| OCPizza  P8  Dossier de conception technique  Version 1.0 |
| --- |
| **Auteur :**  Ianis Richardier  Analyste programmeur |

**Table des matières**

**1.** [**Versions**](#_heading=h.b1hc63eb4l30) **3**

**2.** [**Introduction**](#_heading=h.r92x0etgi5f7) **4**

**2.2** [**Objet du document**](#_heading=h.3147ljb56rkl) **5**

**2.3** [**Références**](#_heading=h.sswu0ox6acqe) **5**

**3.** [**Architecture Technique**](#_heading=h.fpyfc32h5lor) **6**

**3.1** [**Introduction**](#_heading=h.4r0giovno6g) **6**

**3.2** [**Site web**](#_heading=h.w2w5idgc2hzm) **6**

**3.3** [**Diagramme de composants**](#_heading=h.wvng1l82t29) **8**

**3.4** [**Composants généraux**](#_heading=h.yrwnc4ejxn6q) **9**

**4.** [**Architecture de Déploiement**](#_heading=h.2w1cokqhqb5w) **10**

**4.1** [**Modèle physique de données**](#_heading=h.taj8hgbm5mj0) **10**

**4.2** [**Diagramme de déploiement**](#_heading=h.jwbbd3lp5w36) **11**

**4.3** [**Serveur web**](#_heading=h.xkfui1c37e70) **12**

**4.4** [**Serveur de base de données**](#_heading=h.p2dkhtlkzufv) **12**

**5.** [**Architecture logicielle**](#_heading=h.nbp3ilup2emn) **13**

**5.1** [**Principes généraux**](#_heading=h.cw0o63675s4) **13**

**5.2** [**Infrastructure CMS**](#_heading=h.9amxbzfluyjb) **13**

**6.** [**Points particuliers**](#_heading=h.648uoz27wx9g) **15**

**6.1** [**Gestion des logs**](#_heading=h.s8t2th400v9j) **15**

**6.2** [**Fichiers de configuration**](#_heading=h.gjdgxs) **15**

**6.2.1** [**Installation de Magento**](#_heading=h.g7dych8tl2b4) **15**

**6.2.2** [**Module de paiement**](#_heading=h.g1jp3qd219dd) **17**

**6.3** [**Ressources**](#_heading=h.qmcjzpx3c8gf) **18**

**6.4** [**Environnement de développement**](#_heading=h.hwh8kiyxbca4) **19**

**6.5** [**Procédure de packaging / livraison**](#_heading=h.h2k8dv38ymzt) **19**

**7.** [**Glossaire**](#_heading=h.dvx3rlhmu33i) **20**

# Versions

| **Auteur** | **Date** | **Description** | **Version** |
| --- | --- | --- | --- |
| Ianis Richardier | 15/10/2021 | Création du document | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Objet du document

* Ce document est destiné à la ***maîtrise d’ouvrage (MOA)\**** et à la ***maîtrise d'œuvre (MOE)\****. Il constitue le dossier de conception technique du projet OCPizza.
* La finalité de ce document est de permettre aux concepteurs de la solution de réaliser leurs tâches dans les meilleures conditions possibles.
* Les éléments composant ce dossier ont été élaborés à partir du document de spécifications fonctionnelles réalisé par IT Consulting & Development.

## Références

Pour de plus amples informations, se référer également aux éléments suivants:

1. **DCF - 1.0** : Dossier de conception fonctionnelle de l'application
2. **DE - 1.0** : Dossier d’exploitation
3. **PVL - 1.0** : Procès verbal de livraison

***\** Pour plus d’informations concernant les termes en gras & italique, se référer au glossaire en fin de document*.***

# Architecture Technique

## Introduction

Magento permet de ne pas avoir à créer une solution “***from scratch***”. Autrement dit, d’éviter la création d’une solution qui nécessite des investissements conséquents de temps et d’argent (création de la solution, durée de création, maintenance continue…)

Magento permet d’ajouter des plugins et extensions (Third Party Services) qu’il est possible d’obtenir sur une marketplace consacrée à ces derniers. Par exemple, il sera possible de connecter un système bancaire à notre ***CMS*** très simplement.

