



OC Pizza

Nouveau Système Informatique

Dossier d'exploitation

Version 1.0

Auteur

Johann Ulma

Analyste Programmeur

**IT Consulting &
Development**
www.itconsulting.fr

Les Arcades du Palais – 5 rue Jean Jaurès 37000 Tours - 0261918821 – support@itconsulting.fr

S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Tours – SIREN 999 999 999 – Code APE :
6202A



TABLE DES MATIERES

1 - Versions	5
2 - Introduction	6
2.1 - Objet du document.....	6
2.2 - Références	6
3 - Pré-requis	7
3.1 - Système.....	7
3.1.1 - Caractéristiques techniques	7
3.1.2 - Serveur de Base de données	7
3.1.3 - Serveur Web.....	7
3.1.3.1 - Ouverture des ports dans le Pare-feu	7
3.1.3.2 - Le nom de domaine	8
3.1.3.3 - Packages nécessaires aux serveurs.....	8
3.1.3.4 - Packages gestion du serveur	8
3.1.4 - Application Mobile	9
3.2 - Bases de données	9
3.3 - Web-services	10
4 - Procédure de déploiement	11
4.1 - Déploiement du site web	11
4.1.1 - Connexion avec Filezilla.....	11
4.1.2 - Installation du site-web	12
4.1.3 - Configuration Apache	13
4.1.3.1 - Fichier de configuration.....	13
4.1.3.2 - Sécuriser Apache	13
4.1.3.3 - Réglage des droits	14
4.1.4 - Vérifications	14
4.1.5 - Certificat SSL	14
4.2 - Déploiement de la base de données	15
4.2.1 - Connexion avec Filezilla.....	15
4.2.2 - Procédure MySQL	15
4.2.3 - Vérifications	17
4.3 - Déploiement de l'application iOS	17
4.3.1 - Enregistrement des appareil iOS	17
4.3.2 - Créer un provisioning profile iOS Ad Hoc	18
4.3.2.1 - Déclarer un certificat	18
4.3.2.2 - Créer le CSR	19
4.3.2.3 - Télécharger le certificat.....	20
4.3.2.4 - Créer l'App ID et le Bundle ID pour iOS	21
4.3.2.5 - Générer le provisioning profile iOS.....	23
4.3.2.6 - Attribution des appareils.....	24
5 - Procédure de démarrage / arrêt.....	25
5.1 - Serveur.....	25
5.2 - Base de données.....	25
5.3 - Serveur web	25
5.4 - Application	26

6 - Procédure de mise à jour	27
6.1 - Base de données.....	27
6.1.1.1 - Méthode : Ligne de commande	27
6.1.1.2 - Méthode : phpMyAdmin	27
6.1.1.3 - Mise à jour.....	28
6.2 - Serveur web	28
6.2.1 - Archive du Site.....	28
6.2.2 - Nouvelles données pour le site.....	29
6.2.3 - Mise à jour du service	29
6.3 - Application	29
7 - Supervision/Monitoring	30
7.1 - Supervision des serveurs	30
7.2 - Outil de monitoring	31
8 - Procédure de sauvegarde et restauration.....	32
8.1.1 - Le Snapshot.....	32
8.1.2 - Git.....	33
8.1.3 - Le serveur de sauvegarde	33
9 - Glossaire.....	35

1 - VERSIONS

Auteur	Date	Description	Version
J. Ulma	18/05/2021	Création du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application OC Pizza.

Objectif du document :

Définir les différentes procédures de déploiement liées à l'application en incluant la partie maintenance.

Nous passerons aussi en revue les prérequis de mise en œuvre.

Les éléments du présent dossier découlent :

1. Du recueil des besoins et du cahier des charges donné par le client
2. De la présentation de la solution technique avec le Chef de projet
3. Du dossier fonctionnel
4. Du dossier technique

2.2 - Références

Pour de plus amples informations, se référer :

1. **DCF – 1.0** : Dossier de conception fonctionnelle de l'application
2. **DCT – 1.0** : Dossier de conception technique de l'application

3 - PRE-REQUIS

3.1 - Système

Le système sera composé de 3 parties, deux serveurs l'un pour le site web et l'autre pour la base de données et enfin notre application iOS.

3.1.1 - Caractéristiques techniques

Les caractéristiques sont les mêmes pour les deux serveurs à savoir :

Processeur : 2 CPU Xeon de 2.5 Ghz (12 Core, 48 Thread)

RAM : 6 Go RAM

Stockage : 100 Go SSD

OS : Debian 10.9 minimum

3.1.2 - Serveur de Base de données

Le serveur de base de données sera virtualisé et hébergé chez Infomaniak avec un SGBD MySql en version 8.0.25 minimum.

Packages du serveur : Apache, MySQL, phpMyAdmin, Filezilla, OpenSSH, FTP.

Le schéma dans sa dernière version sera déployé sur ce serveur, son Modèle Physique de Données est fourni dans le dossier technique.

3.1.3 - Serveur Web

Le serveur Web sera lui aussi virtualisé et hébergé chez Infomaniak. Le serveur HTTP sera basé sur Apache en version 2.4.46 minimum.

Packages du serveur : Apache, Python, Django, AngularJS, Filezilla, OpenSSH, FTP.

Les ports nécessaires sont déjà ouverts, cependant en cas de besoin il faudra procéder selon la section 3.1.3.1

3.1.3.1 - Ouverture des ports dans le Pare-feu

Depuis l'interface web Infomaniak, il faut ouvrir les ports du pare-feu afin de permettre les connexions externes.

On se rend dans **Manager** → **Serveur Cloud** → Sélection du serveur cible

Puis dans le menu latéral de gauche **Outils Avancés** → **Ouverture de ports** → **Ouvrir un port**

Un menu contextuel s'affiche, il suffit de le suivre. Les informations à renseigner sont les suivantes :

<i>Le type de connexion</i>	: Entrante, Sortante
<i>L'IP ou Hôte cible</i>	: Celle du serveur ou IP Externe
<i>Le port</i>	: Numéro du port
<i>Le protocole</i>	: Si besoin de le remplacer

3.1.3.2 - Le nom de domaine

Le nom de domaine ocpizza.fr a été réservé en amont chez l'hébergeur.

3.1.3.3 - Packages nécessaires aux serveurs

Le serveur est livré déjà paramétré, cependant un ensemble de services est nécessaire pour le bon fonctionnement du serveur Web

Les packages nécessaires sont listés ci-dessous avec le lien des procédures nécessaires à leur installation :

Apache :	https://linuxize.com/post/how-to-install-apache-on-debian-10/
Python :	https://linuxize.com/post/how-to-install-python-3-9-on-debian-10/
Django :	https://linuxize.com/post/how-to-install-django-on-debian-9/
AngularJS :	https://www.howtoforge.com/how-to-install-angular-on-ubuntu-2004/
MySQL :	https://linuxize.com/post/how-to-install-mysql-on-debian-10/
phpMyAdmin :	https://www.asweb.ca/2017/10/28/comment-faire-pour-installer-phpmyadmin-sur-debian-9/
OpenSSH :	https://www.linuxtricks.fr/wiki/ssh-installer-et-configurer-un-serveur-ssh
Filezilla :	https://linuxways.net/debian/how-to-install-filezilla-on-debian-10/
FTP :	https://guide.ubuntu-fr.org/server/ftp-server.html

3.1.3.4 - Packages gestion du serveur

Afin de renforcer la sécurité de nos serveurs mais aussi de surveiller les performances les packages suivants seront déployés :

Fail2ban : Outil IDS/IPS basé sur les logs, il permet de détecter certaines intrusions et ainsi déclencher des actions d'avertissement ou de protection (envoyer un mail, bannir une ip, etc.)

