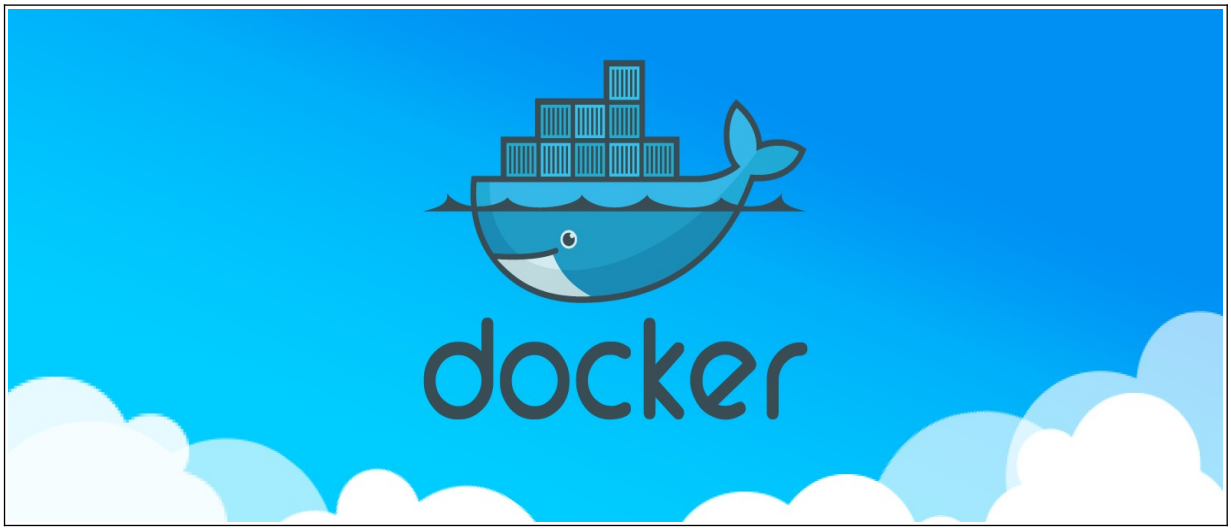
# Conteneurisation Docker




Suite à une demande récurrente des clients, nous allons **répondre à leur attentes** via la mise à disposition de **plateforme CMS**. Réalisation offrant rapidité, simplicité d'accès, volatilité du service, persistance des données ainsi que la haute disponibilité.

Ce document à pour but de guider son lecteur dans la mise à disposition de la technologie choisit.

*Dans ce projet **deux technologies** ont été retenus, dans les deux cas ces dernières devront avoir en leur possession :*

*- le **conteneur** de base hébergeant la **plateforme CMS***
*- le **conteneur** hébergeant l'éventuelle **base de données***
*- le ou les **volumes** contenant les **données des utilisateurs***

---

## Sommaire :

# I - Informations sur les technologies.

## 1./ Docker Compose


Docker Compose est un logiciel pour définir et exécuter des applications à partir de multiples conteneurs. Il est basé sur un fichier **YAML** qui permet de définir les services et les paramètres de leurs créations et ainsi de les démarrer par une commande unique.
Cela permet de gérer ce **stack (cet ensemble de conteneurs, services)** de manières **centralisée**.