Magento permet entre autres :

* **une adaptation des outils d’administration** aux besoins des utilisateurs (tant côté entreprise que côté client)
* **une gestion de catalogue complète**
* **une gestion multi-boutique**
* **l’accès à de nombreux outils marketing et de facturation**
* **une gestion des paiements simplifiée**

Avec une solution clé en main comme Magento, il sera possible de gérer les prix des articles, les stocks, les promotions, la facturation et de manière générale, tous les aspects attenants à la bonne tenue d’un site e-commerce. Les nombreux plugins et extensions permettront d’adapter facilement les fonctionnalités du site aux besoins d’OCPizza.

## Site web

Afin de répondre aux besoins d’OCPizza tout en respectant le budget, nous proposons ici de développer un site web responsive. Cela nous permet de maîtriser le budget tout en ayant une solution adaptée pour tous les appareils (ordinateur, smartphones iOS & Android, tablettes).

La conception du front-end pourra être faite à travers une large gamme de thèmes préconçus développés en ***HTML5, CSS3*** & ***Javascript***, qui sont les technologies les plus populaires utilisées pour les sites web. Les prix de ces derniers sont compris entre 100 et 500$ et permettent une mise en place du front-end rapide et peu onéreuse.

Il sera toutefois possible de modifier le code ***CSS*** pour personnaliser certains aspects comme les couleurs ou les typographies afin de se conformer à la charte graphique.

Magento et les modules proposés sont écrits en ***PHP***, et la base de données à laquelle nous serons connectés sera une ***base de données relationnelle*** MySQL.

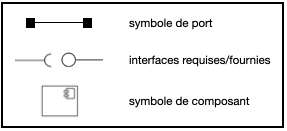
## Diagramme de composants

Le diagramme de composants ci-dessous permet d’**appréhender la structure du système** et d’**illustrer la relation entre les différents composants**.

Il permet notamment :

* d’imaginer la structure physique du système
* d’exposer les composants du système et leurs relations
* mettre en évidence le comportement de chaque service avec l’interface.

# 

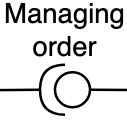


## Composants généraux

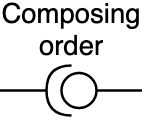
Dans le diagramme de composants exposé dans la section précédente, on peut voir les composants répartis dans Front-end, Back-end, Database et Payment system.

Ces composants sont interchangeables. Par exemple, il sera possible de changer de module de paiement pour un autre sans impacter le reste de la structure.

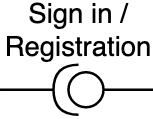
A titre d’exemple, le “Payment module” Magento aura nécessairement besoin d’un composant “API” fourni (dans notre cas, Stripe) pour pouvoir compléter l’action “Processing payment”.



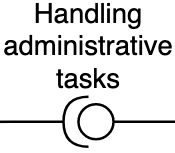
Le composant Order handling communique avec le composant Order service dans le cadre de la gestion de commande.



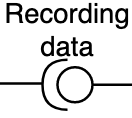
Dans un contexte de composition de commande, une commande créée par un utilisateur via le composant Order creation sera traitée par Order Service. Composing order doit gérer la persistance avec la base de données.



L’accès aux composants présents dans le front-end nécessite une connexion ou une inscription préalable gérée grâce au composant Authentification du back-end.



Le composant Administrative tasks communiquera avec le composant Administrative space afin de gérer les tâches administratives (ce dernier communiquera également avec le SGBD)



De manière générale, tous les composants du back-end communiqueront avec le SGBD afin de récupérer les données nécessaires.