<https://linuxize.com/post/install-configure-fail2ban-on-debian-10/>

DenyHost : Comme Fail2ban il se base sur les logs mais se concentre sur les attaques bruteforce sur le port SSH

<https://www.digitalocean.com/community/tutorials/how-to-install-denyhosts-on-ubuntu-12-04>

Grafana : Plate-forme open-source spécialisé dans l'analyse et la surveillance. Il fonctionne spécifiquement sur des données chronologiques provenant de sources telles que Promtail et Loki.

<https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-grafana-on-ubuntu-20-04-fr>

Promtail : C'est un agent qui expédie le contenu des journaux locaux vers une instance Loki privée ou Grafana Cloud. Il est généralement déployé sur chaque machine sur laquelle des applications doivent être surveillées.

<https://www.aukfood.fr/collecter-et-afficher-les-logs-avec-grafana-loki/>

Loki : C'est un système d'agrégation de journaux scalable horizontalement, hautement disponible et inspiré de Prometheus.

Loki va s'occuper de la centralisation des logs.

<https://www.aukfood.fr/collecter-et-afficher-les-logs-avec-grafana-loki/>

3.1.4 - Application Mobile

L'application mobile sera codée en Swift 5.4 et la connexion se fera au back-end du serveur Web.

Les API seront utilisées au sein de l'application, le détail se trouve en section 3.3

3.2 - Bases de données

Les bases de données et schémas suivants doivent être accessibles et à jour :

- **Script_oc_pizza.sql** : version 1.0 qui sera le schéma de création de la base de données
- **Datas_oc_pizza.sql** : version 1.0 qui va contenir les données de la pizzeria
- **Dump_oc_pizza.sql** : version 1.0 qui va contenir le modèle et ses données

Un compte ayant les droits nécessaires pour la mise en place de la base de données et l'injection des données.

3.3 - Web-services

Les web services suivants doivent être accessibles et à jour :

- **Google Maps API** : Google Maps sera utilisé sur le site pour afficher une carte des différents points de vente, mais aussi pour la partie Géolocalisation de l'application afin de fournir un guidage aux livreurs.

Documentation générale :

<https://developers.google.com/maps/documentation?hl=fr>

Documentation technique de l'API :

https://cloud.google.com/docs/authentication/api-keys?hl=FR&visit_id=637016438679695018-2120839193&rd=1

- **Monext** : En charge du système de paiement, chaque paiement est vérifié par la banque avant sa validation.

Documentation générale :

<https://docs.monext.fr/display/DT/Guides+Payline>

Documentation technique de l'API :

<https://docs.monext.fr/display/DT/API+Reference>

4 - PROCEDURE DE DEPLOIEMENT

4.1 - Déploiement du site web

Le code source de notre site web sera fourni sous le format Archive, il contiendra l'arborescence du site web depuis la racine :

```
var/www/ocpizza/
```

L'archive compressée sera donc nommée :

```
oc_pizza.tar.gz
```

Une seconde archive nommée `oc_pizza_apache.tar.gz` va contenir les fichiers de configurations pour Apache.

Dans section 6.2.1 la méthode création d'une archive est rappelée.

4.1.1 - Connexion avec Filezilla

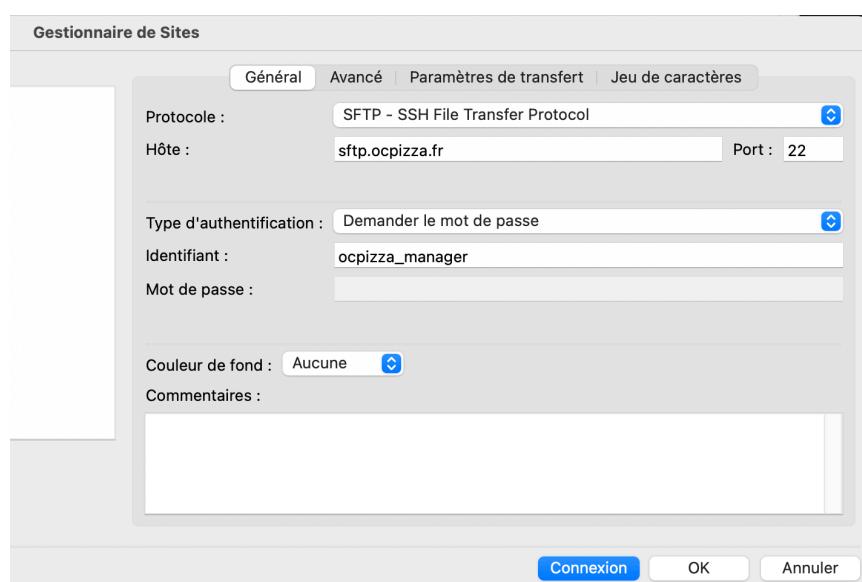
Nous aurons besoin du logiciel Filezilla, il est disponible à cette adresse :

<https://filezilla-project.org/download.php?type=client>

Via Filezilla nous allons procéder au transfert de notre archive vers le serveur, FTP est déjà installé et sécurisé comme expliqué dans les prérequis.

Depuis le Menu allez dans le Gestionnaire de Sites :

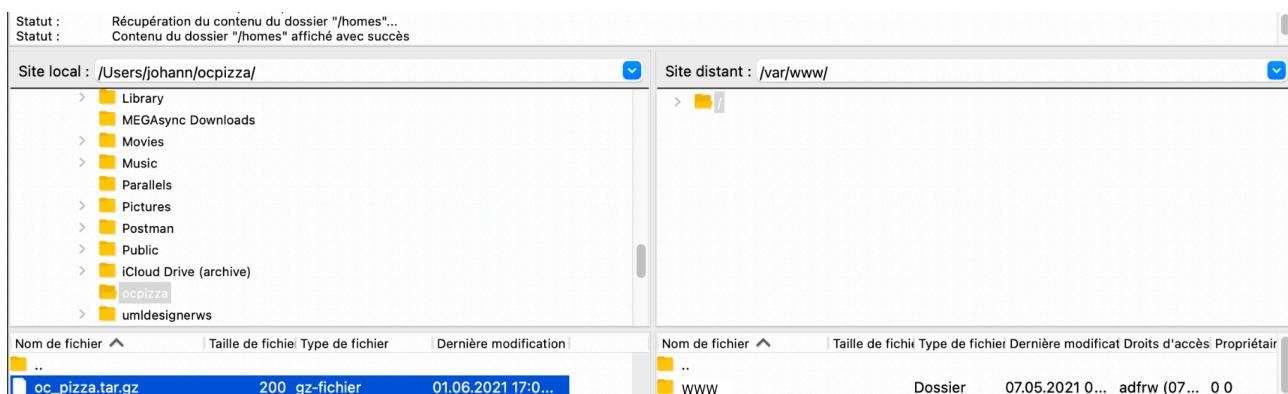
On va créer un **Nouveau Site**



Puis renseigner les informations de notre serveur va héberger le site :

Hôte (adresse du site web) :	<u>sftp.ocpizza.fr</u>
Protocole :	SFTP
Type d'authentification :	Demander le mot de passe
Identifiant :	Un login fournit au client, on ne se connecte pas avec le root
Mot de passe :	Mot de passe associé au login

