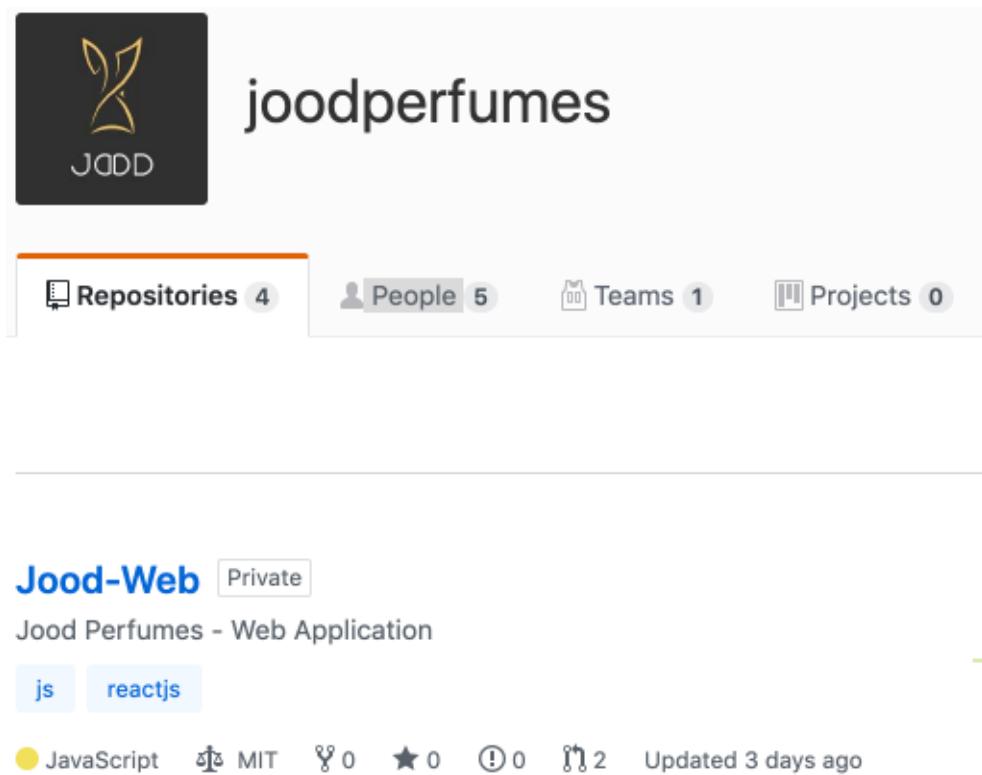


FIGURE 6.1: Le référentiel du front-end sur github

6.2 Organisation du projet

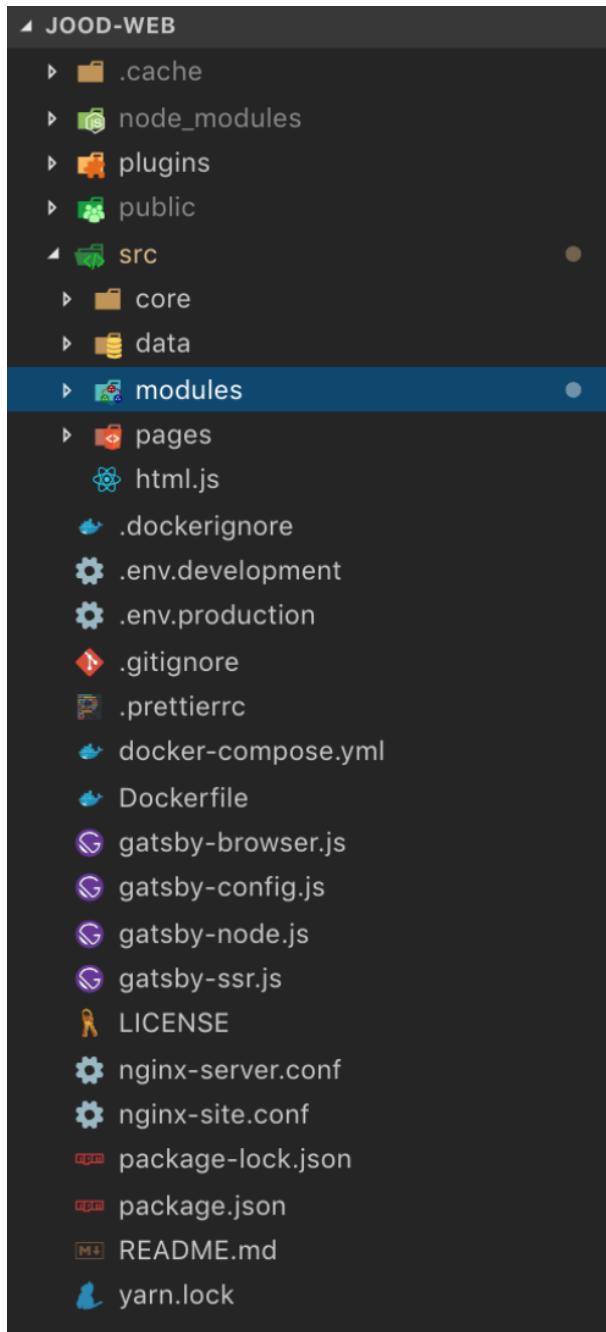


FIGURE 6.2: La structure du Jood-Web

Dossiers

- **cache** : Généré automatiquement. Ce dossier est un cache interne créé automatiquement par Gatsby. Les fichiers contenus dans ce dossier ne sont pas destinés à être modifiés. Devrait être ajouté au fichier .gitignore s'il n'a pas déjà été ajouté.
- **public** : généré automatiquement. La sortie du processus de construction sera exposée dans ce dossier. Devrait être ajouté au fichier .gitignore s'il n'a pas déjà été ajouté.
- **node_module, /package.json** : les packages modules des dépendances des framework javascript pour le projet frontend.
- **plugins** : Ce dossier héberge tous les plugins spécifiques au projet («locaux») qui ne sont pas publiés sous forme de paquet npm. Consultez la documentation du plugin pour plus de détails.

- **src** : Ce répertoire contiendra tout le code lié à ce que vous verrez sur le frontend de votre site (ce que vous voyez dans le navigateur), comme tête de votre site, ou un modèle de page. « Src » est une convention pour « code source »

pages : Les composants sous /src/pages deviennent automatiquement des pages avec des chemins basés sur leur nom de fichier

data : Ce répertoire contient les fichiers comportant les données statiques de l'application tel que countries.js et currencies.js.

core : Ce répertoire contient les fournisseurs de service pour l'application sous le répertoire /provider (tel que : apolloClient, AuthManager...etc) ainsi que la gestion du state avec Redux dans la totalité de l'application

modules : Ce répertoire contient tous les composants partiels dans l'application, catégorisé par fonctionnalité (tel que : auth : gestion d'authentification, order : gestion des commandes, giftWrapping. .etc) chaque module peut contenir plusieurs sous répertoire tel que /hoc, /actions, /component, /container