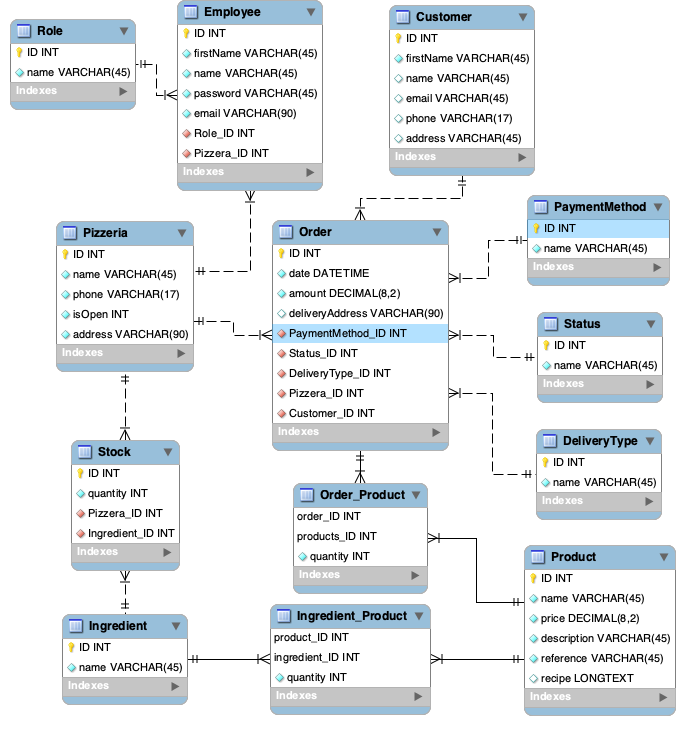
Le composant Payment module va communiquer avec l’API de Payment system dans le cadre de la gestion du paiement. C’est le composant Payment module qui va faire la demande de paiement au composant Payment system via une requête API. Le Payment system se chargera de contacter la banque et de transmettre au Payment module le statut du paiement (accepté ou refusé). Ce dernier passe ensuite la réponse au composant Order service.

# Architecture de Déploiement

## Modèle physique de données

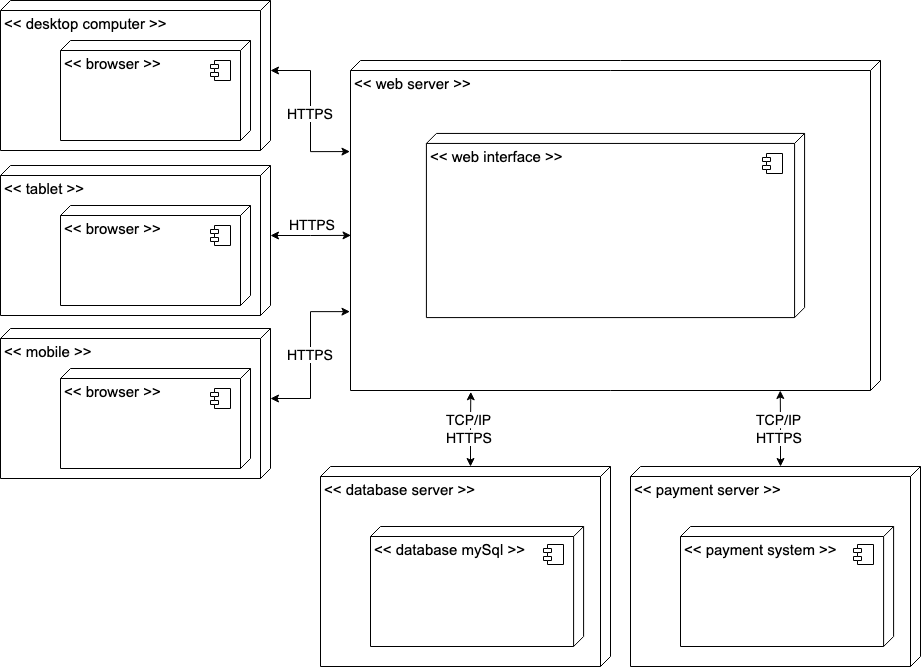
Le modèle physique de données est élaboré en se basant sur le diagramme de classes. C’est une représentation graphique d’une base de données. On y retrouve les propriétés avec des types adaptés, et il introduit les notions de clé primaire et clé étrangère. Les tables ont des relations identifiantes (lignes pleines) ou non identifiantes (lignes pointillées) entre elles, et possèdent des clés primaires, la plupart du temps auto-incrémentées (ici, les propriétés “ID”).