Une fois la connexion établie, dans **Site local** on sélectionne l'archive **oc_pizza.tar.gz**
 Côté **Site distant** on va dans le répertoire cible qui sera **/var/www/**



Site local : /Users/johann/ocpizza/	Site distant : /var/www/																																	
<ul style="list-style-type: none"> > Library > MEGAsync Downloads > Movies > Music > Parallels > Pictures > Postman > Public > iCloud Drive (archive) > ocpizza > umldesignerws 	<ul style="list-style-type: none"> > .. 																																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Nom de fichier</th> <th>Taille de fichier</th> <th>Type de fichier</th> <th>Dernière modification</th> <th></th> <th>Nom de fichier</th> <th>Taille de fichier</th> <th>Type de fichier</th> <th>Dernière modification</th> <th>Droits d'accès</th> <th>Propriétaire</th> </tr> </thead> <tbody> <tr> <td>..</td> <td>200</td> <td>gz-fichier</td> <td>01.06.2021 17:0...</td> <td></td> <td>..</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>oc_pizza.tar.gz</td> <td></td> <td></td> <td></td> <td></td> <td>WWW</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		Nom de fichier	Taille de fichier	Type de fichier	Dernière modification		Nom de fichier	Taille de fichier	Type de fichier	Dernière modification	Droits d'accès	Propriétaire	..	200	gz-fichier	01.06.2021 17:0...		..						oc_pizza.tar.gz					WWW					
Nom de fichier	Taille de fichier	Type de fichier	Dernière modification		Nom de fichier	Taille de fichier	Type de fichier	Dernière modification	Droits d'accès	Propriétaire																								
..	200	gz-fichier	01.06.2021 17:0...		..																													
oc_pizza.tar.gz					WWW																													

Enfin on fait clic droit et **Envoyer**



La procédure est la même pour l'archive **oc_pizza_apache.tar.gz** cependant le répertoire de destination sera **/tmp**

Une fois les fichiers reçus, on peut passer à l'installation côté serveur.

4.1.2 - Installation du site-web

On se connecte en SSH via un client type :

Putty disponible ici : <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Hyper disponible ici : <https://hyper.is/#installation>

Une fois connecté sur le serveur via le *login / mot de passe* fourni

On se rend dans le répertoire : `cd /var/www`

Et on décomprime l'archive avec la commande : `tar -zxvf oc_pizza.tar.gz`
On vérifie que la décompression est correcte avec la commande : `ls -la`

Lorsque l'opération est finie, on supprime ensuite l'archive avec la commande :

```
rm oc_pizza.tar.gz
```

4.1.3 - Configuration Apache

4.1.3.1 - Fichier de configuration

Une fois l'import effectif, on peut s'occuper de notre archive Apache dans le dossier `/tmp`

Et on décomprime l'archive avec la commande : `tar -zxvf oc_pizza_apache.tar.gz`
Il contient les fichiers du dossier : `/etc/apache2/sites-available/`

Nous déplaçons les fichiers à leur bonne destination via la commande :

```
mv -R /tmp/oc_pizza_apache/sites-available/ /etc/apache2/sites-available/
```

Puis depuis le répertoire : `cd /etc/apache2/sites-available/`
On exécute la commande : `a2ensite ocpizza.fr`

Cette commande va créer les liens symboliques dans le répertoire :

```
/etc/apache2/sites-enabled
```

4.1.3.2 - Sécuriser Apache

Notre dossier `/tmp` va contenir d'autres fichiers de configurations qui ont été préconfiguré pour Apache.

Par ailleurs le fichier `/etc/apache2/sites-available/ocpizza.fr` est déjà sécurisé avec le VirtualHost

Nous procédons ici aussi à leur copie, de la même manière :

```
mv /tmp/«NomDuRépertoire»/«NomDuFichier» /emplacementDuRépertoire/NomDuFichier
```

Les répertoires de destination sont les suivants :

- Paramètres de sécurité d'Apache notamment sur les entêtes HTTP
`/etc/apache2/conf-available/security.conf`
- Fichier de configuration générale :
`/apache2/apache2.conf`

Pour plus d'informations sur la sécurisation d'apache, il faut se référer à la documentation suivante :

https://wiki.debian-fr.xyz/Sécuriser_Apache2#htaccess

4.1.3.3 - Réglage des droits

La dernière étape consiste au changement des droits sur les répertoires :

```
chown -R root:www-data /var/www/ocpizza.fr/
```

De cette manière les droits d'écriture sont limités au compte root

On change le niveau des droits comme ceci :

```
chmod -R 755 /var/www
```

4.1.4 - Vérifications

Une fois les réglages effectifs, on redémarre le serveur apache :

```
sudo service apache2 restart
```

Si le message affiché contient **active (running)**, alors notre serveur est démarré.

```
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled;
  vendor preset: enabled)
    Active: active (running) since Wed 2021-05-30 20:39:39 UTC;
  2min ago
```

On lance la page : ocpizza.fr qui doit afficher le site web

4.1.5 - Certificat SSL

L'installation du certificat SSL se fait depuis le **Manager** Infomaniak → **Serveur Cloud** → Sélection du serveur cible.

La méthode diffère selon le type de certificat :

Choix certificat externe :

Dans **Actions rapides** → **Installer un certificat SSL** → **Gérer** → Importer un certificat s'il s'agit d'un certificat externe

Choix certificat interne :

Dans **Actions rapides** → **Installer un certificat SSL** → **Gérer** → changer de certificat SSL s'il s'agit d'un certificat disponible chez Infomaniak.

Par défaut nous installons un certificat interne *Let's Encrypt* sur le serveur Web qui est gratuit.

4.2 - Déploiement de la base de données

4.2.1 - Connexion avec Filezilla

La procédure reste la même mais pour le serveur de base de données.

Via Filezilla nous allons procéder au transfert de notre dump vers le serveur.

Depuis le Menu allez dans le Gestionnaire de Sites :

On va créer un **Nouveau Site**

Puis renseigner les informations de notre serveur va héberger le site :

Hôte (adresse du site web) :	sftp.databaselogocpizza.fr
Protocole :	SFTP
Type d'authentification :	Demander le mot de passe
Identifiant :	Un login fournit au client, on ne se connecte pas avec le root
Mot de passe :	Mot de passe associé au login

Une fois la connexion établie, dans **Site local** on sélectionne le fichier `dump_oc_pizza.sql`

Côté **Site distant** on va dans le répertoire cible qui sera `/tmp/`

Enfin on fait clic droit sur le fichier et **Envoyer**

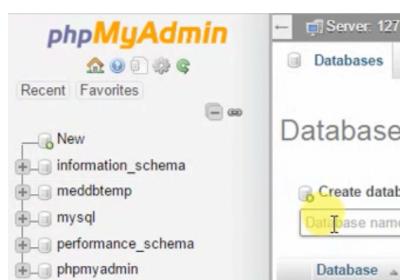
Une fois le fichier reçu, on peut passer à l'installation côté serveur.

