

Documentation d'architecture - François BONNIN

Introduction :

Tout au long de l'année, nous avons vu divers outils pour la mise en place d'infrastructure sécuriser via des systèmes de virtualisation (GNS3 ou VMWare) à travers des technologies telles que Debian, Alpine Linux, OpenWRT, Pfsense et Windows Server.

Le projet choisi a été choisi en duo, avec Raimana (qui a dû, pour raison personnelle quitter l'école), c'est **une architecture permettant de déployer des applications web redondées et sécurisées**. C'est un sujet en lien avec la spécialisation de l'année prochaine (développement web), il était donc judicieux de se lancer là-dedans.

Sommaire

1. [Schéma](#)
2. [Outils utilisés](#)
3. [Plan IP](#)
4. [Firewall PfSense](#)
5. [Reverse Proxy](#)
6. [Serveur Nginx](#)
7. [MariaDB](#)
8. [Wordpress](#)
9. [Wekan](#)
10. [Haute disponibilité](#)
11. [Installer une application web](#)
12. [Contact](#)

Schéma

Afin d'avoir une sécurité optimale j'ai choisi de mettre en place un serveur PfSense qui possèdera le reverse proxy, et le loadbalancer.

Deux serveurs web me permettront d'avoir des applications web redondées, liés entre eux par HeartBeat afin d'avoir de la Haute Disponibilité.

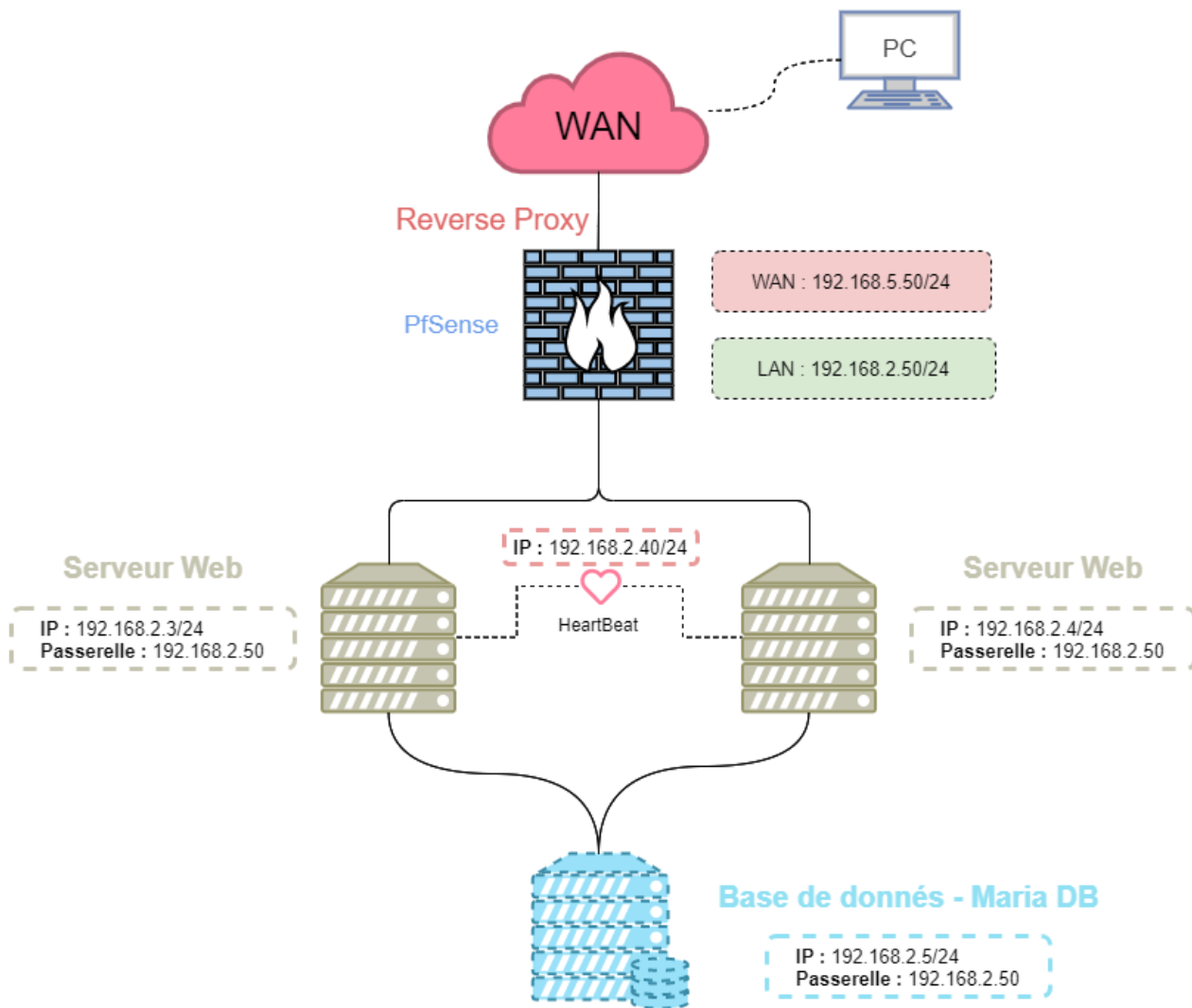
Ainsi qu'une base de données sous MariaDB.