Le rôle de clé primaire peut être endossé par un attribut de la table même si la création d’un ID à part entière est une possibilité fiable. La clé primaire d’une table intermédiaire comme “Order\_Product” est une clé primaire composée. Elle est composée des clés étrangères de “Order” et “Product”.



## Diagramme de déploiement

Le diagramme de déploiement décrit le déploiement physique des informations générées par le logiciel sur des composants matériels, et permet de montrer la disposition des ressources matérielles.



### 

## Serveur web

Nous utiliserons la solution Amazon : ***AWS - Lightsail***. Ainsi, les coûts liés au serveur seront maîtrisés. Il s’agit d’une solution en cloud ce qui indique que la gestion des serveurs est effectuée directement par Amazon. Par ailleurs, ce service est très simple de prise en main et propose énormément de solutions, certaines étant spécifiques aux bases de données.

## Serveur de base de données

Le serveur de base de données ***Amazon RDS*** a été choisi pour sa réputation et sa fiabilité et parce qu’il communique avec le système de gestion de base de données (SGBD) MySQL. Le modèle physique de données a été réalisé avec ***MySQL Workbench***, et les données pourront être gérées depuis le backend de l’application web.

# Architecture logicielle

## Principes généraux

La solution retenue est le CMS E-commerce open source Magento. C’est une solution clé en main qui permet un déploiement grâce à une configuration simple, et dont l’architecture est prédéfinie.

Seuls le thème du site et les fichiers qui concernent le frontend pourront être modifiés.

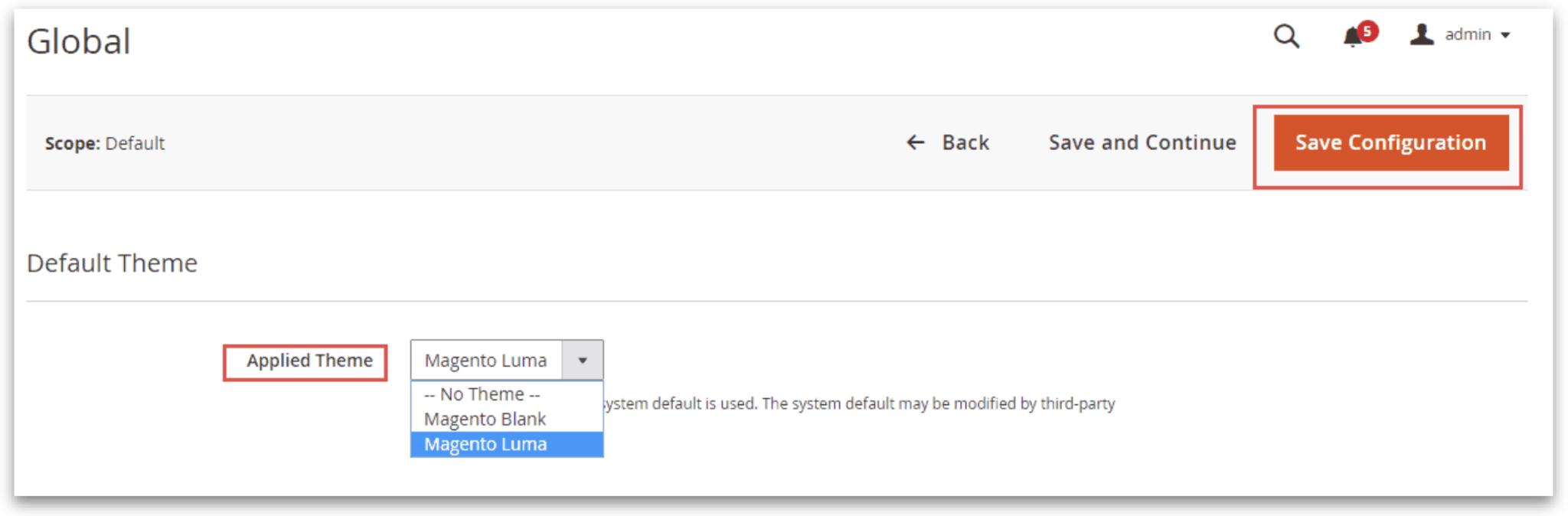
C’est au moment de l’installation du CMS qu’il faudra le connecter à la base de données et notamment au serveur sur lequel celle-ci sera hébergée, puis configurer le reste du système, comme le nom de domaine rattaché, les comptes admins, etc.