html.js : Pour la configuration personnalisée de .cache/default _html.js par défaut

Fichiers

- **gatsby-browser.js** : Ce fichier est l'endroit où Gatsby s'attend à trouver un usage quelconque des API de navigateur Gatsby (le cas échéant). Ceux-ci permettent la personnalisation / extension des paramètres par défaut de Gatsby affectant le navigateur.
- **gatsby-config.js** : Il s'agit du fichier de configuration principal d'un site Gatsby. C'est là que vous pouvez spécifier des informations sur votre site (métadonnées), telles que le titre et la description du site, les plugins Gatsby que vous souhaitez inclure, etc. Consultez la documentation de configuration pour plus de détails.
- **gatsby-node.js** : ce fichier est l'emplacement où Gatsby s'attend à trouver un usage quelconque des API de nœud Gatsby (le cas échéant). Ceux-ci permettent la personnalisation / extension des paramètres par défaut de Gatsby qui affectent des éléments du processus de construction du site.
- **gatsby-ssr.js** : ce fichier est l'endroit où Gatsby s'attend à trouver un usage quelconque des API de rendu côté serveur de Gatsby (le cas échéant). Celles-ci permettent la personnalisation des paramètres par défaut de Gatsby ayant une incidence sur le rendu côté serveur.

6.3 Workflow de JoodPerfumes

Cette partie représente un scénario d'utilisation du projet par un utilisateur ordinaire dans lequel je vais expliquer plus le fonctionnement interne de chaque partie

6.3.1 Interfaces d'utilisateur principale

Après ouverture du projet en local en accédant à **http://localhost:8000/** ou via **https://qa-web.joodperfumes.com/** la page d'accueil est par défaut initialisé par les données adaptées pour country= kuwait https .../kw/