Nous obtenons le schéma d'infrastructure suivant : qui possèdera le reverse proxy, et le loadbalancer.

Deux serveurs web me permettront d'avoir des applications web redondées, liés entre elles par HeartBeat afin d'avoir de la Haute Disponibilité.

Ainsi qu'une base de données sous MariaDB.

Nous obtenons le schéma d'infrastructure suivant :



Outils utilisés

Voici les différents outils utilisés au cours de ce TP :

- **VMware Workstation 16 Pro** va nous permettre d'exécuter des systèmes d'exploitation (PfSense et Debian) en tant que machines virtuelles afin de simuler dans de réelles conditions ce projet.
- **Debian** est un des systèmes Linux les plus fiables et les plus stables. Il a également un avantage économique car c'est un système d'exploitation open-source donc gratuit. Nos serveurs web et notre base de données seront donc mis en place sur Debian.
- Relativement pratique avec son interface graphique et sa gratuité, **Pfsense**, basé sur FreeBSD, est un routeur/pare-feu dont l'ajout de packages peut le rendre multifonctionnel. Nous utiliserons donc Pfsense, avec le package Squid pour la mise en place du reverse proxy.
- Le choix de serveur HTTP se porte vers **Nginx** car il possède une importante stabilité, de hautes performances, une faible consommation de ressources, et surtout il est gratuit.
- **MariaDB** est un système de gestion de base de données gratuit et performant basé sur MySQL. Il propose un large choix de moteurs de base de données et un optimisateur de requête SQL efficace.
- **Wordpress** est un CMS, gratuit et libre qui permet de créer et de gérer différents types de sites internet.
- **Wekan** est un outil collaboratif de gestion de tâches, gratuit et open source.

- **HeartBeat** va nous permettre de surveiller la disponibilité des serveurs et lorsque l'un d'eux sera défaillant, les services seront automatiquement basculés sur l'autre serveur. Un programme simple et gratuit nous permettant d'avoir de la Haute Disponibilité sur nos serveurs web.

Plan IP

Afin de préparer au mieux la mise en place et la configuration de nos machines virtuelles, il est important de schématiser à travers un tableau les différentes IP, masques de sous-réseaux et passerelles (Gateway) utilisées.

	pfSense WAN	pfSense LAN	Cluster HeartBeat	Serveur Web1 : Debian	Serveur Web2 : Debian	Base de donnée: Debian
IP	WAN : 192.168.5.50	LAN : 192.168.2.50	192.168.2.40	192.168.2.3	192.168.2.4	192.168.2.5
Netmask	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0
Gateway	192.168.5.2		192.168.2.50	192.168.2.50	192.168.2.50	192.168.2.50

Pour configurer les serveurs en ip static on utilise `nano /etc/network/interfaces` puis on modifie la valeur de X en fonction du tableau ci-dessus :

```
auto lo
iface lo inet loopback

auto ens33
iface ens33 inet static
    address 192.168.2.X
    netmask 255.255.255.0
    gateway 192.168.2.50
```

Pour le *DNS* nous utiliserons le DNS de PfSense (192.168.2.50), on peut le modifier via le fichier `/etc/resolv.conf` .