## Infrastructure CMS

Il sera possible d’installer un thème pour le site, de nombreux templates sont disponibles en ligne, notamment sur le marketplace de Magento ou encore sur Template Monster ou ThemeForest.

Il faudra ensuite pour installer ce thème :

1. le télécharger
2. l’installer sur la boutique en ligne en passant par le *FTP* : dézipper l’archive puis uploader les dossiers « app » et « skin » à la racine du site.
3. définir le thème depuis le tableau de bord en se rendant dans système > configuration > design > éditer un thème > choisir le thème dans « thème appliqué » puis sauvegarder



Par ailleurs, il sera possible d’apporter des modifications à votre template (typos couleurs…) en modifiant directement les feuilles de style CSS.

Pour cela, il faudra accéder au dossier Theme ( *<THEME\_DIR>* ) dans lequel sont référencés les feuilles de styles modifiables :

* */<Namespace>\_<Module>/web/css* : style spécifiques aux modules
* */web/css* : contient
  + *\_styles.less :* fichier composant, inclus tous les fichiers less utilisés dans le theme
  + *styles-m.less:* permet de générer les styles spécifiques au mobile, inclus \_styles.less
  + *style-l.less :* permet de générer les styles spécifiques au desktop, inclus \_styles.less
  + */source :* sous fichier qui contient les fichier de configuration Less faisant appels à la bibliothèque Magento UI
  + */source/\_theme.less :* override les valeurs des variables de la bibliothèque Magento UI

# Points particuliers

## Gestion des logs

Les logs permettent d’identifier les raisons et/ou l’origine d’une panne informatique. Ils sont également très utilisés afin de veiller à la sécurité des sites web et identifier rapidement une menace en cas d’attaque.

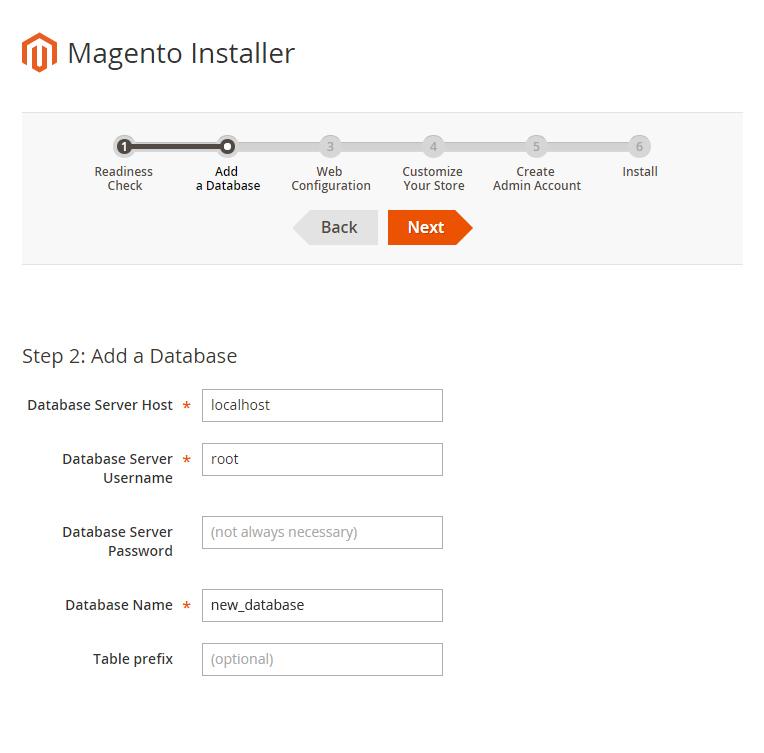
Avec Magento, il sera possible de créer, déployer et manager son propre fichier de logs. Toute une partie de la documentation Magento explique la marche à suivre pour les déployer :

