

Projet

29/01/2024

—

BARILLY Dylan

2024



Objectifs du projet

L'objectif du projet est l'installation opérationnelle de Wordpress redondant et tolérant à une panne.

- Serveur Wordpress (10.20.5.118)
- Serveur BDD (10.20.5.119)

User Wordpress : Dylan / Azerty123!

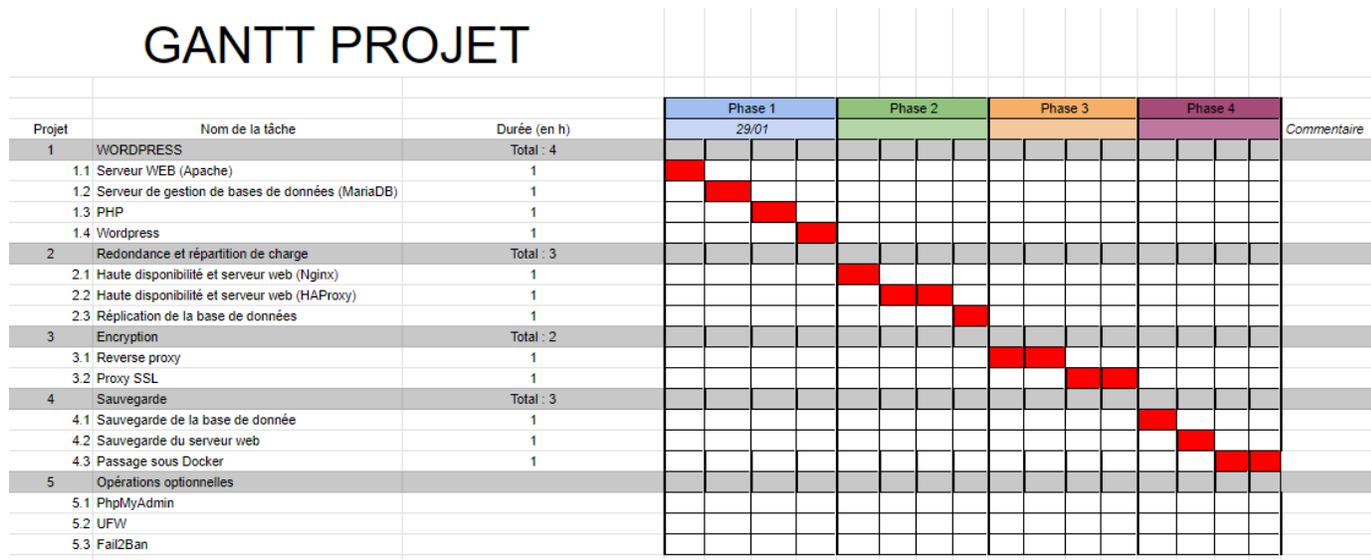
Serveur BDD : 192.168.61.6

Serveur Web : 192.168.61.7

Serveur Web 2 : 192.168.61.8

Serveur Nginx : 192.168.61.9

Séance 1



Étapes exécutées lors de la séance

Serveur WEB :

- Installation d'apache
 - Mettre en admin le compte pour avoir les permissions : **"su -"**
 - Installation apache : **"apk get apache2"**
 - Lancement d'apache : **"rc-service apache2 start"**
 - Mettre à jour apache : **"rc-update add apache2"**
 - Relancer apache : **"rc-service apache2 restart"**

On vérifie l'installation en récupérant l'IP de l'hôte et en vérifiant sur la barre de recherche, si cela fonctionne, "It works!" est censé s'afficher. Si cela ne fonctionne pas, il faut vérifier si le service est bien démarré avec la commande **"rc-service apache2 status"** afin de l'apercevoir en ligne de commande, il suffit de taper **"curl localhost"** et si tout fonctionne bien il est censé afficher le html du localhost : `<html><body><h1>It works!</h1></body></html>`

- Installation Php
 - Editer le fichier suivant "**nano /etc/apk/repositories**"
 - Pour récupérer la liste des paquets communautaires : "**apk update**"
 - Pour installer le module php : "**apk add php-apache2**"
 - Se rendre dans : "**cd /var/www/localhost/htdocs**" puis décommenter la seconde ligne :
 - Se rendre dans "**nano index.php**" puis indiquer :

```
<?php  
    phpinfo();  
?>
```
 - Vérifier avec "**curl localhost/index.php**" si les modifications ont bien été faites.
 - Redémarrer l'instance : "**rc-service apache2 restart**"

On vérifie l'installation en se connectant à nouveau avec "<http://10.20.5.118/index.php>" dans la barre de recherche, si tout est bon, vous devez apercevoir l'environnement php. Afin de l'apercevoir en ligne de commande, il suffit de taper "**curl localhost/index.php**" et il est censé vous afficher le contenu.