4.2.2 - Procédure MySQL

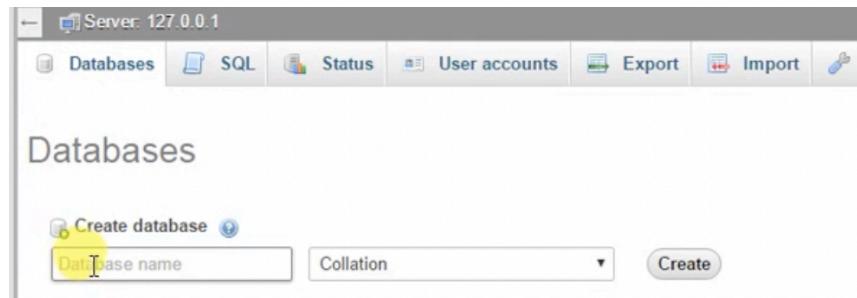
Ayant installé phpMyAdmin sur le serveur, nous allons l'utiliser pour l'import de notre dump.

La procédure guidée est détaillée si dessous.

On sélectionne l'onglet **New** dans le menu de gauche



Cela va créer une nouvelle base de données, on nomme notre base de données qui sera **oc_pizza**



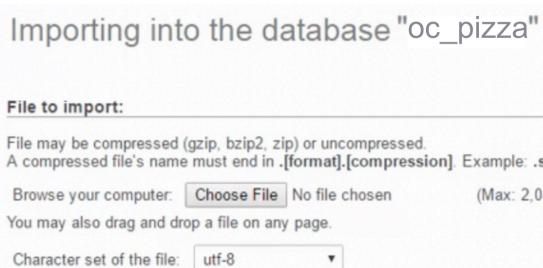
Une fois la base créée, on retourne sur le menu de gauche.

On sélectionne la base de données nouvellement créée, sa structure est actuellement vide.

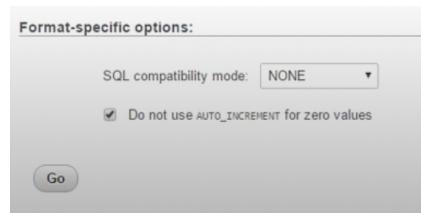
On clic sur l'onglet **Import**



On fait un clic sur **Choose File** et on sélectionne alors notre **dump_oc_pizza.sql**



On défile en bas de page puis on clic sur le **Go**



Dès que l'import est finalisé, une fenêtre confirme cela :

Import has been successfully finished, 23 queries executed. (dump_oc_pizza.sql)

4.2.3 - Vérifications

On vérifie que le serveur SQL est bien lancé via la commande :

```
sudo service mysql status
```

Si le message affiché contient **active (running)**, alors notre serveur est démarré.

Toujours dans phpmyadmin, il est possible de voir dans le navigateur de gauche le détail des tables. Mais aussi de vérifier via une requête SQL pour cela on se rend dans l'onglet SQL de notre base de données et tester avec la commande suivante :

```
SHOW TABLES FROM oc_pizza;
```

Qui va afficher l'ensemble des tables de notre base de données **oc_pizza**

Cependant si aucune erreur n'est reportée lors de l'import, celui-ci sera validé.

4.3 - Déploiement de l'application iOS

Concernant notre application elle sera déployée en mode **Ad Hoc** afin d'être accessible uniquement au personnel de la pizzeria.

Le mode « **Provisioning Profile Ad Hoc** » permet la diffusion d'applications auprès d'une flotte d'appareils enregistrés.

Pour ce faire, il faut disposer d'un compte Apple Developer

Il est disponible à cette adresse : <https://developer.apple.com>

4.3.1 - Enregistrement des appareil iOS

On doit enregistrer tous les appareils iOS dans l'espace Apple Developer en rentrant les identifiants UDID de chaque appareil.

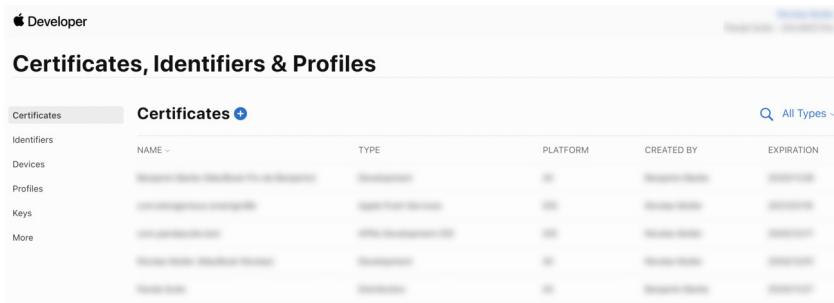
- Enregistrement des appareils iOS
 - Connectez votre appareil iOS à un Mac
 - Ouvrez **iTunes**
 - Cliquez sur l'icône représentant votre appareil
 - Cliquez sur le numéro de série pour afficher le **numéro UDID**. Faites un clic droit pour copier ce numéro
 - Saisissez ce **numéro UDID** dans la section **Devices** de votre interface Apple Developer

4.3.2 - Créer un provisioning profile iOS Ad Hoc

4.3.2.1 - Déclarer un certificat

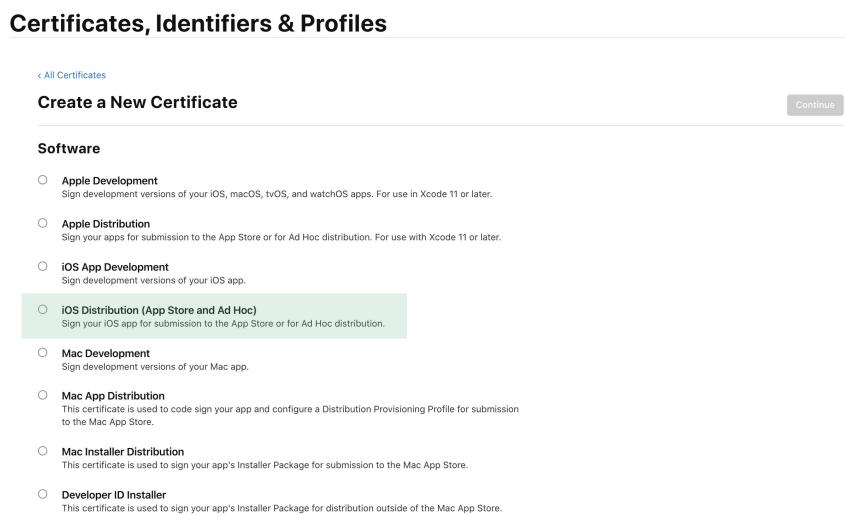
Rendez-vous dans la rubrique **Certificates, IDs & Profiles** de votre espace Développeur, dans le menu à gauche.

Cliquez sur **Certificates**, puis cliquez sur le bouton +



The screenshot shows the 'Certificates' tab selected in the top navigation bar. Below it, there are tabs for 'Identifiers', 'Devices', 'Profiles', 'Keys', and 'More'. A search bar labeled 'All Types' is on the right. The main area displays a table with columns: NAME, TYPE, PLATFORM, CREATED BY, and EXPIRATION. Several rows of certificate entries are visible.

Sélectionnez **iOS Distribution (App Store and Ad Hoc)** dans la partie Software



The screenshot shows the 'Software' section of the certificate creation interface. It lists several options with radio buttons:

- Apple Development**: Sign development versions of your iOS, macOS, tvOS, and watchOS apps. For use in Xcode 11 or later.
- Apple Distribution**: Sign your apps for submission to the App Store or for Ad Hoc distribution. For use with Xcode 11 or later.
- iOS App Development**: Sign development versions of your iOS app.
- iOS Distribution (App Store and Ad Hoc)**: Sign your iOS app for submission to the App Store or for Ad Hoc distribution.
- Mac Development**: Sign development versions of your Mac app.
- Mac App Distribution**: This certificate is used to code sign your app and configure a Distribution Provisioning Profile for submission to the Mac App Store.
- Mac Installer Distribution**: This certificate is used to sign your app's Installer Package for submission to the Mac App Store.
- Developer ID Installer**: This certificate is used to sign your app's Installer Package for distribution outside of the Mac App Store.

