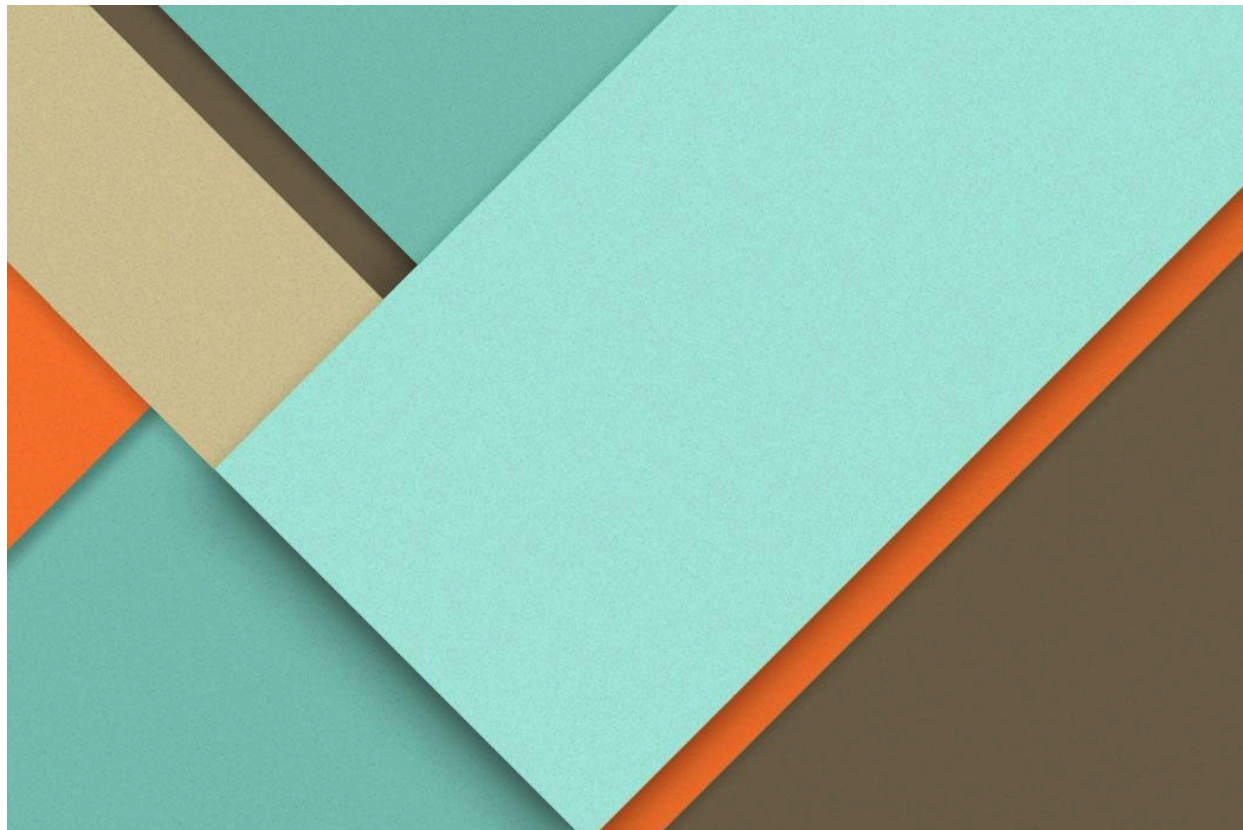




Travail en mode projet

WordPress

29-01-2024

Margaux TANET
2023-2024

Objectifs du projet

Installation propre de Wordpress avec comme élément un serveur web, PHP, BDD, Proxy, SSL en s'appuyant sur le diagramme de Gantt. Faisant redondance et tolérance.

Séance 1

Étapes exécutées lors de la séance

Passer en mode administrateur: commande **su-**

Cela permet d'avoir les permissions accordées ainsi que plus d'actions.