Serveur BDD :

- Installation de la bases de données MariaDB
 - Ajouter le module Mysql : **"apk add mysql mysql-client"**
 - Installation de MariaDB : **"mysql_install_db --user=mysql --datadir=/var/lib/mysql"**
 - Lancer MariaDB : **"rc-service mariadb start"**
 - Mettre des permissions pour manager MariaDB : **"mysqladmin -u root password toor"**
 - Récupérer la liste des paquets : **"apk update"** puis **"apk upgrade"**
 - Lancer MariaDB avec : **"rc-service mariadb start"**

Afin de vérifier que cela fonctionne, on s'assure que MariaDB est bien lancé avec la commande : **"rc-service mariadb status"** et si c'est en *"started"* cela est fonctionnel, on utilise la commande : **"mysql"**, si cela fonctionne on entre directement dans la base de donnée.

- Connecter MariaDB depuis la VM Web
 - Installer le paquet client de mysql : **"apk add mysql-client"**
 - Se rendre sur le serveur BDD afin d'ouvrir le port mysql, donc aller dans le répertoire : **"cd /etc/my.cnf.d"**
 - Se rendre dans le mariadb-server.cnf avec la commande : **"nano mariadb-server.cnf"**
 - Puis commenter la ligne **"skip-networking"** qui interdisait la communication réseau
 - Redémarrer le service : **"rc-service mariadb restart"**
 - Avec **"nmap 10.20.5.119"** on vérifie que le port mysql est bien ouvert
 - Cependant il faut créer un user qui a le droit de se connecter depuis une machine distante, il faut se connecter dans la bdd avec **"mysql"** puis **"CREATE USER 'test'@'%' IDENTIFIED BY 'test';"**
 - Donner les permissions à l'utilisateur test avec : **"GRANT ALL PRIVILEGES ON *.* TO 'test'@'%' WITH GRANT OPTION;"** puis **"FLUSH PRIVILEGES;"**

Pour vérifier la connexion à distance depuis le serveur web, il suffit de taper la commande **"mysql -h 10.20.5.119 -u test -p"** puis de se connecter avec le mot de passe *"test"*, si cela fonctionne on se retrouve bien dans la base de donnée.

- Créer la base de données Wordpress et l'utilisateur correspondant
 - Créer la base de données Wordpress : **"CREATE DATABASE wordpress_db;"**
 - Créer un utilisateur : **"CREATE USER 'test'@'%' IDENTIFIED BY 'test';"**
 - Donner des permissions à l'utilisateur : **"GRANT ALL PRIVILEGES ON wordpress_db.* TO 'test'@'%' WITH GRANT OPTION;"**

- Installation de Wordpress
 - Se rendre dans le répertoire **"cd /var/www"** dans le serveur web
 - Prendre les paquets de wordpress : **"wget <https://wordpress.org/latest.tar.gz>"**
 - Décompresser avec : **"tar xzf latest.tar.gz"**
 - Redémarrer le service apache : **"rc-service apache2 restart"**
 - Aller dans le répertoire : **"cd /etc/apache2/conf.d/"**
 - Afin de créer le wordpress.conf **"nano wordpress.conf"** puis y mettre :
<VirtualHost *:80>