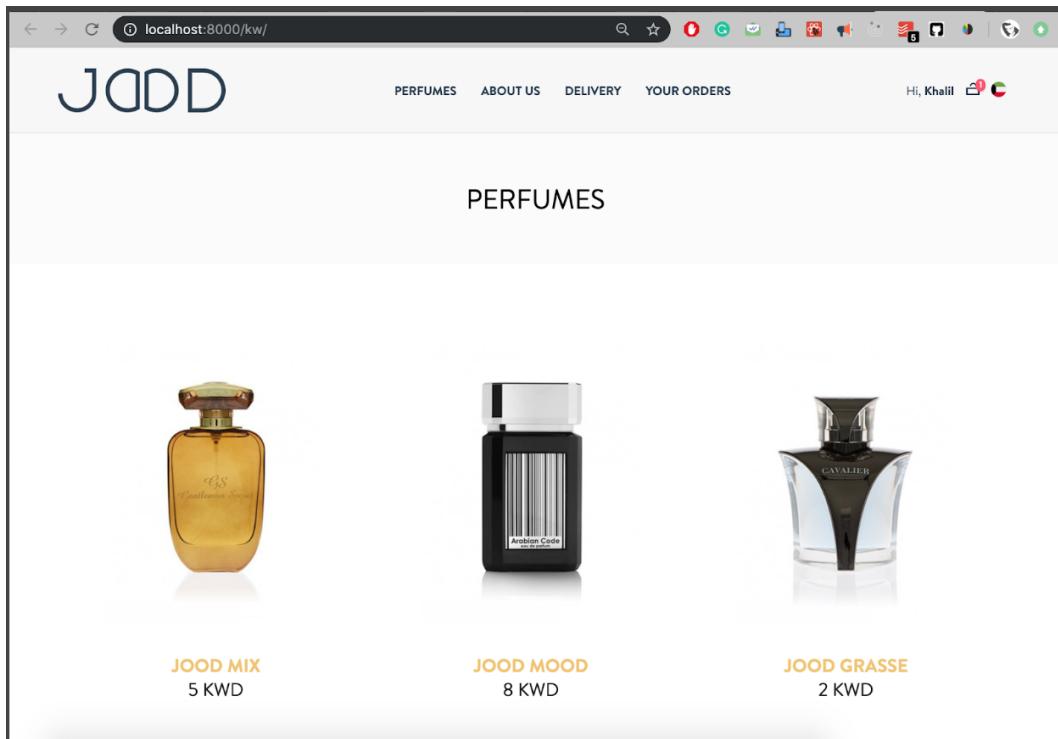


FIGURE 6.3: L'interface principale du Joodperfumes

La page est composé de plusieurs Components, dont **Products** est le plus principale : ce composant liste tous les produits venant d'un composant d'ordre supérieur (higher order component, référencé aussi HOC) appelé **withProducts** Qui à son tour est basé sur une requête **GraphQL query**

```
const Products = ({ products: { fetched, items } }) => (...  
  <Container className="pv5 nl3 nr3 cf">  
    {fetched &&  
      Object.keys(items).map(id => (  
        <div key={id} className="fl-ns w-33-ns ph4 flex justify-center">  
          <ProductCard id={id} />  
        </div>  
      ))} </Container>)
```

WithProducts est un composant d'ordre supérieur, localisé sous ‘src/modules/products/hoc/withProducts‘

Sa seule mission est de récupérer les produits existants dans la base de données depuis l'interface GraphQL en utilisant la requête mentionnée précédemment

Ensuite ce composant transmis les produits récupérés au props à l'aide de la fonction mapStateToProps

De l'autre côté, le composant qui a besoin des produits doit inclure le WithProduct dans son HOC (composant d'ordre supérieur) pour recevoir tous les produits comme props

Scenario : Ajout des produits au panier Un utilisateur peut naviguer dans les différents produits offerts par l'application, et ajouter les produits qu'il désire au panier, Pourtant l'ajout d'un produit au panier donne accès à deux scénarios possibles : ‘Ajout normal’ ou ‘Ajout avec papier d'emballage cadeau’

Dans le cas d'ajout d'un produit avec l'option ‘Gift Wrapping’, l'utilisateur est demandé de choisir un papier cadeau via une fenêtre Pop-up

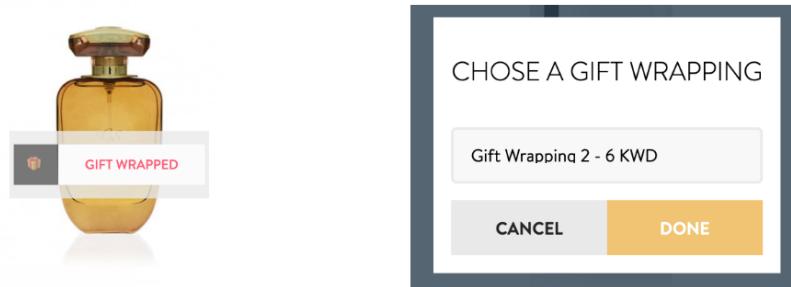


FIGURE 6.4: Le choix d'un papier cadeau (Gift Wrapping)