<https://devdocs.magento.com/cloud/project/log-locations.html>

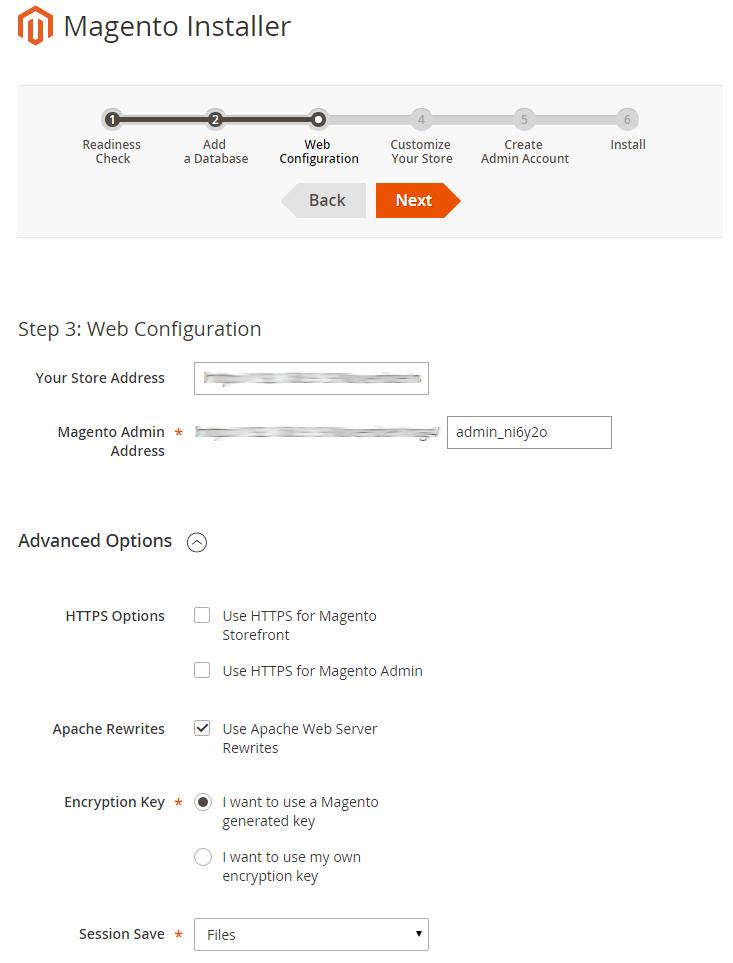
## Fichiers de configuration

### Installation de Magento

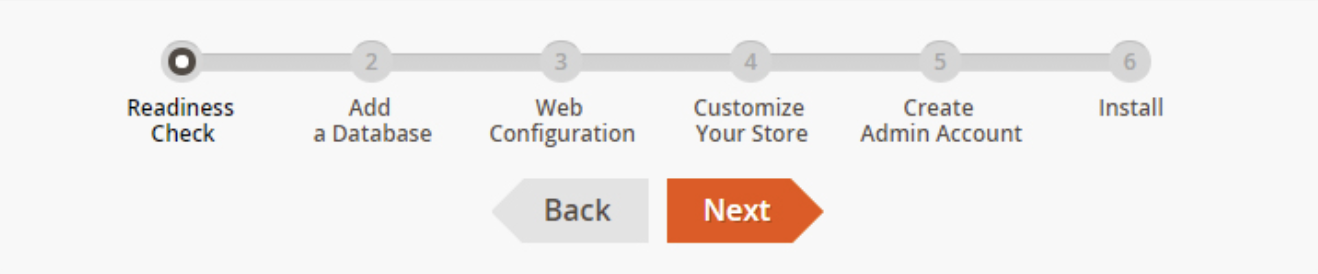
Nous utiliserons le CMS Magento qui est une solution clé en main. Il faudra dans un premier temps configurer la base de données qui est obligatoire pour l’installation d’un site e-commerce Magento.



Il faudra par la suite renseigner sur Magento les informations de base pour configurer le système : nom de domaine rattaché, information sur la base de données, configuration des paramètres et des conditions de connexions.



Toutes ces étapes sont expliquées pas à pas sur la plateforme lors de son installation via une interface graphique.