 DocumentRoot /var/www/wordpress

 ServerName 172.18.109.149

</VirtualHost>

<Directory "/var/www/wordpress">

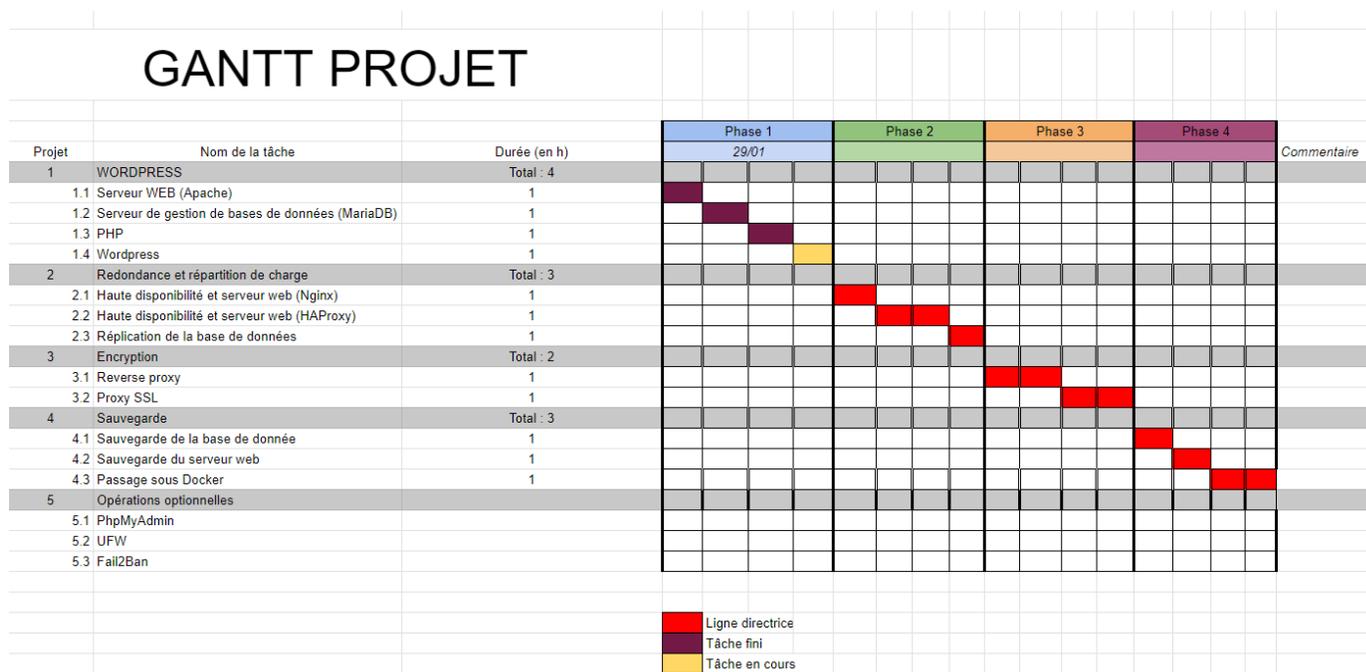
 Options Indexes FollowSymLinks

 AllowOverride None

 Require all granted

</Directory>
 - Ajouter le module : **"apk add php-mysqli"**

Pour vérifier si c'est bien fonctionnel, il suffit de se connecter sur le lien **"http://10.20.5.118/wp-admin/"** et normalement vous pouvez apercevoir le wordpress d'installer.



Dans cette première séance, nous avons installé le Serveur Web et le Serveur de BDD, puis l'installation de Php et pour finir l'installation de Wordpress, il nous manque juste de donner des droits à l'utilisateur afin de configurer le Wordpress.