Après avoir sélectionné les produits qu'il désire commander, l'utilisateur a la possibilité de visualiser son panier 'card' en cliquant sur l'icône située dans le header (figure suivante)

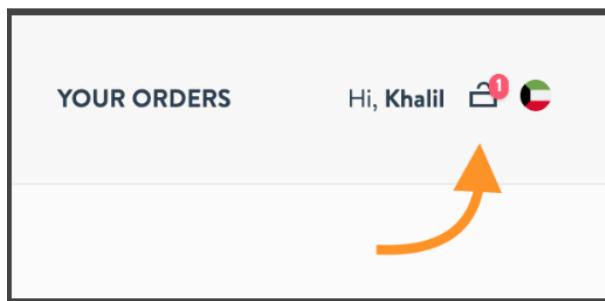


FIGURE 6.5: L'icône de panier du plateforme Jood

Chaque élément ajouté dans le panier modifie le state global de l'application selon le module

```
addresses: {fetched: false, items: ...}
app: {country: null, notifications: Array(0)}
auth: {authenticated: true, token: "...", user: null}
cart: {fetched: false, items: ...}
giftWrapping: {fetched: false, items: ...}
orders: {fetched: false, items: ...}
products: {fetched: false, items: ...}
```

Dans cet exemple Cart contient toutes informations en relation avec le panier d'utilisateur et les produits qui l'intéressent. Toutes ces informations à propos du panier sont regroupés dans un component général appelé cartPage

```
const CartPage = () => (<Layout><Cart /></Layout>)
```

cartPage donne la main à l'utilisateur de faire des modifications à son panier, principalement :
Modifier la quantité de chaque produit Ajouter ou supprimer les 'Gift Wrapping' pour chaque produit Supprimer un produit du panier

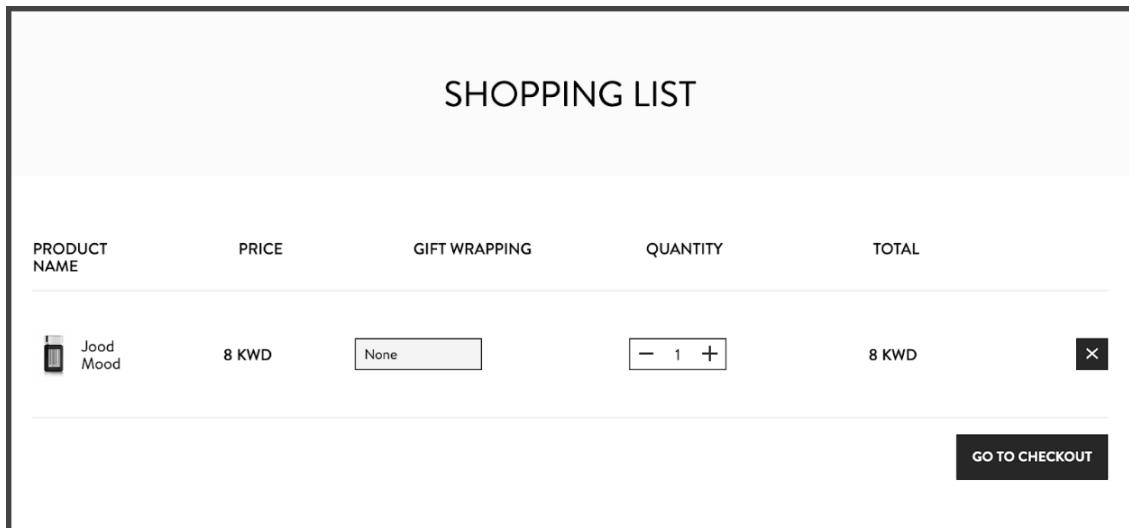


FIGURE 6.6: Le contenu d'un panier dans Jood

Chaque élément du panier est visualisé grâce à un composant appelé Row, qui s'occupe de la gestion de toutes fonctionnalités spécifiques à ce produit

```
{Object.entries(items).map(([product, item]) => {
  return (<Row key={product} quantity={item.quantity} selected={ item.giftWrapping.id}
    product={product} />)
})
}
```