A 'Continue' button is visible on the right.

Cliquez sur **Continue** pour arriver sur l'interface pour envoyer votre **CSR (Certificate Signing Request)**.

Certificates, Identifiers & Profiles



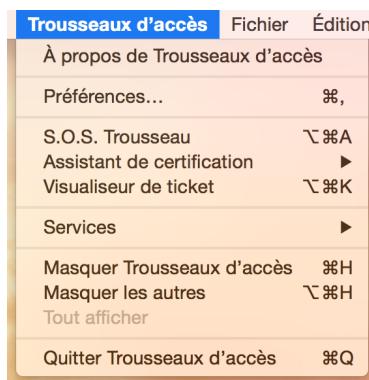
The screenshot shows the 'Create a New Certificate' interface with the 'CSR' tab selected. It includes a note about manually generating a certificate and a 'Choose File' button for uploading the CSR file.

4.3.2.2 - Créer le CSR

Via Mac OS

Rendez-vous dans l'application **Trousseaux d'accès** située dans le dossier **Applications/Utilitaires**.

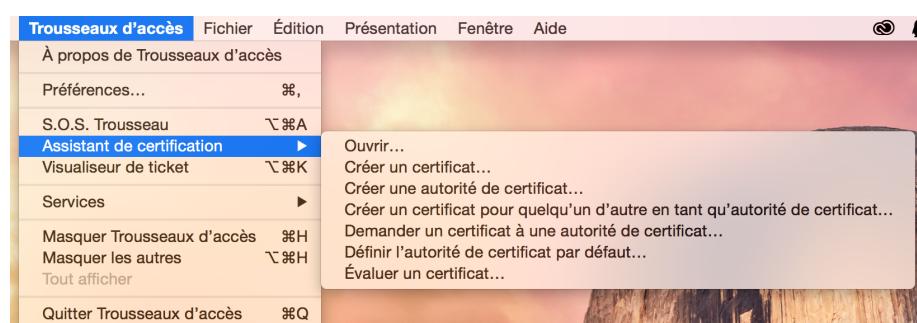
Dans le menu du haut, sélectionnez **Préférences**.



Dans la fenêtre **Préférences**, cliquez sur l'onglet **Certificats**. Désactivez les options **Protocole d'état des certificats en ligne (OCSP)** et **Liste des certificats révoqués (CRL)**. Fermez cette fenêtre.



Dans le menu du haut, cliquez sur **Assistant de Certification > Demander un certificat à une autorité de certificat**.

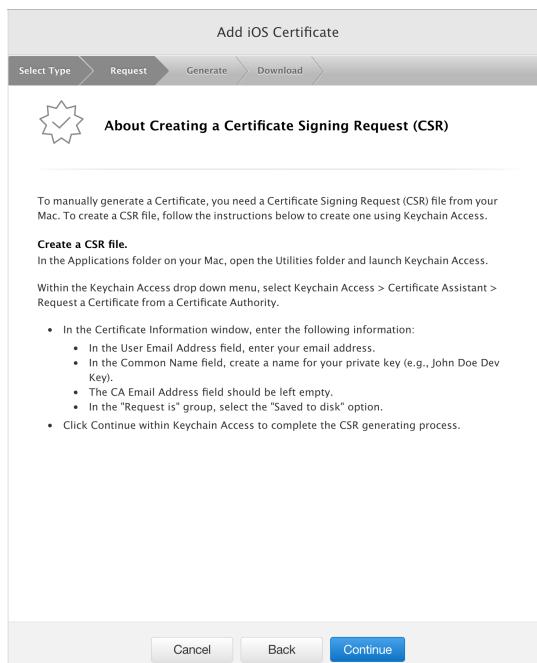


Entrez le nom et l'email associé à votre compte **Apple Developer**. L'adresse e-mail de l'AC n'est pas requise. Sélectionnez **Enregistrée sur disque** et cliquez sur **Continuer**.

Enregistrez ce fichier (*CertificateSigningRequest.certSigningRequest*) sur votre disque dur.

4.3.2.3 - Télécharger le certificat

Une fois que vous avez votre CSR, reprenez votre démarche sur Apple Developer, vous en étiez ici



Cliquez sur **Continuer** et importez votre fichier **CSR** fraîchement créé.

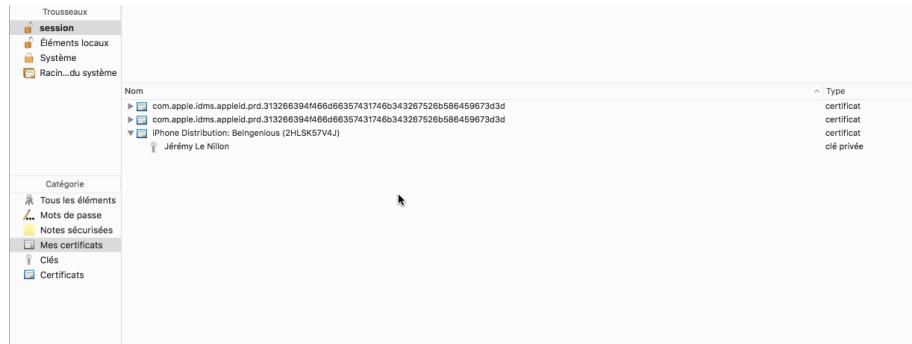
Téléchargez le fichier **iOS Distribution**, c'est votre certificat.

Créer votre fichier **P12**

Ouvrez votre certificat (double clic) et ajoutez-le à votre trousseau d'accès.



Sélectionnez ensuite votre certificat et votre clé privée dans votre trousseau, exportez les en **.p12**.



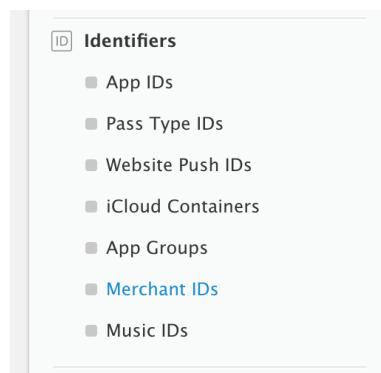
Choisissez votre mot de passe.

4.3.2.4 - Créez l'App ID et le Bundle ID pour iOS

Il s'agit maintenant d'annoncer à Apple ce que votre appli va contenir. C'est à cette étape que vous dressez la carte d'identité de votre application, en quelque sorte.

Cette étape démarre également sur l'espace Développeur d'Apple.

- Rendez-vous dans l'onglet **Certificates, IDs & Profiles** dans le menu de gauche de votre compte Apple Developer
- Dans la catégorie **Identifiers**, cliquez sur **App IDs**



- Cliquez sur le bouton + en haut à droite pour ajouter une application
- Remplissez les champs demandés
- Dans le champ App ID Description, tapez le nom de votre application.

IL NE S'AGIT PAS du nom qui s'affichera pour l'appli dans l'App Store, c'est un identifiant caché qui vous sert de repères

App ID Description

Name:

You cannot use special characters such as @, &, *, ', "

- Gardez bien l'**Apple ID Prefix** (Team ID)
- Laissez la sélection sur **Explicit App ID**.