*Cette technologies requiert les mises en œuvres suivantes :*

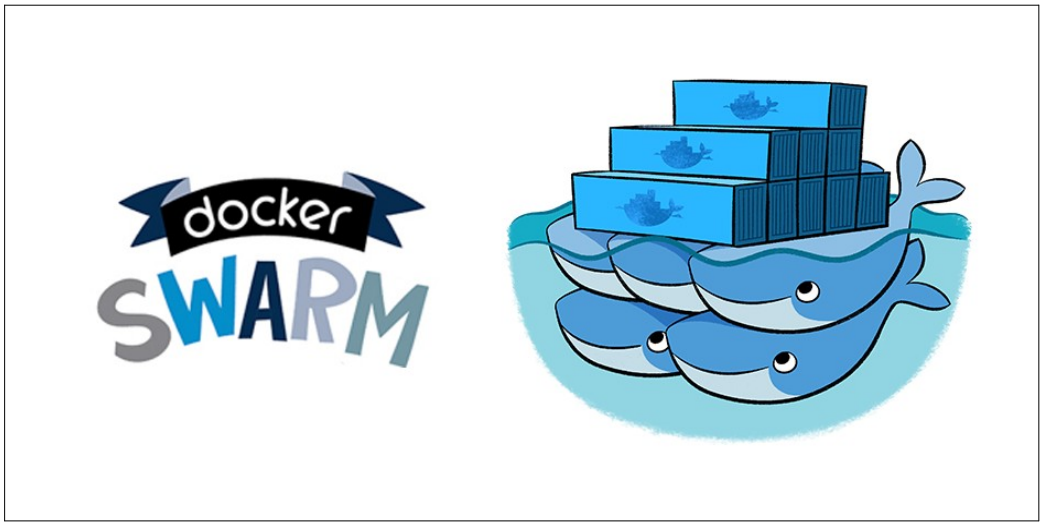
*- Gérer ce stack (cet ensemble de conteneurs, services) de manières centralisée.*

*- La personnalisation de la page d'accueil du CMS, contenant le nom de l'entreprise*

*- (Facultatif) l'envoi au client les informations de connexion (l'envoi n'est pas forcement automatique, par contre la génération des informations de connexion l'est)*

*- (Facultatif) la destruction des sites arrivés à expiration*

# *2./ Docker Swarm*




Docker Swarm est un logiciel produit par les développeurs de Docker qui **consolide** n'importe quel nombre **d'hôtes** Docker dans un cluster et permet la **gestion centrale** de celui-ci ainsi que l'**orchestration des conteneurs**.

Vous pouvez répartir rapidement, efficacement et de manière confortable les applications dans le réseau en tant que tâches.
En mode Swarm, Docker dispose de fonctions de **répartition de charge** (load balancing) intégrées.

*Cette technologies requiert les mises en œuvres suivantes :*

*- La mise en cluster des sites clients, permettant la tolérance de panne et la répartition de charge*

# *II - Schéma fonctionnel de l'architecture*

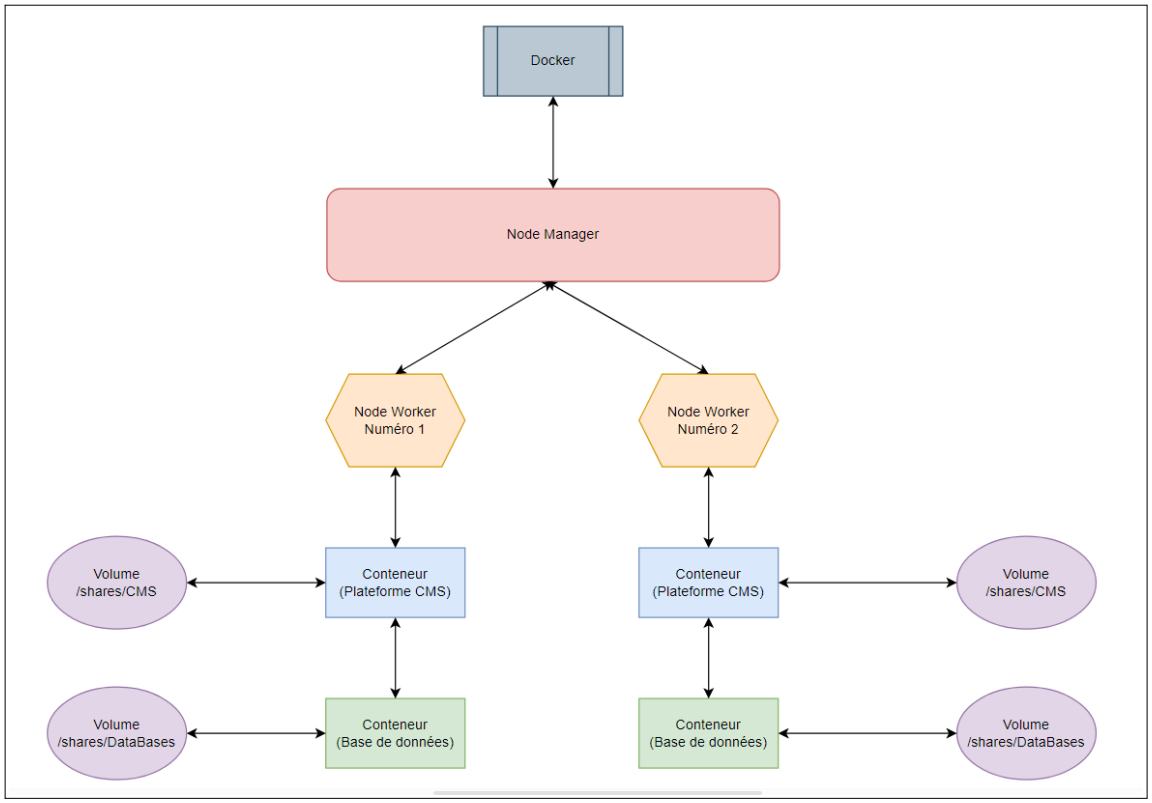
La technologie **Docker Swarm** à donc été retenue afin de répondre aux attentes du client, une **schématisation** du réseau de l'infrastructure docker est réalisé dans le but de détaillé et visualiser l'approche du système.