Items représente les éléments du panier (cart), chaque élément est un objet json :

```
{
  item : {
    quantity,
    giftWrapping
  },
  product,
}
```

La gestion des éléments du panier, (appelé cardItem) est basé sur 3 composants HOC (Composants d'ordre supérieur) principaux qui interagissent avec l'interface GraphQL :

- **withAddCartItem** : ajout d'un élément au panier en exécutant la mutation addCartItem

```
const ADD_CART_ITEM = gql`mutation addCartItem($input: AddCartItemInput!){...}`
```

- **withUpdateCartItem** : modification d'un élément au panier avec la mutation updateCartItem

```
const UPDATE_CART_ITEM = gql`mutation updateCartItem($id: ID!, $input: UpdateCartItemInput!){...}`
```

- **withDeleteCartItem** : suppression d'un élément au panier en exécutant la mutation deleteCartItem

```
const DELETE_CART_ITEM = gql`mutation deleteCartItem($id: ID!) {...}`
```

Scenario : Procéder au paiement Afin de pouvoir procéder au paiement, l'utilisateur doit accéder à la page de paiement via Go to Checkout (voir la figure précédente), ce qui entraîne le chargement des nouveaux composants (figure suivante).

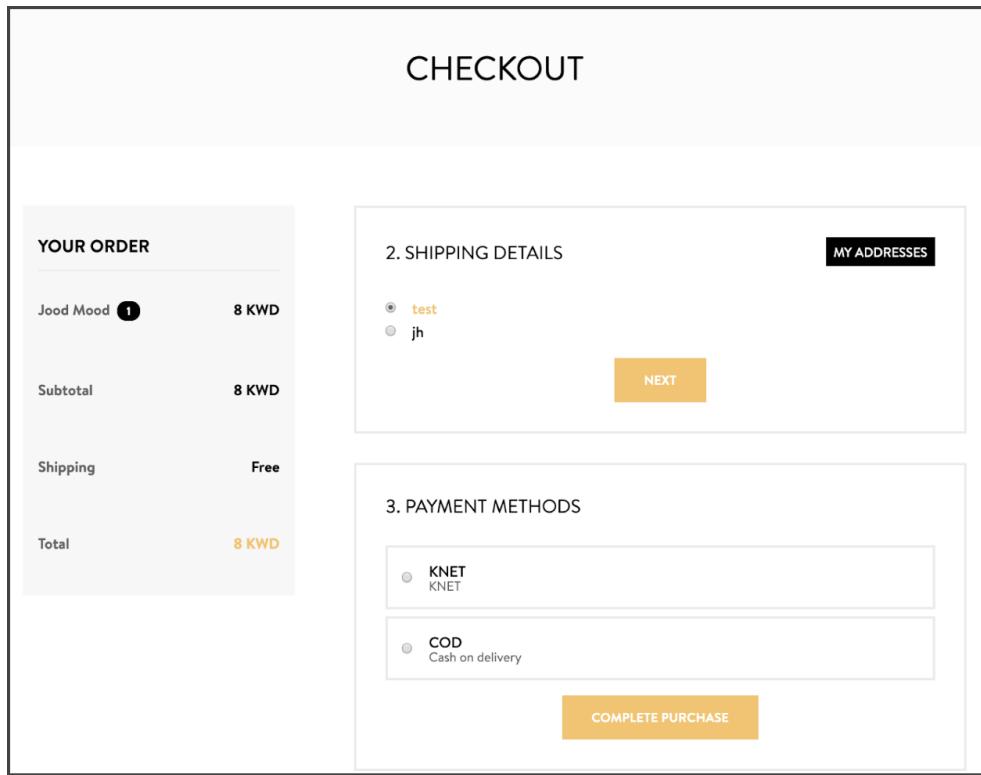


FIGURE 6.7: Le choix de la méthode de paiement

La gestion du module checkout est regroupé sous src/modules/order Principalement ce module gère :

- Le choix de l'adresse de livraison Shipping Details qui permet à l'utilisateur de choisir une addresses existantes dans son profil ou bien ajouter une nouvelle adresse. (Le composant src/modules/order/components/ShippingDetails)
- Le choix de la méthode de paiement Payment Methods (en utilisant l'interface de paiement KNet ou l'option du paiement à la livraison COD). (Le composant src/modules/order/components/Payment)
- Le calcul du montant de la commande

Après la saisie des informations nécessaires, et l'envoi de la requête l'utilisateur doit s'attendre à l'une des deux choses :

- La redirection vers la plateforme de paiement en ligne KNet pour effectuer le paiement.
- la redirection vers la page du succès de la commande