App ID Suffix

Explicit App ID

If you plan to incorporate app services such as Game Center, In-App Purchase, Data Protection, and iCloud, or want a provisioning profile unique to a single app, you must register an explicit App ID for your app.

To create an explicit App ID, enter a unique string in the Bundle ID field. This string should match the Bundle ID of your app.

Bundle ID:

We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).

- Dans le champ **Bundle ID**, entrez un identifiant unique ayant la forme suivante : com.[nom de votre organisation].[nom de votre appli].
 - Dans la liste des **App Services**, sélectionnez **Associated Domains** (lien universel) et **Push Notifications** si vous souhaitez en envoyer.
- Choisissez uniquement les services que vous utiliserez dans votre application. Cliquez sur **Continuer**.

App Services

Select the services you would like to enable in your app. You can edit your choices after this App ID has been registered.

- Enable Services:
- App Groups
 - Apple Pay Payment Processing
 - Associated Domains
 - Data Protection
 - Complete Protection
 - Protected Unless Open
 - Protected Until First User Authentication
 - Game Center
 - HealthKit
 - HomeKit
 - Hotspot
 - iCloud
 - Compatible with Xcode 5
 - Include CloudKit support (requires Xcode 6)
 - In-App Purchase
 - Inter-App Audio
 - Multipath
 - Network Extensions
 - NFC Tag Reading
 - Personal VPN
 - ClassKit
 - Push Notifications
 - SiriKit
 - Wallet
 - Wireless Accessory Configuration

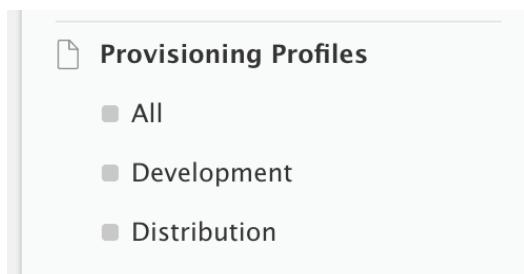
Vous avez déclaré votre application sur Apple.

4.3.2.5 - Générer le provisioning profile iOS

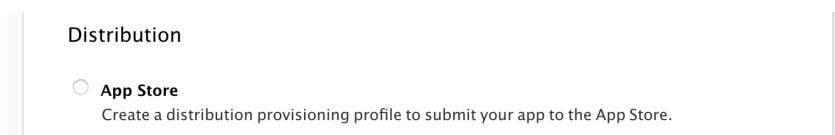
Maintenant que vous avez votre certificat et votre **Bundle ID** pour votre application, vous êtes prêt(e) pour générer votre **provisioning profile** !

Cette étape démarre également sur l'espace Développeur d'Apple.

- Dans la rubrique **Certificates, Identifiers & Profiles**, rendez-vous sur **Provisioning files** dans l'encadré de gauche, en bas.
- Choisissez le sous-espace **Distribution** dans le menu de gauche.



- Cliquez sur +.
- Dans la liste **Distribution**, sélectionnez **App Store**



- Retrouvez l'**App ID** que vous avez créé dans la liste.
- Sélectionnez votre certificat.
- Téléchargez le profil de configuration généré (***.mobileprovision**) et enregistrez-le sur votre ordinateur.

Une fois que ce fichier est prêt, vous pouvez le télécharger dans votre espace personnel afin de générer l'IPA.

4.3.2.6 - Attribution des appareils

A la fin du processus, veuillez choisir les **Devices** (Appareils) qui sont associés à ce certificat. Vous retrouvez ici la liste des appareils précédemment enregistrés. Seuls ces appareils pourront installer l'application.

Votre application est prête à être déployée.

5 - PROCEDURE DE DEMARRAGE / ARRET

Les procédures de démarrage / arrêt vont impacter grandement les accès au site web et à notre application.

Il est donc conseiller d'effectuer ces manœuvres seulement en cas de problème ou en dehors des heures de production des pizzerias.

5.1 - Serveur

Il est possible de redémarrer les serveurs depuis l'espace Infomaniak

1. ouvrir le Manager (manager.infomaniak.com)
2. aller dans Serveur Cloud
3. si nécessaire cliquer sur le serveur concerné
4. cliquer sur Gérer
5. choisir de Redémarrer votre serveur

Il est aussi possible de le faire depuis l'OS après une connexion en SSH au serveur :

Redémarrer : **reboot**

Éteindre : **shutdown -h now**

La commande éteindre, oblige de passer par le manager pour allumer de nouveau le serveur

5.2 - Base de données

On peut interagir directement avec le processus MySQL via ces commandes :

Statut de MySQL : **sudo /etc/init.d/mysql status**

Recharger : **sudo /etc/init.d/mysql reload**

Redémarrer : **sudo /etc/init.d/mysql restart**

Éteindre : **sudo /etc/init.d/mysql stop**

Démarrer : **sudo /etc/init.d/mysql start**

5.3 - Serveur web

Pour le serveur web les commandes sont similaires :

Statut d'Apache : **sudo /etc/init.d/apache2 status**

Recharger : **sudo /etc/init.d/ apache2 reload**

Redémarrer : **`sudo /etc/init.d/ apache2 restart`**

Éteindre : **`sudo /etc/init.d/ apache2 stop`**

Démarrer : **`sudo /etc/init.d/mysql start`**

5.4 - Application

L'application peut être arrêtée ou redémarrée en la quittant.

Si OC Pizza souhaite ne plus utiliser l'application (retirer son accès), elle devra en référer à l'équipe technique avant toute manipulation.

De même si l'application doit évoluer vers un autre modèle de distribution cela ne peut être fait sans l'appui d'une équipe technique.

6 - PROCEDURE DE MISE A JOUR

6.1 - Base de données

Avant tout mise à jour importante de la base de données, il est conseillé de procéder à une sauvegarde :

6.1.1.1 - Méthode : Ligne de commande

Soit depuis la ligne de commande :

```
mysqldump --user=mon_user --password=mon_password --databases nom_de_la_base > fichier_destination.sql
```

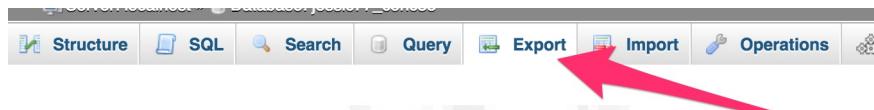
En remplaçant les champs par les informations nécessaires

Exemple :

```
mysqldump --user=ocmanager --password=ceciestmonmdp --databases oc_pizza > backup_oc_pizza.sql
```

6.1.1.2 - Méthode : phpMyAdmin

Depuis phpMyAdmin, en faisant un Export de la base de données



Exporting tables from [REDACTED] database

Export Method:

- Quick - display only the minimal options
- Custom - display all possible options

Format:

SQL

Go

6.1.1.3 - Mise à jour

Si on souhaite changer la base de données, on peut faire un import comme cela est évoqué à la section 4.2.2

Et on redémarre le processus à la fin :

```
sudo /etc/init.d/mysql restart
```

Si on veut mettre à jour la version de MySQL, on va vérifier la version `mysqladmin -V`

Puis aller sur le site de MySQL pour vérifier les dernières versions.