### Module de paiement

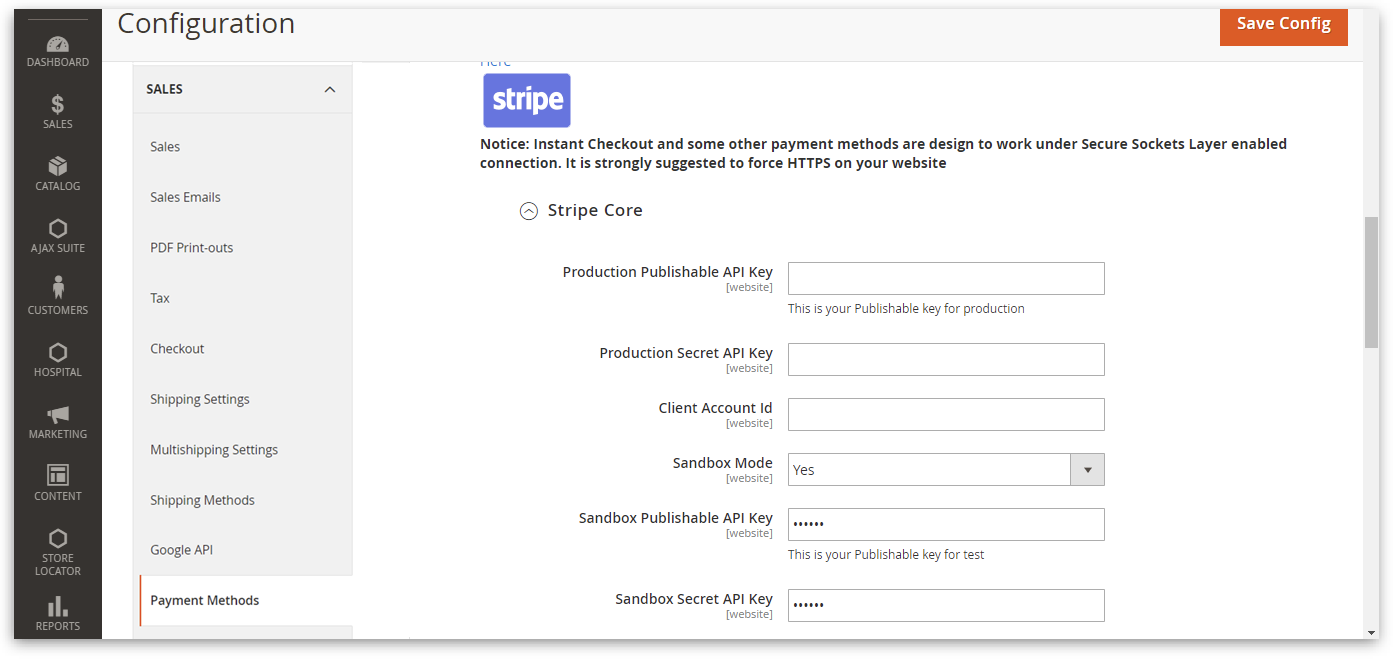
La plateforme utilisée pour la gestion des paiements en ligne sera Stripe qui compte parmi les plus utilisées. Réputé pour sa fiabilité, Stripe propose des options d’intégration flexibles.

Stripe pourra se connecter à Magento via une API. Pour cela, il faudra se rendre sur le back office > configuration > payment methods > Stripe puis renseigner les informations demandées. Si besoin, se référer à la documentation Stripe :

https://stripe.com/docs/plugins/magento-2.

Pour plus d’informations, se référer à la documentation officielle ci-dessous :

* <https://stripe.com/docs/plugins/magento-2>



## Ressources

Ci-dessous, différents liens redirigeant vers les pages de documentation et/ou des supports techniques disponibles :

**Magento *:***

* <https://devdocs.magento.com/>
* <https://support.magento.com/>

**AmazonWebService :**

* <https://docs.aws.amazon.com/>
* <https://aws.amazon.com/fr/premiumsupport/knowledge-center/>
* [https://aws.amazon.com/fr/whitepapers/](https://aws.amazon.com/fr/whitepapers/?whitepapers-main.sort-by=item.additionalFields.sortDate&whitepapers-main.sort-order=desc&awsf.whitepapers-content-type=*all&awsf.whitepapers-tech-category=*all&awsf.whitepapers-industries=*all&awsf.whitepapers-business-category=*all&awsf.whitepapers-global-methodology=*all)
* https://docs.aws.amazon.com/rds/index.html
* <https://forums.aws.amazon.com/index.jspa>