Scenario : Ajout d'une nouvelle adresse Quand procéder au paiement, le composant ShippingDetails fournit une manipulation à double sens, l'utilisateur peut choisir une des addresses lié à son compte, sinon, l'application lui donne la main d'ajouter d'une nouvelle adresse en cas de besoin à l'aide du formulaire suivant

The screenshot shows a form titled "2. SHIPPING DETAILS". At the top right is a button labeled "MY ADDRESSES". The form contains several input fields:

- Phone number: +212652525389
- Area
- Type selection: Apartment (selected), House, Office
- Address Title
- Block
- Building
- Street
- Floor
- Avenue (Optional)
- Apartment No.
- Additional Directions (Optional)

At the bottom is a large orange "ADD ADDRESS" button.

FIGURE 6.8: Ajout d'une nouvelle adresse

Les adresses diffèrents selon le type de l'habitation (apartment, House, Office), par exemple : une adresse de type House ne contient pas les champs tel que [Building, Floor, Apartment No., Office]

La validation des formulaires est assurée par Yup qui est un validateur de schéma d'objet JavaScript et analyseur d'objet alors que le formulaire est assuré par Formik qui est une bibliothèque destinée pour la gestion de tous cas d'utilisation du formulaire.

En plus de ca, la gestion des adresses de l'utilisateur est disponible grâce au menu situé au header

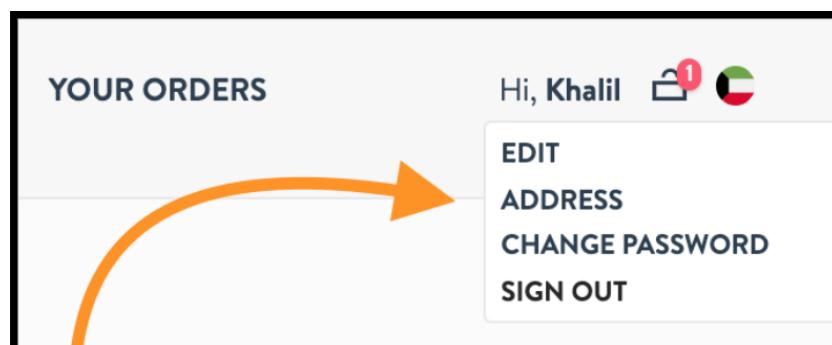


FIGURE 6.9: Menu de gestion (d'adresses, mot de passe, profile)

Ce qui redirige l'utilisateur vers le composant ListAddressPage :

ADDRESS LIST		
NAME	ADDRESS	
test	fgh dfg build, floor 15	<button>EDIT</button> <button>X</button>
jh	ihb hjg bw, jhbfe fed	<button>EDIT</button> <button>X</button>
ADD NEW ADDRESS		

FIGURE 6.10: Interface de gestion d'adresses

ListAddressPage donne possibilité de gérer les adresses de l'utilisateur grâce aux HOC (Composants d'ordre supérieur) qui interagissent avec l'interface GraphQL :

- withAddAddress : ajout d'une nouvelle adresse en exécutant la mutation suivante

```
const ADD_ADDRESS = gql`  
mutation create($input: CreateShippingAddressInput!) {...}`
```

- withUpdateAddress : contient la mutation qui permet la modification d'une adresse

```
const UPDATE_ADDRESS = gql`
```

```
mutation update($input: UpdateShippingAddressInput!, $id: ID!) {...}`
```

- withDeleteCartItem : contient la mutation qui permet la suppression d'une adresse

```
const DELETE_ADDRESS = gql`  
mutation($id: ID!) {  
  deleteShippingAddress(id: $id) {...}  
}
```

6.3.2 Autres pages et composants

Composant Detail Product sous src/modules/product/container/Detail Visualise les informations importantes d'un seul produit