Avant tout il faut comprendre le fonctionnement de Docker Swarm ans une infrastructure.

*Une fonctionnalités est absolument requise, celle que l'on nomme « **nodes** », il en existe deux types :*

- **Workers** = Ils vont **exécuter les tâches** confiées par les managers. Un agent s'exécute sur chaque nœud et rend compte des **tâches qui lui sont affectées**. Il informe ainsi les nodes managers de **l'état des tâches affectées**.

- **Managers** = Ce sont les nodes **gestionnaires** de votre cluster. Ils **distribuent les tâches** aux nodes workers et ils effectuent également les **fonctions d'orchestration et de gestion**.



# III - Fonction des éléments Docker employés et leur utilité.

## - Volume :

Un volume docker joue le même rôle qu'un dossier, vous pouvez donc y stocker ou en extraire des données. Il permet également de monter les données dont un conteneur a besoin afin que ce dernier puisse fonctionner correctement lors de son lancement.

```
root@debian:~# docker volume create --name userdata
userdata
```

## - Conteneur :

Les conteneurs sont une forme de virtualisation des systèmes d'exploitation. C'est un format d'empaquetage qui regroupe tout le code et les dépendances d'une application dans un format standard, qui permet une exécution rapide et de fiable dans l'ensemble des environnements informatiques.

(Création d'un conteneur avec l'image Wordpress, plateforme CMS)

```
root@debian:~# docker run --name some-wordpress -p 9000:80 --network swarmnet2 -
d wordpress
Unable to find image 'wordpress:latest' locally
```

```
root@debian:~# docker service create --name some-wordpress -e WORDPRESS_DB_HOST=
172.17.102.5:9005 -e WORDPRESS_DB_USER=wordpress -e WORDPRESS_DB_PASSWORD=123AZE
qsd! -p 9000:80 --network swarmnet2 --constraint=node.role==worker -d wordpress
```

## *IV - Commandes principales de gestion du système.*

**- Visualisation des différents services :**

*Docker service ls*

- Supprimer **un service :**

*Docker service rm (ID du service)*

**- Visualisation des différents conteneurs :**

*Docker container ls [-a]*

*- Supprimer **un conteneur :***

*Docker container rm [ID du conteneur]*

*- Observer **tous les nœuds :***

*docker node ls*

## *V - Manques fonctionnels ou défauts rencontrés.*

Des possibles problèmes de permission au lancement d'un conteneur lié au réseau peuvent apparaître, la commande suivante permet de rétablir ce soucis :

*docker network create –driver overlay –attachable swarmnet2*