

## Documentation

### Architecture pour des applications web redondantes et sécurisées

#### Introduction

Dans cette documentation, vous trouverez les directives et bonnes pratiques qui nous ont permis de mettre en place une architecture permettant de déployer des applications web (type Wordpress) redondantes et sécurisées.

#### Infrastructure mise en place

Pour faire fonctionner notre serveur nous avons utilisé un ordinateur avec les configurations minimum nécessaire.

```
System information as of jeu. 15 juin 2023 11:41:55 UTC

System load:      0.0205078125      Processes:          184
Usage of /home:   0.0% of 195.80GB  Users logged in:    1
Memory usage:     10%              IPv4 address for docker0: 172.17.0.1
Swap usage:       0%               IPv4 address for wlp3s0: 192.168.1.53
Temperature:     59.0 C
```

Nous avons installé ubuntu serveur sur un ordinateur qui fait office de serveur et l'avons connecté au wifi (qui joue le rôle de routeur). Nous pouvons donc accéder à notre serveur depuis l'extérieur.

Pourquoi l'avoir choisi ? →

```
melaine@melaineserver:~$ grep MemTotal /proc/meminfo
MemTotal:      8028288 kB
melaine@melaineserver:~$
```

on fait :  $8028288 / 1048576 =$  environ 8 GB

Nous avons environ 8 GB de mémoire ram ce qui est suffisant pour faire tourner notre serveur.

On regardant aussi la place mémoire on à pu remarquer que l'on avait plus de 500 GB d'espace disque, ce qui est correct à notre échelle.

## Définition du réseau et des hôtes

### Segmenter le réseau

Il faut diviser le réseau en segments logiques : nous aurons les serveurs web stockés d'une part, la partie backend dans une autre et un dernier segment pour la base de données.

L'outil utile pour jongler entre ces divers segments : Docker !

Ainsi, lors de la création d'un nouveau wordpress, la partie serveur apparaît dans un conteneur tandis que la base de données affiliée dans un autre !

### Ouvrir les bons ports

Port 2222 : connexion SSH au serveur (nous avons configuré un ordinateur pour qu'il nous serve de serveur, il a fallu y installer *ubuntu server*)

→ **Bonne pratique** : Limiter l'accès à ce port aux adresses IP autorisées afin de renforcer la sécurité (configurer votre pare-feu)

Port 80 : il doit être ouvert pour permettre aux serveurs web Apache d'accéder aux applications WordPress.

→ **Bonne pratique** : S'assurer que le port est accessible depuis les clients extérieur (faire des tests de ports avec des outils en ligne "Open Port Check Tool")

Plage de ports 8081 à 8200 : On prévoit d'héberger plusieurs instances de WordPress avec des numéros de ports uniques !

Exemple : dns:8081 ,dns:8082,...

→ **Bonne pratique** : S'assurer que ces ports sont ouverts sur votre pare-feu pour permettre le trafic entrant(=données/requêtes réseau envoyées vers le serveur)

### SSH & DNS

S'assurer d'avoir une bonne stratégie d'attribution d'adresses IP pour chaque hôte du réseau !

Pour vous connecter à votre serveur, il est crucial de mettre en place une connexion SSH, le configurer sur le port 2222 puis créer le nombre d'utilisateurs qui vous convient !

Nous, nous avons choisi DUCK DNS pour créer notre domaine ! Mais libre à vous de choisir un autre service !

## Surveillance et performance grâce à Nagios

Nagios est un logiciel open source de surveillance qui permet de surveiller en temps réel les hôtes, les services réseau et les métriques système.

Il offre une interface web pour visualiser l'état de votre infrastructure et peut nous envoyer des alertes en cas de problèmes.

Dans notre cas, Nagios est utile pour maintenir la disponibilité et les performances de votre serveur Ubuntu et détecter rapidement les problèmes potentiels.

On peut configurer des notifications (messages, mails, sms) directement depuis les fichiers afin d'être alerté en temps réel.

On peut choisir lesquels objets on veut surveiller, par exemple, nous allons surveiller si nos conteneurs fonctionnent correctement et non pas de problème de performance. Il permet aussi la planification de tâches et plein d'autres fonctionnalités très utiles au bon fonctionnement d'un serveur.