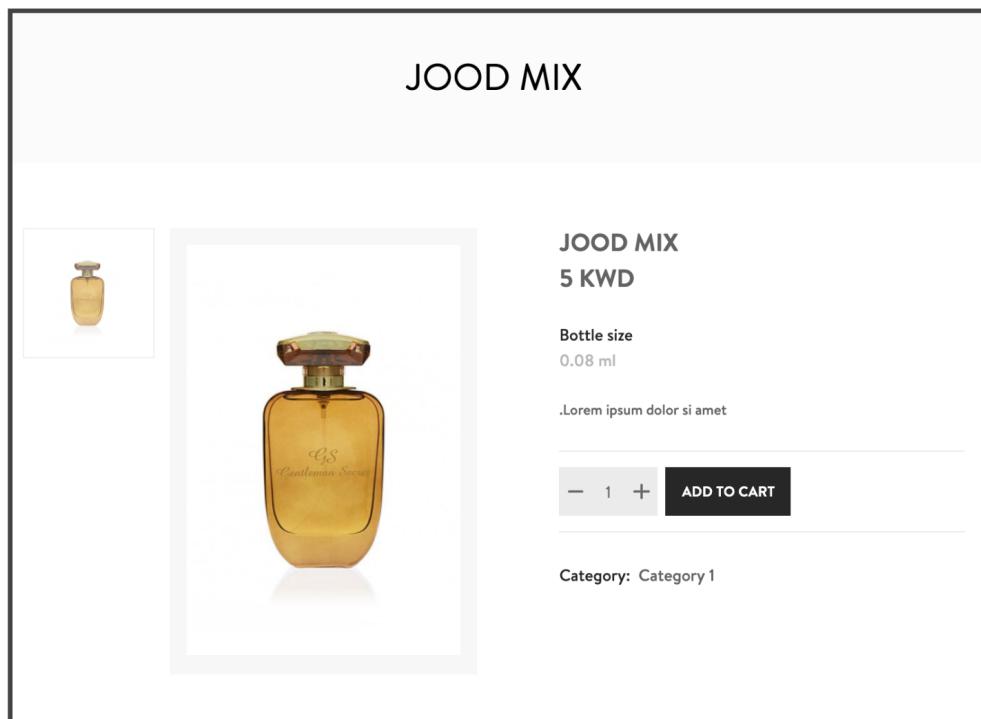


FIGURE 6.11: détails d'un produit

Composant Edit Profile sous src/modules/profile/container/Edit Donne la main à l'utilisateur a modifié ses données personnelles

A screenshot of a web page titled "EDIT PROFILE". On the left is a placeholder for a user profile picture with dimensions "160 x 160". To the right are two input fields: "First name *" containing "Khalil" and "Last name *" containing "MAGHRAOUI". Below these are two input fields: "Phone number *" containing "+212652525389". At the bottom is a yellow "SAVE" button.

FIGURE 6.12: Modification des données d'un utilisateur

Composant Change Password sous src/modules/profile/container/ChangePassword Donne la main à l'utilisateur à réinitialiser son mot de passe

The screenshot shows a 'CHANGE PASSWORD' form. At the top, the title 'CHANGE PASSWORD' is centered. Below it, there are three input fields: 'Old password *', 'New Password *', and 'Password Confirmation *'. Each field has a corresponding empty input box below it. At the bottom, there is a large orange button labeled 'CHANGE PASSWORD'.

FIGURE 6.13: Interface pour changer un mot de passe

6.4 Conclusion

Dans ce chapitre, on a pu présenter le fonctionnement de la côté front end, et aussi le fruit de notre travail sur la plateforme de tout en mettant en place les différents outils techniques et logicielles qui nous a permis de concevoir les différents modules de projet.

Conclusion générale et perspectives

Le présent rapport représente la synthèse du projet de fin d'études que j'ai effectué au sein de l'organisme Orange Bytes, et qui consiste à le développement d'une plateforme e commerce , concevoir et à implémenter une infrastructure Cloud adaptée pour l'architecture d'une plateforme e-commerce.

Notre mission a consisté à contribuer dans partie BackEnd et DevOps dans le développement, la conception et l'implémentation des environnements Cloud pour les différents services. Notre mission a consisté aussi sur la maintenance corrective et préventive de la totalité de l'infrastructure Cloud de projet, la correction des différents anomalies .

Nous avons aussi travaillé pour améliorer la sécurité et la performance de la plateforme via la création d'une infrastructure Cloud scalable, hautement disponible et résiliente, augmenter la productivité et accélérer les processus de déploiement et de mise en production par la mise en place d'un système d'intégration continue (CI) et de déploiement continu (CD) lié avec l'infrastructure de projet, à cela s'ajoute la mise en œuvre des évolutions souhaitées par le client.

Nous avons également mis en pratique un système de Monitoring pour surveiller la santé des environnements Cloud de A à Z afin de livrer un projet qui répond aux normes de qualité et qui satisfait le client et l'utilisateur final.

Pour mettre en œuvre notre projet, nous avons été amenés, dans un premier lieu, à établir une étude conceptuelle afin de dégager les différents modules de la plateforme, ainsi qu'une étude des outils et technologies susceptibles de convenir à sa réalisation et la réalisation de sa infrastructure Cloud.