Séance 2

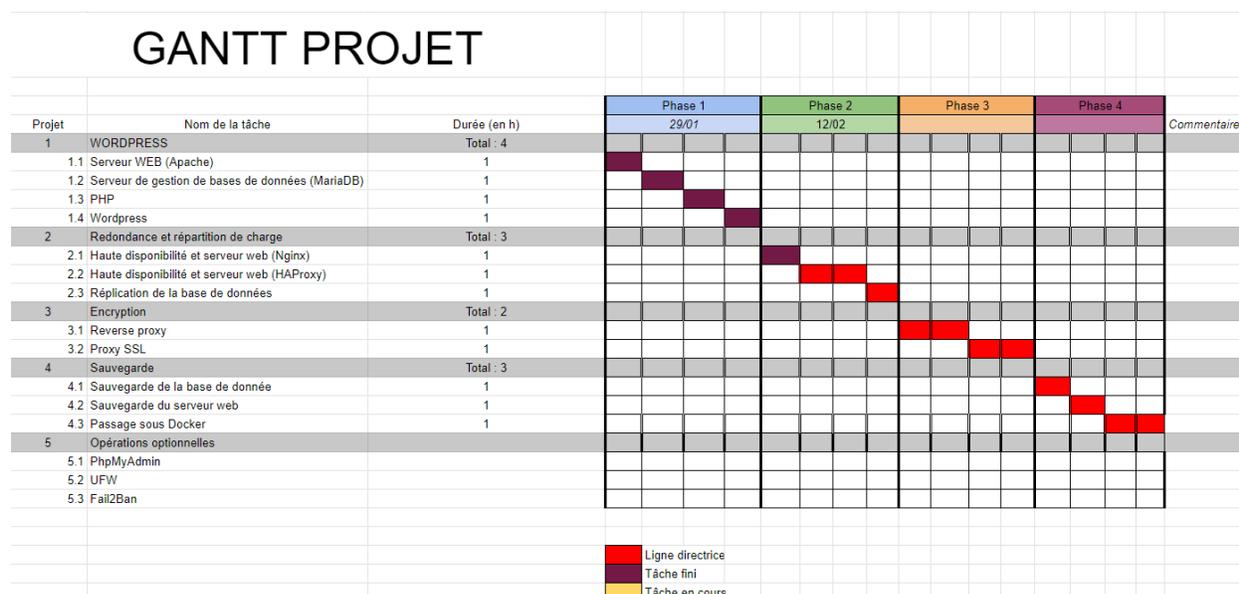
Pour finir l'installation de Wordpress, ajouter sur le serveur Web : **"chown apache:apache -R /var/www/wordpress"** afin de donner les droits à l'utilisateur puis configurer le wordpress sur le site.

Nginx :

- Installation Nginx
 - Faire : **"apk update"**
 - Installer Nginx : **"apk add nginx"**
 - Créer un nouvel utilisateur : **"adduser -D -g 'www' www"**
 - Créer un répertoire pour les fichiers html : **"mkdir /www"** puis **"chown -R www:www /var/lib/nginx"** et **"chown -R www:www /www"**
- Pour mettre une ip statique sur une VM il suffit de faire : **"nano /etc/network/interfaces"** puis changer en fonction des configurations réseaux sur les trois serveurs existants :

```
auto lo
iface lo inet loopback

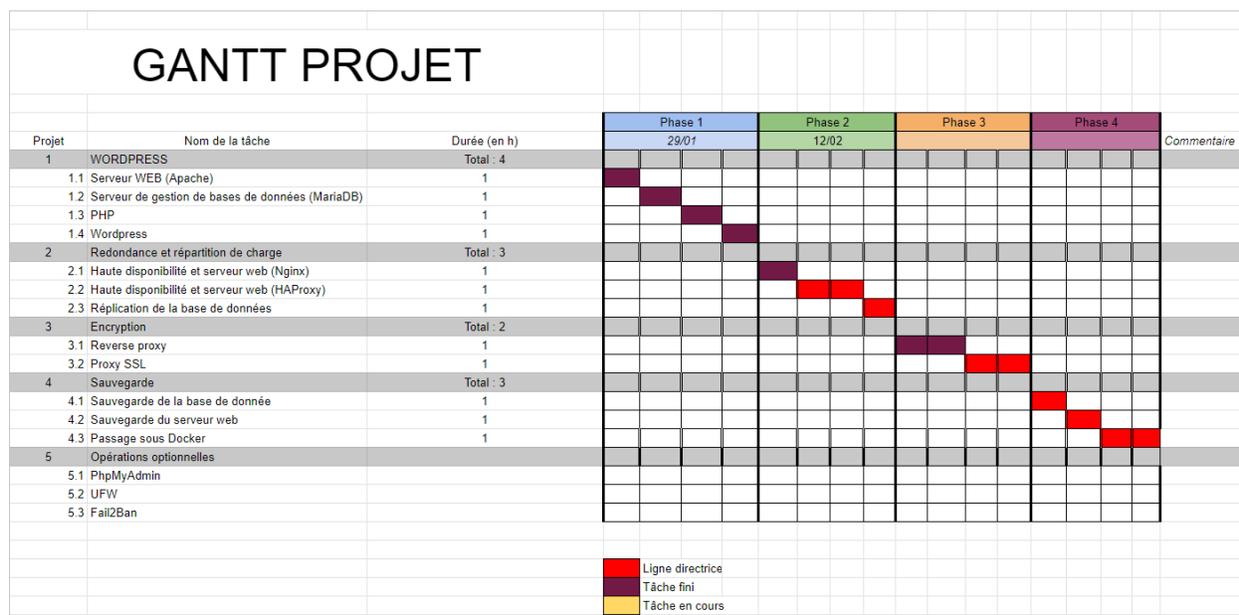
auto eth0
iface eth0 inet static
    address 192.168.61.6
    netmask 255.255.255.224
    gateway 192.168.61.30
```