Firewall PfSense

Afin de pouvoir accéder à l'interface et aux serveurs depuis l'extérieur (WAN = Wide Area Network) nous ajouterons des règles de redirection de port (*Firewall/NAT/Port Forward*). Ainsi nous ajouterons des règles pour le protocole SSH (port 22) et pour le protocole HTTP (port 80):

Port Forward

1:1

Outbound

NPT

Rules

<input type="checkbox"/>		Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
<input type="checkbox"/>	<input checked="" type="checkbox"/>	WAN	TCP	*	*	WAN address	2222	192.168.2.40	22 (SSH)	SSH - Cluster Heartbeat	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	WAN	TCP	*	*	WAN address	3222	192.168.2.40	80 (HTTP)	Wordpress	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	WAN	TCP	*	*	WAN address	4222	192.168.2.40	8000	Wekan	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	WAN	TCP	*	*	WAN address	2223	192.168.2.3	22 (SSH)	SSH - Server1	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	WAN	TCP	*	*	WAN address	3223	192.168.2.3	80 (HTTP)	SrvWeb1	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	WAN	TCP	*	*	WAN address	2224	192.168.2.4	22 (SSH)	SSH - Server2	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	WAN	TCP	*	*	WAN address	3224	192.168.2.4	80 (HTTP)	SrvWeb2	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	WAN	TCP	*	*	WAN address	2225	192.168.2.5	22 (SSH)	SSH - MariaDB	

Ainsi pour le ssh nous entrerons la commande dans une invite de commande et pour http dans un navigateur.

- Cluster HeartBeat :
 - ssh : **ssh user@pfSenseWANAddress -p 2222**
 - http : **pfSenseWANAddress:3222**
- Serveur Web1 :
 - ssh : **ssh user@pfSenseWANAddress -p 2223**
 - http : **pfSenseWANAddress:3223**
- Serveur Web2 :
 - ssh : **ssh user@pfSenseWANAddress -p 2224**
 - http : **pfSenseWANAddress:3224**
- Base de donnée:
 - ssh : **ssh user@pfSenseWANAddress -p 2225**
- Wekan :
 - http : **pfSenseWANAddress:4222**










































Ensuite, nous allons ajouter différentes règles WAN afin d'autoriser le flux des port ci dessus afin que PfSense ne bloque pas leur accès:





The changes have been applied successfully. The firewall rules are now reloading in the background.
Monitor the filter reload progress.



Floating **WAN** LAN

Rules (Drag to Change Order)

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	0/0 B	*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks	
<input type="checkbox"/>	0/0 B	IPv4 ICMP echo req	*	*	*	*	*	none		Allox Ping LAN to WAN	    
<input type="checkbox"/>	0/2.71 MiB	IPv4+6 *	*	*	*	*	*	none		Allow all ipv4+ipv6 via pfSsh.php	    
<input type="checkbox"/>	0/0 B	IPv4 TCP	*	*	*	80 (HTTP)	*	none			    
<input type="checkbox"/>	0/0 B	IPv4 TCP	*	*	*	443 (HTTPS)	*	none			    
<input type="checkbox"/>	0/0 B	IPv4 TCP	*	*	192.168.2.4	22 (SSH)	*	none		NAT SSH - Server2	    
<input type="checkbox"/>	0/0 B	IPv4 TCP	*	*	192.168.2.5	22 (SSH)	*	none		NAT SSH - MariaDB	    
<input type="checkbox"/>	0/0 B	IPv4 TCP	*	*	192.168.2.40	22 (SSH)	*	none		NAT SSH - Cluster Heartbeat	    
<input type="checkbox"/>	0/0 B	IPv4 TCP	*	*	192.168.2.40	8000	*	none		NAT Wekan	    

 Add
  Add
  Delete
  Save
  Separator

Reverse Proxy

Un reverse-proxui va nous ajouter de la sécurité en publiant de façon sécurisé plusieurs sites web (Wordpress, Wekan...) qui sont eux mêmes hébergés par plusieurs serveurs web à travers notre PfSense.