Dans un second lieu, nous avons abordé l'analyse et la conception du projet. Un certain nombre de diagrammes ont été élaborés afin de mieux diviser le projet, ce qui a facilité sa mise en œuvre. Finalement, l'implémentation des différents service et ses environnements Cloud dont Le résultat final répond aux besoins déjà cités dans ce rapport.

Liste des abréviations

<i>AMQP</i>	Advanced Message Queuing Protocol
<i>API</i>	Application Programming Interface
<i>AWS</i>	Amazon Web Service
<i>CD</i>	Continuous Development
<i>CI</i>	Continuous Integration
<i>CLI</i>	Command Line Interface
<i>CMS</i>	Content Management System
<i>CPU</i>	Central Processing Unit
<i>CRUD</i>	Create, Read, Update and Delete
<i>CSRF</i>	Cross-Site Request Forgery
<i>CSS</i>	Cascading Style Sheets
<i>CSV</i>	Comma-Separated Values
<i>DDD</i>	Domain Driven Design
<i>DevOps</i>	DEVelopment and OPerationS
<i>DNS</i>	Domain Name System
<i>DRY</i>	Don't Repeat Yourself
<i>ESB</i>	Enterprise Service Bus
<i>HTML</i>	HyperText Markup Language
<i>HTTP</i>	HypterText Transfer Protocol
<i>IaaS</i>	Infrastructure as a Service
<i>IANA</i>	Internet Assigned Authority
<i>IP</i>	Internet Protocol
<i>IT</i>	Information Technology
<i>ITS</i>	Intelligent Tutorial System
<i>JPL</i>	Jet Propulsion Laboratory
<i>JSON</i>	JavaScript Object Notation
<i>LMS</i>	Learning Management System
<i>MVC</i>	Model View Controller
<i>MVT</i>	Model View Template
<i>NASA</i>	National Aeronautics and Space Administration
<i>NoSQL</i>	Not only SQL
<i>ORM</i>	Object Relational Mapper

<i>PaaS</i>	Platform as a Service
<i>PDF</i>	Portable Document Format
<i>REST</i>	REpresentational State Transfer
<i>RPC</i>	Remote Procedure Call
<i>RSS</i>	Rich Site Summary
<i>SaaS</i>	Software as a Service
<i>SGBD</i>	Système de Gestion de Base de Données
<i>SI</i>	Système d'Information
<i>SOA</i>	Service Oriented Architecture
<i>SOAP</i>	Simple Object Access Protocol
<i>SQL</i>	Structured Query Language
<i>SSO</i>	Single Sign-On
<i>TDD</i>	Test Driven Developpement
<i>UI</i>	User Interface
<i>UML</i>	Unified Modeling Language
<i>URI</i>	Uniform Resource Identifier
<i>URL</i>	Uniform Ressource Locator
<i>UUID</i>	Universally Unique IDentifier
<i>UX</i>	User eXperience
<i>VM</i>	Virtual Machine
<i>XML</i>	eXtensible Markup Language
<i>XSS</i>	CROSS-Site Scripting

Webographie

Le 05/02/2019	http://www.abytes.com/
Le 07/02/2019	https://fr.wikipedia.org/wiki/Python_(langage)
Le 14/02/2019	https://docs.docker.com/
Le 21/02/2019	https://docs.djangoproject.com/en/2.2/
Le 22/02/2019	https://fr.wikipedia.org/wiki/SQLite
Le 22/02/2019	http://www.djangoproject-rest-framework.org/tutorial/quickstart/
Le 01/03/2019	https://www.howtographql.com/
Le 02/03/2019	https://blog.octo.com/graphql-et-pourquoi-faire/
Le 03/03/2019	https://stackabuse.com/building-a-graphql-api-with-django/
Le 05/04/2019	https://www.youtube.com/watch?v=BDBvHOaaKHo&list=PLv2a_5pNAko0Mijc6mnv0
Le 05/04/2019	https://docs.aws.amazon.com/
Le 22/04/2019	https://learn.hashicorp.com/terraform/
Le 01/05/2019	https://www.terraform.io/docs/providers/aws
Le 08/05/2019	https://www.terraform.io/docs/providers/github/index.html
Le 15/05/2019	https://www.terraform.io/docs/providers/cloudflare
Le 02/06/2019	https://medium.com/devopslinks/deep-dive-into-aws-vpc-and-vpc-peering-3
Le 15/06/2019	https://medium.com/@amolkokje/aws-setup-bastion-host-ssh-tunnel-f5ec5cf
Le 05/07/2019	https://www.lucidchart.com/