Le repository pour APT est disponible ici : <https://dev.mysql.com/downloads/repo/apt/>

La procédure de mise à jour est la suivante :

```
wget -c https://dev.mysql.com/get/mysql-apt-config_0.8.17-1_all.deb
```

```
sudo dpkg -i mysql-apt-config_0.8.17-1_all.deb
```

```
sudo apt-get update
```

```
sudo apt-get install mysql-server
```

On redémarre avec `sudo /etc/init.d/mysql start`

Et enfin `sudo apt autoremove`

6.2 - Serveur web

Pour le serveur web on procède aussi à une sauvegarde en amont.

6.2.1 - Archive du Site

On va archiver notre dossier ocpizza :

```
tar -zcvf oc_pizza.tar var/www/ocpizza
```

- -z : signifie que l'on compresse l'archive
- -c : signifie créer une archive tar
- -v : signifie afficher le détail des opérations
- -f : signifie assembler l'archive dans un fichier

L'archive compressée sera donc nommée :

```
oc_pizza.tar.gz
```

6.2.2 - Nouvelles données pour le site

Plusieurs méthodologies sont possibles nous avons évoqué la méthode Filezilla en section 4.1.1 mais il peut aussi s'agir de récupérer un repository GitHub pour ce faire nous aurons besoin du paquet Git.

La méthodologie d'installation est disponible à cette adresse :

<https://git-scm.com/book/fr/v2/Démarrage-rapide-Installation-de-Git>

Pour ce faire on va procéder en plusieurs étapes

On se place dans le dossier /tmp avec la commande `cd`

Puis :

```
git clone https://github.com/itconsultinganddevelopment/oc_pizza.git  
cd oc_pizza  
mv -R ocpizza /var/www/
```

Enfin on redémarre apache :

```
sudo /etc/init.d/ apache2 restart
```

6.2.3 - Mise à jour du service

Pour la mise à jour du processus Apache dans sa dernière version, il nous suffit de faire :

<code>sudo apt update</code>	: mise à jour de la liste des paquets
<code>apt list -upgradable</code>	: afficher la liste des paquets avec une mise à jour disponible
<code>sudo apt upgrade</code>	: mise à jour des paquets installés

6.3 - Application

La mise à jour de l'application passera par le Store dédié à l'hébergement de l'application, celle-ci est effectuée par l'équipe technique en accord avec le client.

7 - SUPERVISION/MONITORING

7.1 - Supervision des serveurs

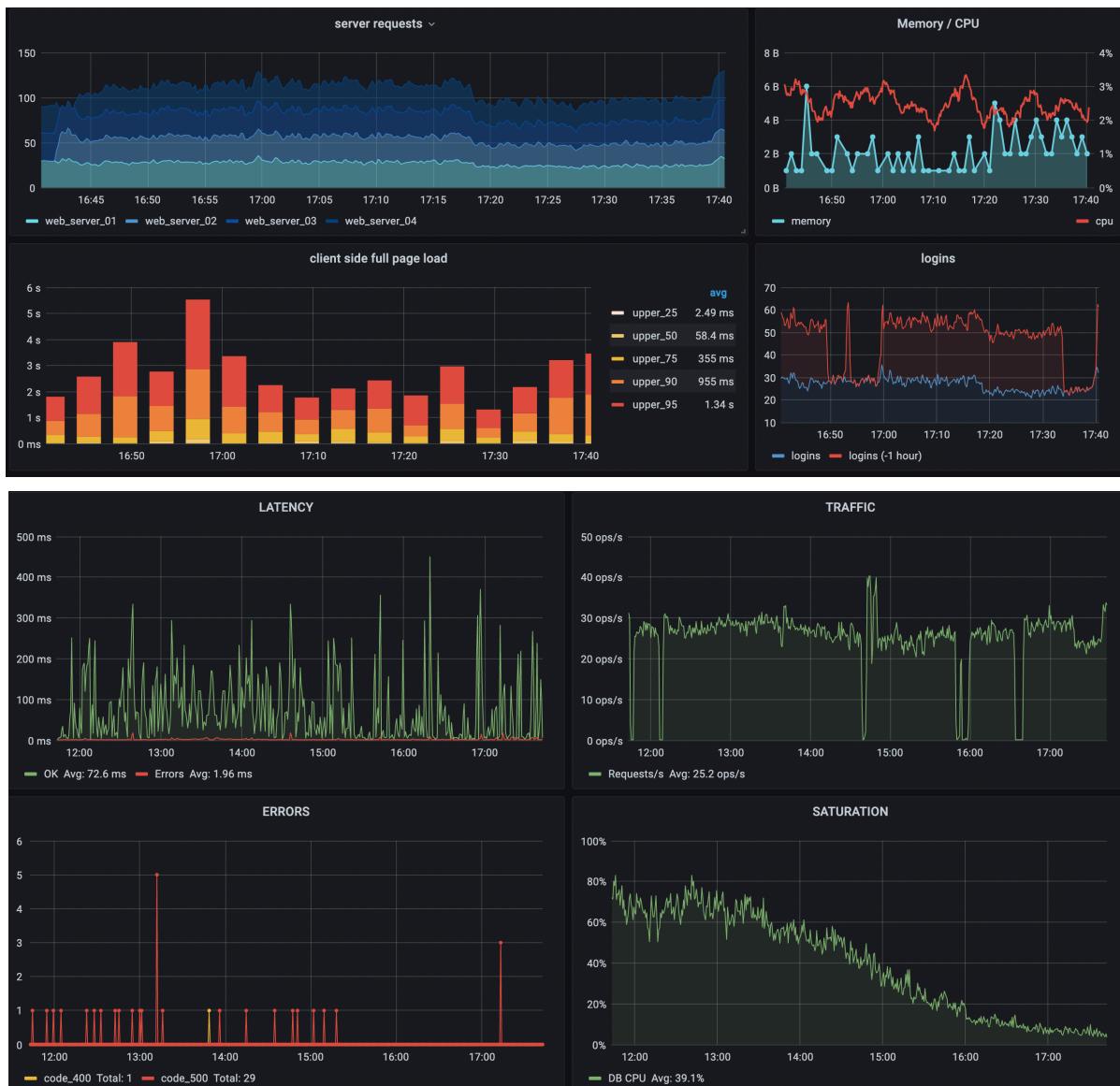
Afin de tester que les serveurs sont toujours fonctionnels nous allons nous appuyer sur Grafana qui permet une centralisation des logs mais aussi le monitoring du serveur.

Il va nous permettre de surveiller le CPU, la RAM, l'usage des disques, la charge réseau et autant d'éléments que nous voudrons grâce à ses plugins disponibles.

La documentation de Grafana est disponible à cette adresse : <https://grafana.com>

Ainsi qu'une Sandbox : <https://play.grafana.org/>

Afin de tester l'applicatif.