→ Installation apache:

Taper les commandes suivantes:

- **apk add apache2** => *télécharge et installe les fichiers nécessaires pour exécuter Apache sur le système.*
- **rc-service apache2 start** => *démarre le service Apache HTTP Server*
- **rc-update add apache2** => *ajoute le service Apache HTTP Server au niveau d'exécution par défaut du système. Cela signifie que le service sera automatiquement démarré au démarrage du système.*
- **rc-service apache2 restart** => *Cette commande sert à quand on apporte des modifications, Apache redémarre*

Après l'installation et les mises à jour: récupérer l'adresse IP du pc en ayant tapé au préalable "**ip a**" pour la connaître et l'indiquer dans la barre de recherche du pc hôte. Après avoir rentré cette adresse un message "it works!" est censé s'afficher. Cela permet de vérifier que cela marche.

Attention: s'il n'y a pas de message, bien pensé à redémarrer le service avec la commande "rc-service apache2 status"

La commande **apk add curl** permet d'installer l'utilitaire curl via le gestionnaire de paquets apk. Une fois cette commande exécutée, curl sera installé sur le système et vous pourrez l'utiliser pour effectuer des transferts de données via différents protocoles réseau tels que HTTP, HTTPS, FTP, etc

Si tout est bon il devrait apparaitre ce message: "**it works**" avec des balises HTML de chaque côté.

La commande `curl localhost` permet d'effectuer une requête HTTP vers l'adresse localhost (notre propre machine).

Il est judicieux d'installer l'éditeur de fichier **nano** pour pouvoir modifier les packages. Installation de cet éditeur: **apk add nano**.

→ Installation PHP

Se rendre dans le répertoire **/etc/apk/repositories** et tapé la commande suivante: **nano /etc/apk/repositories**. Cette commande va permettre de donner l'autorisation de la bibliothèque. Il faut décommenter la ligne <http://mirrors.ircam.fr/pub/alpine/v3.16/community>

On installe le module PHP pour apache grâce à la commande "**apk update**" pour récupérer la liste des paquets communautaire.

→ Création fichier index.php

Se rendre dans le répertoire avec les fichiers servis par le serveur WEB pour créer un fichier `index.php`. Pour se faire: "**apk update**" puis "**apk add php-apache2**".

Se rendre dans le répertoire **/var/www/localhost/htdocs** puis **nano /var/www/localhost/htdocs** et entrer `<?php phpinfo(); ?>` et enregistrer.

Tester avec la commande `curl`: **curl localhost/index.php**. Cela permet de récupérer le contenu de la page web située à l'adresse `http://localhost/index.php`

Redémarrer l'instance avec **rc-service apache2 restart**.

Pour **vérifier** sur l'hôte du navigateur, il faut taper l'adresse IP de la machine virtuelle suivi de **/index.php**. Il devrait y avoir une page.

→ Création de Mariadb

Créer une autre machine virtuelle. Taper les commande suivante:

- **apk add mariadb** => utilise le gestionnaire de paquets apk pour installer le serveur MariaDB sur le système.
- **rc-service mariadb status** => vérifie l'état du service MariaDB, c'est-à-dire s'il est en cours d'exécution ou non.
- **rc-service mariadb start** => démarre le service MariaDB.
- **/etc/init.d/mariadb setup** => utilisée pour configurer MariaDB, notamment pour initialiser la base de données et définir les paramètres de sécurité initiaux.

- ***rc-service mariadb start*** => démarre le service MariaDB, après modification
- ***rc-service mariadb status*** => vérifie l'état du service MariaDB après le démarrage.
- ***apk add mariadb-client*** => utilise le gestionnaire de paquets apk pour installer le client MariaDB sur le système, ce qui permettra aux utilisateurs de se connecter à des bases de données MariaDB à partir de la ligne de commande

On vérifie le tout en lançant la machine virtuelle locale: **mysql -u root**

→ Installation côté client:

Les commandes suivantes:

- **apk update**
- **apk add mysql_client**

Pour la connexion via le port par défaut sur l'IP du serveur de base de données: **apk add nmap**

Modifier la configuration dans le répertoire **/etc/my.cnf.d** puis dans le fichier **mariadb-server.cnf**

En effet, le port n'est pas ouvert il faut **décommenter skip-networking**.