**Stripe :**

* <https://support.stripe.com/>
* https://stripe.com/docs

## Environnement de développement

Magento est une plateforme ***open-source***, développée à l'aide du langage de programmation ***PHP*** et du framework ***Laminas***.

L’environnement de développement du site web sera l’*IDE Sublime text* qui permet de rédiger du code en *PHP,* en important le *framework* *Laminas* (Version 3.0.0)*.*

Le front end sera développé en ***HTML5***, ***CSS3*** et ***JavaScript***.

## Procédure de packaging / livraison

Tous les fichiers de configuration pour déployer le site sur les serveurs web AWS , ainsi que la base de données à déployer sur les serveurs AWS - RDS se trouvent sur le *repository* *GitHub* suivant : github.com/petitsuix/ocpizza.

Le compte super-admin a été créé pour le directeur général d’OCPizza ainsi que les comptes des différentes pizzerias déjà existantes.

Les accès à Magento, AWS et Stripe seront fournis via l’application sécurisée de gestion de mots de passes “Keeper”.

# Glossaire

| **CMS** | Le système de gestion de contenu (SGC) ou Content Management System (CMS) regroupe une catégorie de logiciels qui permettent de concevoir, gérer et mettre à jour des sites Web ou des applications mobiles de manière dynamique et de gérer séparément la forme et le contenu. |
| --- | --- |
| **Diagramme UML** | Un diagramme UML (Unified Modeling Language) fournit une représentation visuelle d'un aspect d'un système. Ils illustrent les aspects quantifiables d'un système qui peuvent être décrits visuellement, tels que les relations, le comportement, la structure ou la fonctionnalité. |
| **HTML** | L'HyperText Markup Language, HTML, désigne un type de langage informatique descriptif. Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web. |
| **CSS** | Acronyme pour « Cascading Style Sheets » ce qui signifie « feuille de style en cascade ». Le CSS correspond à un langage informatique permettant de mettre en forme des pages web (HTML ou XML). Ce langage est donc composé des fameuses « feuilles de style en cascade » également appelées fichiers CSS**.** |
| **JavaScript** | Langage de programmation utilisé pour créer des effets d’animation pouvant être interactifs dans les navigateurs web. |
| **From scratch** | Créer quelque chose en partant de rien. Équivalent anglais de l'expression latine Ex nihilo. Peut s'appliquer dans beaucoup de domaines tels que la cuisine (…) ou le développement informatique. |
| **MOE/MOA** | Maîtrise d’ouvrage (MOA) : commandite le projet (client)  Maîtrise d’oeuvre (MOE) : répond aux demandes de la MOA |
| **Open source** | La désignation open source ou code source ouvert, s'applique aux logiciels dont la licence respecte des critères précisément établis par l'Open Source Initiative, c'est-à-dire les possibilités de libre redistribution, d'accès au code source et de création de travaux dérivés. |
| **PHP** | Hypertext Preprocessor, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, |
| **Base de données**  **relationnelle** | Avec une base de données relationnelle, vos données sont divisées en plusieurs zones de stockage séparées – appelées tables – plutôt que de tout regrouper dans une seule grande unité de stockage. |
| **Laminas** | Anciennement Zend Framework, le projet a été créé dans le but de simplifier le développement web, tout en recommandant les bonnes pratiques et la conception orientée objet en offrant des outils aux développeurs. |
| **MySQL Workbench** | MySQL Workbench est un logiciel de gestion et d'administration de bases de données MySQL |
| **Amazon RDS** | Amazon Relational Database Service est un service de base de données relationnelle distribué par Amazon Web Services. |
| **AWS - Lightsail** | Amazon Lightsail offre des instances de serveur privé virtuel (VPS), des conteneurs, du stockage, des bases de données |