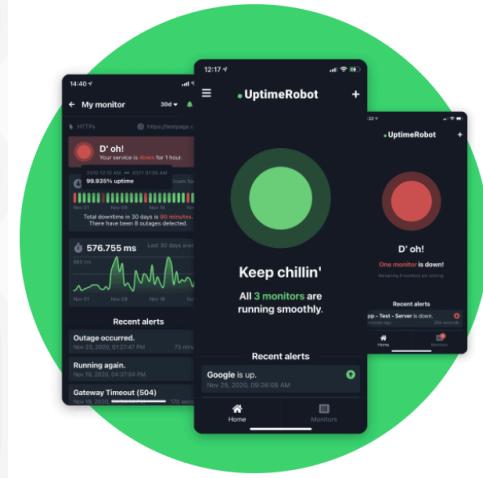
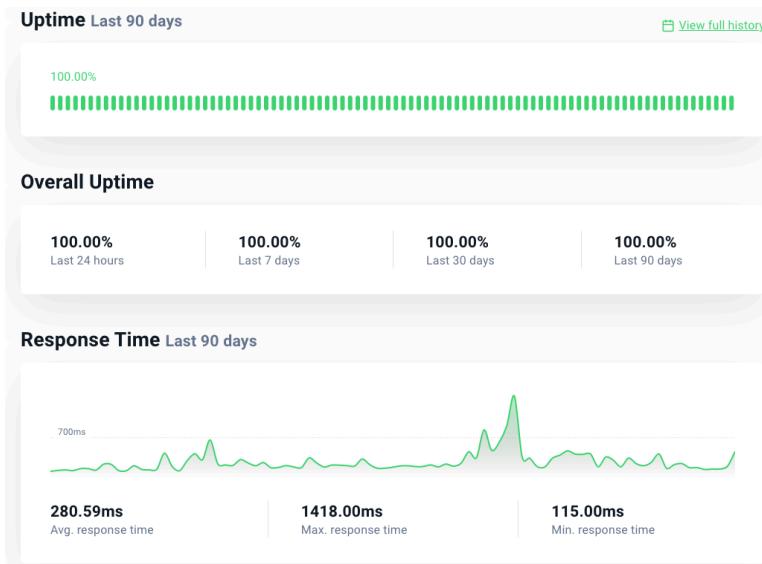


7.2 - Outil de monitoring

Pour plus de sécurité nous recommandons d'avoir un second outil de monitoring qui sera UptimeRobot, il permet de surveiller gratuitement notre site web en faisant un contrôle toutes les 5 minutes.

De cette manière nous sommes certains que nos deux sites restent accessibles.

L'outil est gratuit, il est disponible à cette adresse : <https://uptimerobot.com> et dispose de clients mobiles.



8 - PROCEDURE DE SAUVEGARDE ET RESTAURATION

Concernant les moyens de sauvegarde / restauration nous avons pu voir qu'il y avait des sauvegardes manuelles possibles.

Cependant nous avons d'autres méthodes de sauvegardes disponibles :

8.1.1 - *Le Snapshot*

Deux types de snapshots sont possibles :

- **Snapshot des données** : permet de créer une image du volume qui contient vos données.
- **Snapshot du système d'exploitation** : permet de créer une image du volume qui contient votre système d'exploitation.

Ce type de sauvegarde implique une interruption de quelques minutes de vos applications et sites Internet.

Vous bénéficiez d'un snapshot gratuit pour sauvegarder l'intégralité de votre serveur dans un espace dédié. Il est possible d'en acheter d'autre dans la boutique Infomaniak.

Il faudra donc faire tourner le snapshot de manière régulière.

Nous recommandons le snapshot du système uniquement dans le cadre d'une migration majeure du système : Changement d'OS, de Software.

La meilleure solution sera le snapshot des données avant chaque changement sur les données : fichiers de configuration du site web, nouvelle table de la base de données, ...

La création du snapshot se passe de cette manière :

- 1 - ouvrir le Manager (manager.infomaniak.com)
- 2 - aller dans Serveur Cloud
- 3 - cliquer sur le Serveur non managé concerné
- 4 - cliquer sur Snapshot depuis le menu latéral gauche
- 5 - cliquer sur Créer un snapshot
- 6 - choisir le type de snapshot à créer (données ou système d'exploitation)

8.1.2 - Git

Il est possible d'utiliser Git de manière à sécuriser nos données et les sauvegarder dans un repository GitHub qui sera privatif.

C'est une solution fonctionnelle pour la base de données : Cependant le risque majeur sera la volumétrie de la base de données

Pour cela il faudra automatiser le processus avec un crontab, un article détaille la méthodologie et les risques en termes de volumétrie des données :

<https://wodric.com/backup-base-avec-git/>

8.1.3 - Le serveur de sauvegarde

La solution la plus simple à mettre en œuvre sera la sauvegarde via un outil de backup dédié.

Infomaniak propose ce service via son système Swiss Backup :

Swiss Backup vous permet de sauvegarder et synchroniser facilement tout type de serveurs fonctionnant avec la distribution Linux de votre choix.

Les fonctions sont les suivantes :

- Sauvegarde manuelle grâce au fichier de configuration rClone
- Sauvegarde automatique via Acronis
- Compatible avec CyberDuck (Client FTP)
- Planification mensuelle, hebdomadaire, journalière ou par heure
- Sauvegarde de fichiers ou d'une image du système
- Sauvegarde incrémentielle et différentielle
- Gestion centralisée des sauvegardes

Son tarif est évolutif selon la volumétrie mais pour un plan de sauvegarde des 2 serveurs quotidien lors suivante sera adaptée pour débuter :

—	200 Go	+
—	2 Appareils	+
—	1 an	+
3,75 € / mois		
Soit 45,00 € / année		

Le plan de sauvegarde sera le suivant pour chaque serveur :

Dimanche	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi
Complète	Incrémentielle	Incrémentielle	Incrémentielle	Incrémentielle	Incrémentielle	Incrémentielle

Nous faisons le choix de l'incrémentielle car elle va se focaliser que sur les fichiers ayant subi des modifications tout en faisant profiter d'un espace de stockage plus faible.

Le dimanche la sauvegarde réalisée est une sauvegarde intégrale du système, et du lundi au samedi on copie uniquement les fichiers ayant été modifiés.

De cette manière chaque semaine nous avons une sauvegarde complète du système.

La sauvegarde est conservée 30 jours et elle s'effectue de nuit à 2H00 pour une durée inférieure à une heure mais qui va varier en fonction de la volumétrie des données.

9 - GLOSSAIRE

API	Une API pour « Application Programming Interface », que l'on traduit par interface de programmation d'application. Elle permet d'utiliser les ressources, données ou fonctionnalités, d'une application web depuis une autre application.
Backup	En informatique, le Backup ou la sauvegarde est l'opération qui consiste à dupliquer et à mettre en sécurité les données contenues dans un système informatique.
Crontab	Cron est un programme disponible sur les systèmes de type Linux permettant de planifier des tâches régulières.
Git	Logiciel de gestion de versions.
Logs	Le terme log désigne un type de fichier, ou une entité équivalente, dont la mission principale consiste à stocker un historique des événements.
Monitoring	Le monitoring est l'anglicisme du terme surveillance et définit la mesure d'une activité.
Package	On appelle paquet (<i>package</i>) une archive comprenant les fichiers informatiques, les informations et procédures nécessaires à l'installation d'un logiciel.
SGBD	Système de Gestion de Base de données.
Snapshot	Il s'agit d'un instantané de stockage. Il peut être décrit comme une « photo » des données stockées sur un système de stockage de données à un instant précis.
SSH	SSH permet à deux ordinateurs d'établir une connexion directe et sécurisée au sein d'un réseau potentiellement non sécurisé, tel qu'Internet.
SSL	SSL signifie couche des sockets sécurisés (secure sockets layer). Protocole pour navigateurs Web et serveurs qui permet l'authentification, le chiffrement et le déchiffrement des données envoyées sur l'Internet.
SQL	SQL est un langage informatique normalisé servant à exploiter des bases de données relationnelles.
Repository	Répertoire de fichier situé dans le cloud.
Webservice	Protocole d'interface informatique permettant la communication et l'échange de données entre applications et systèmes.