Redémarrer les services mariadb: **rc-service mariadb restart**

- On créer un utilisateur dans mariadb qui a le droit de se connecter depuis la machine distante : **CREATE USER 'test'@'%' IDENTIFIED BY 'test';**
- On lui donne les privilèges: **GRANT ALL PRIVILEGES ON *.* TO 'test'@'%' WITH GRANT OPTION;**
- Et ne pas oublier de mettre à jour les modifications: **FLUSH PRIVILEGES;**

Depuis apache on essaye de se connecter depuis la base de donnée : **mysql -h 10.20.5.129 -u test -p** (adresse IP de ma base de donnée).

→ Création de la base de données pour Wordpress sur mariadb

création d'une base de données qui sera utilisée par WP et un nouvel utilisateur (mdp: test).

Pour se faire :

- **CREATE DATABASE wordpress_db;**
- **CREATE USER 'wordpress_dbuser'@'%' IDENTIFIED BY 'test';**
- **GRANT ALL PRIVILEGES ON wordpress_db.* TO 'wordpress_dbuser'@'%' WITH GRANT OPTION;**

→ Téléchargement de wordpress:

Se rendre sur la machine virtuelle web:

Se rendre dans le répertoire: **cd /var/www**

Installer wordpress et le décompresser:

- **wget https://wordpress.org/latest.tar.gz**
- **tar xzf latest.tar.gz**
- Redémarrer: **rc-service apache2 restart**

Pour ajouter des hôtes virtuels pour chaque service il faut se rendre dans le répertoire **etc/apache2/conf.d/wordpress.conf** puis **nano wordpress.conf** et coller la configuration ci-dessous:

```
<VirtualHost *:80>
    DocumentRoot /var/www/wordpress
    ServerName 10.20.5.116
</VirtualHost>

<Directory "/var/www/wordpress">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Mon GANTT à l'issue de cette séance:

GANTT CONTEXTE

NUMÉRO	TITRE DE LA TÂCHE	1ère séance			
		1	2	3	4
1	Installation fonctionnelle Wordpress				
1.1	Apache				
1.1.1	PHP				
1.2	Mariadb				
1.3	Wordpress				
2	Redondance répartition				
2.1	Nginx				
2.2	HA Proxy				
2.3	Réplication base de données				
3	Encryption				
3.1	Reverse Proxy				
3.2	Proxy SSL				
4	Sauvegarde				
4.1	base de données				
4.2	Serveur web				
4.3	Docker				
4.4	Performances du projet				
5	Opérations optionnelles				
5.1	PhpMyAdmin				
5.2	UFW				
5.3	Fail2Ban				

Séance 2

Étapes exécutées lors de la séance

Se rendre sur un navigateur web et entrer l'IP du serveur suivi de */wp-admin*.

→ Finalisation de l'installation de Wordpress:

Lors de l'installation WordPress, si vous avez l'erreur sur l'impossibilité d'écrire wp_config.php, cela est dû aux droits sur le répertoire /var/www/wordpress. Nous l'avons installé avec root mais le service apache utilise un autre utilisateur pour accéder aux fichiers. Il faut donc céder la propriété de ces fichiers à l'utilisateur qui est responsable du processus Apache en tapant la commande:

```
chown apache:apache -R /var/www/wordpress
```

Relancer la page de navigateur, vous devriez atterrir sur la page de configuration de Wordpress. Il vous est demandé de donner un nom au site, un identifiant, mot de passe et un mail. Puis poursuivre l'installation en cliquant sur le bouton install puis être redirigé vers votre wordpress (<http://10.20.5.88/wp-admin/>).

2ème partie: redondance et répartition de charge

Création d'un serveur Nginx qui transmet la requête aux serveurs web en redondance:

→ Installer nginx:

- **apk add nginx** => Installe le serveur web Nginx.
- **apk add nano** => Installe l'éditeur de texte Nano.
- **adduser -D -g 'www' www** => Crée un utilisateur système "www" sans mot de passe ni répertoire personnel.
- **chown -R www:www /var/lib/nginx** => Modifie le propriétaire et le groupe de /var/lib/nginx en "www:www".
- **chown -R www:www /www** => Modifie le propriétaire et le groupe de /www en "www:www".

Se rendre sur la machine virtuelle de MariaDB et lui attribuer une IP fixe:

- **nano /etc/network/interfaces**
- **address 192.168.62.6**
- **netmask 255.255.255.0**

- gateway 192.168.62.30 (le stormshield)
- service networking restart

Actions à réaliser sur la machine virtuelle Wordpress:

Réaliser ces commandes:

- nano /etc/network/interfaces
- address 192.168.62.7
- netmask 255.255.255.0
- gateway 192.168.62.30 (le stormshield)
- service networking restart

Actions à réaliser sur la machine virtuelle Nginx:

- nano /etc/network/interfaces
- address 192.168.62.9
- netmask 255.255.255.0
- gateway 192.168.62.30 (le stormshield)
- service networking restart

Dupliquer la machine virtuelle Wordpress pour en créer une deuxième et faire la même procédure pour avoir une IP fixe:

Actions à réaliser sur la machine virtuelle Wordpress 2:

- nano /etc/network/interfaces
- address 192.168.62.8
- netmask 255.255.255.0
- gateway 192.168.62.30 (le stormshield)
- service networking restart

Il faut configurer maintenant **NGINX**. Pour cela se rendre dans le répertoire **/etc/nginx/http.d/*.conf;**

Et taper la commande suivante: **nano /etc/nginx/http.d/wp_conf** pour créer un fichier "wp_conf" avec la configuration suivante:


```

http {
    upstream loadbalancedservers {
        server IP SERVEUR WORDPRESS;
        server IP SERVEUR WORDPRESS 2;
    }

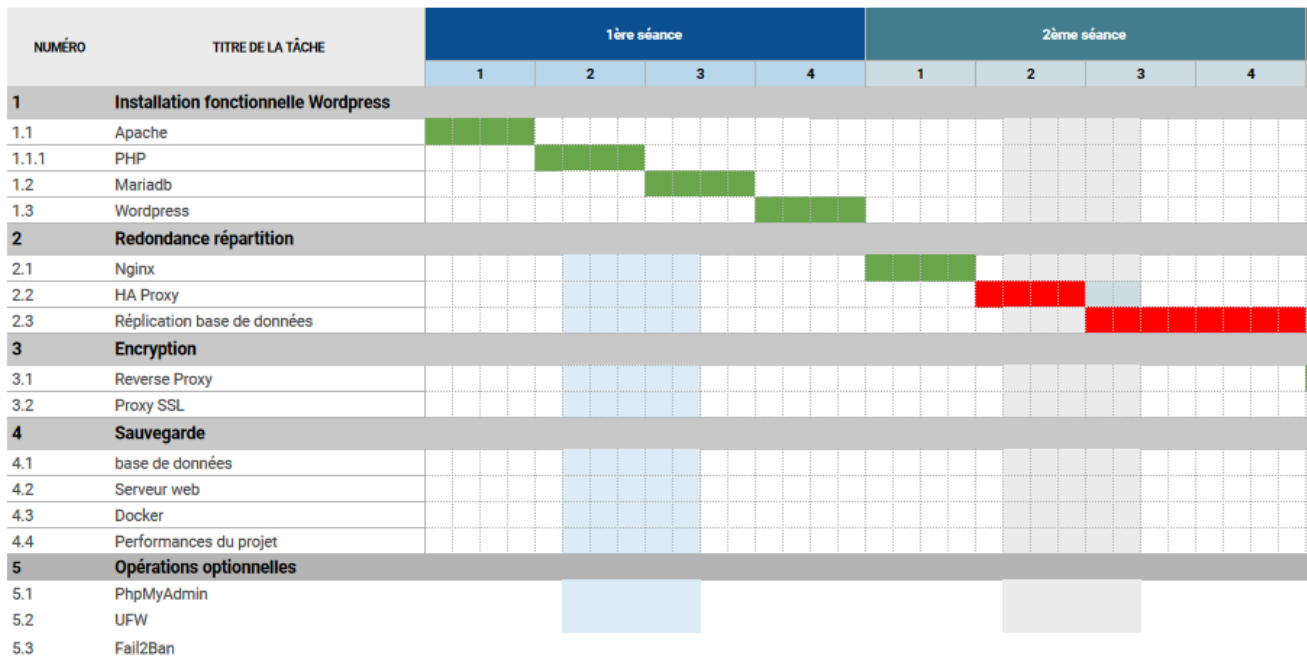
    server {
        listen 80;