Séance 3

Reverse proxy :

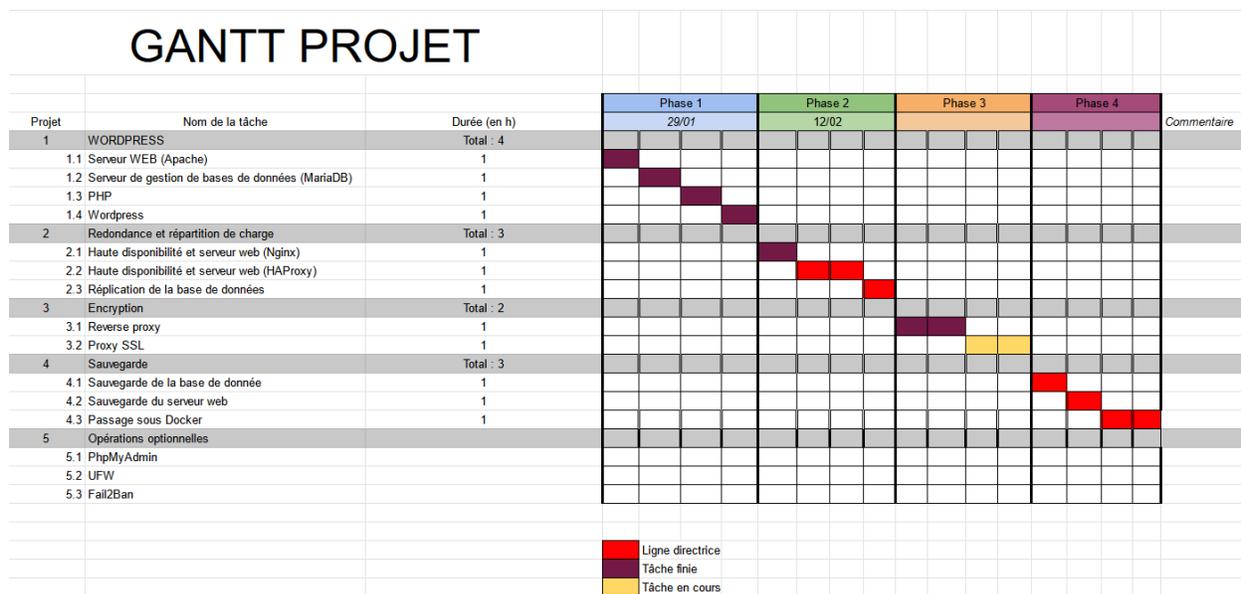
- Ajouter le package nginx : **"apk add nginx"** puis **"apk update && apk upgrade"**
- Créer une nouvelle VM en y configurant une IP static : **"nano /etc/network/interfaces"**
- Ajouter le package apache2 **"apk add apache2"**
- Désigner le bon serveur au bon port



Séance 4

Proxy SSL :

- Ajout du SSL : **"apk add openssl openssl-dev"**



Réalisation Finale

Objectifs atteints

- Réalisation d'un GANTT affichant la planification et le suivi de toutes les séances
- Mise en place d'un serveur web redondant
- Mise en place de Wordpress
- Mise en place d'un serveur BDD
- Mise en place d'un serveur PHP
- Mise en place d'un reverse proxy

Objectifs non atteints

- La réalisation du proxy SSL est en cours, des problèmes durant la séance ont fait ralentir la progression

Auto-évaluation

mission	non réalisé	en cours	réalisé
GANTT			X
Serveur Web			X
Serveur PHP			X
Serveur Base de données			X
WordPress			X
PHPMyAdmin	X		
SSL		X	
Reverse Proxy			X
Sauvegarde automatique de la Base de données	X		
Redondance Web NGINX			X
Redondance Web HAProxy	X		



Dockerisation	X		
UFW	X		
FAil2Ban	X		