Pour la mise en place de celui-ci, nous allons nous diriger vers le paquet Squid.

On appelle reverse-proxy car comme son nom l'indique, il permet de faire l'inverse du proxy :

- **Proxy** : accéder à Internet depuis le réseau local.
- **Reverse-proxy** : accéder au réseau local depuis internet.

En terme de sécurité, le reverse proxy possède plusieurs avantages :

- Anonymise les serveurs web lors des requêtes du client.
- Optimise le temps de réponse des requêtes
- Répartis la charge entre plusieurs serveurs
- Analyse les flux en amont (afin d'éviter les erreurs)

1. Installer les packages Squid dans *System/Package Manager/Available Packages*. Rechercher *squid* et cliquer sur **Install**.
2. Créer la règle "System Tunables" dans *System/Advanced/System Tunables*, cliquer sur **New** et ajoutez les valeurs si dessous :

System / [Advanced](#) / [System Tunables](#) / [Edit](#) ?

[Admin Access](#) [Firewall & NAT](#) [Networking](#) [Miscellaneous](#) [System Tunables](#) [Notifications](#)

Edit Tunable

Tunable	<input type="text" value="net.ipv4.ip.portrange.reservedhigh"/>
Value	<input type="text" value="0"/>
Description	<input type="text" value="Reverse proxy"/>

Save

3. Configurer Squid Reverse Proxy dans *Services/Squid Reverse Proxy* et ajouter un élément dans la section **Web Servers**. Ainsi on va pouvoir ajouter un serveur web.

[General](#) [Web Servers](#) [Mappings](#) [Redirects](#) [Real Time](#) [Sync](#)

Status	Alias	IP Address	Port	Protocol	Description	
on	projet-infra-cluster	192.168.2.40	80	HTTP	site wordpress	
						Add

Save

4. Pour associer à un nom de domaine, il faut se rendre dans la section **Mappings**, ajouter avec **Add** et sélectionner le serveur qui héberge le site. Puis entrer le nom de domaine dans la zone **URI**

[General](#) [Web Servers](#) [Mappings](#) [Redirects](#) [Real Time](#) [Sync](#)

Squid Reverse Peer Mappings

Enable This URI ☒ If checked, then this URI (Uniform Resource Name) will be available for reverse proxy config.

Group Name
Name to identify this URI on Squid reverse proxy configuration. Example: URI1

Group Description
URI Group Description (Optional)

Peers

Apply these group mappings to the selected peers. Use CTRL + click to select multiple peers.

URI Settings Use URI row(s) below to add URL **regex** expression(s) to match (one per row).

URI
Enter URL **regex** to match. Click Info above for examples.

Add Add

Save

5. Vérifier que le flux HTTPS est autorisé sur le firewall dans les règles WAN, provenant de n'importe quelle source et à destination du port 443.
6. Afin de sécuriser les connexions entre le serveur web et le navigateur du client, il est nécessaire d'avoir un certificat SSL. Il permettent principalement de chiffrer les informations confidentielles du client et de sécuriser les données entre les serveurs.

Pour cela on se rend dans *System/Cert. Manager* puis dans la section CAs, cliquer sur **Add**.

Ajouter un nom, et la method ci-dessous :

System / Certificate Manager / CAs / Edit

CAs

Certificates

Certificate Revocation

Create / Edit CA

Descriptive name	Projet-Infra CA
Method	Create an internal Certificate Authority
Trust Store	<input type="checkbox"/> Add this Certificate Authority to the Operating System Trust Store When enabled, the contents of the CA will be added to the trust store so that they will be trusted by the operating system.
Randomize Serial	<input type="checkbox"/> Use random serial numbers when signing certificates When enabled, if this CA is capable of signing certificates then serial numbers for certificates signed by this CA will be automatically randomized and checked for uniqueness instead of using the sequential value from Next Certificate Serial.