        location / {
            proxy_pass http://loadbalancedservers;
        }
    }
}

```

Mon GANTT à l'issue de cette séance:

GANTT CONTEXTE



Séance 3

Etapas exécutées lors de la séance

3ème partie : Encryption et redirection

→ Reverse proxy:

Créer une autre machine virtuelle avec Nginx en laissant la configuration carte WAN.

Réaliser ces commandes:

- **apk update**
- **apk add nginx**
- **apk upgrade**

Réaliser la configuration dans **cdc /etc/network/interfaces:**

Passer en **static**

Renseigner les adresses suivant votre contexte:

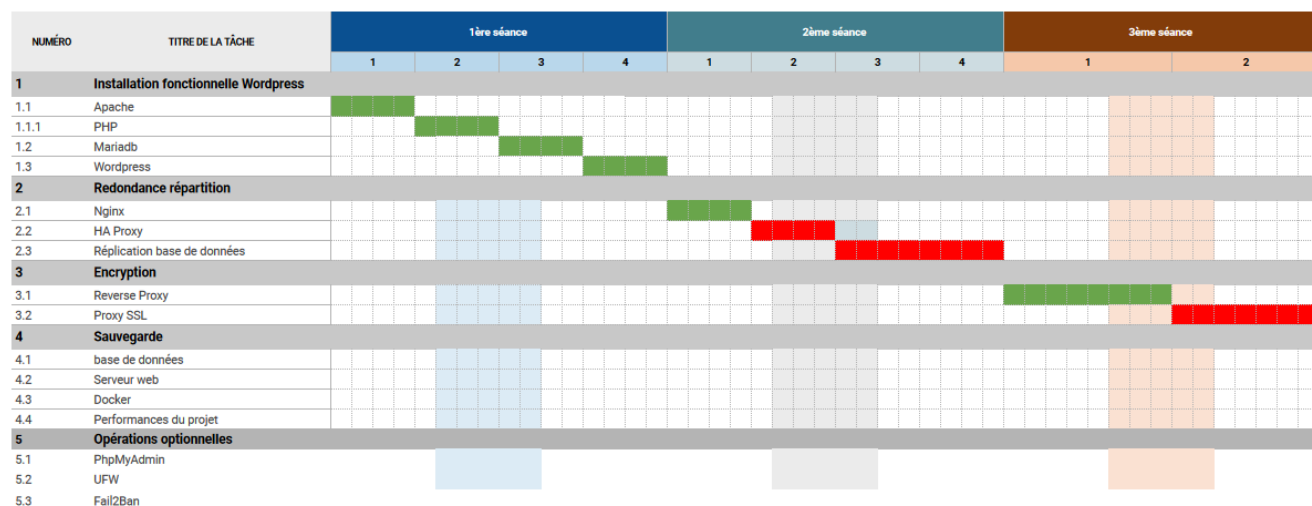
- **address** 192.168.62.22
- **netmask** 255.255.255.224
- **gateway** 192.168.62.30

Se rendre dans **nano /etc/nginx/ http.d/default.conf** et rajouter **proxy_pass http:// SERVEUR WP** donc ici pour moi 192.168.62.8

Démarrer sur la machine virtuelle cliente en tapant 192.168.62.22. (adresse de mon reverse proxy)

Mon GANTT à l'issue de cette séance:

GANTT CONTEXTE



Réalisation Finale

Proxy SSL

Objectif 1 : Création de certificat et renvoie sur le bon serveur:

Se rendre sur la VM NIGINX et taper *"apk add openssl openssl-dev"*.

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		