Ensuite passer dans la section Certificates puis créer un certificat.

Add/Sign a New Certificate

Method	Create an internal Certificate
Descriptive name	Projet-Infra CERT

Internal Certificate

Certificate authority	Projet-Infra CA
Key type	RSA
	2048 The length to use when generating a new RSA key, in bits. The Key Length should not be lower than 2048 or some platforms may consider the certificate invalid.
Digest Algorithm	sha256 The digest method used when the certificate is signed. The best practice is to use an algorithm stronger than SHA1. Some platforms may consider weaker digest algorithms invalid
Lifetime (days)	3650 The length of time the signed certificate will be valid, in days. Server certificates should not have a lifetime over 398 days or some platforms may consider the certificate invalid.
Common Name	e.g. www.example.com
The following certificate subject components are optional and may be left blank.	
Country Code	FR
State or Province	France
City	Toulouse
Organization	projet-infra.fr

7. Activation et ajout du certificat SSL : Dans *Services/Squid Reverse Proxy*, activer **Enabled HTTPS Reverse Proxy** et au niveau du champ **Reverse SSL Certificate**, sélectionner le certificat nouvellement créé. Activer également **Enabled HTTP Reverse Proxy**

8. Pour activer complètement **Squid**, il est nécessaire de se rendre dans *Services/Squid Proxy*, et d'activer **Enable Squid Proxy**.

Serveur Nginx

Afin d'installer correctement nos applications web, nous avons choisi Nginx comme serveur HTTP. Pour mettre en place celui, munissez-vous tout d'abord d'une machine virtuelle avec Debian et installez.

1. Dans un premier temps, installer Nginx avec `apt-get install nginx` puis arrêter Apache s'il était déjà installé avec `systemctl stop apache2.service`.
2. Ensuite, décommenter la ligne `server_tokens off;` dans le fichier `/etc/nginx/nginx.conf` pour plus de sécurité, afin de ne pas envoyer les informations telles que le numéro de version de Nginx.

MariaDB

MariaDB est l'un des systèmes de gestion de base de données les plus utilisés sous Debian. À présent nous allons l'installer nous créer une **base de données** (*ynov*) ainsi qu'un **utilisateur** (*admin*) avec un **mot de passe** (*Passw0rd*) et lui accorder les **droits d'accès** à cette base.

1. Installation des paquets avec `apt-get install mariadb-server`.
2. Configuration de celle-ci avec `mysql_secure_installation`


```

root@SrvBDD:/home/user# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@SrvBDD:/home/user# █

```

3. Se connecter avec `mysql -u root -p` puis :

- créer une base de donnée *ynov* : `CREATE DATABASE blog;`
- créer un utilisateur *admin* : `CREATE USER "admin"@"localhost";`
- accorder lui un mot de passe : `SET password FOR "admin"@"localhost" = password('Passw0rd');`
- accorder lui les droits : `GRANT ALL ON ynov.* to 'admin'@'%' IDENTIFIED BY 'Passw0rd' WITH GRANT OPTION;` puis `FLUSH PRIVILEGES;` .

4. Par défaut MariaDB n'accorde l'accès uniquement à localhost.

On doit donc modifier le fichier `/etc/mysql/mariadb.conf.d/50-server.cnf` et passer la ligne `bind-address` à **0.0.0.0**.

Wordpress

Nous allons maintenant, installer Wordpress afin de pouvoir créer et gérer un site internet.

1. Installation :

- se placer dans le répertoire `/var/www`
- télécharger la dernière version de Wordpress avec `wget http://fr.wordpress.org/latest-fr_FR.tar.gz`
- décompresser l'archive `tar -xzf latest-fr_FR.tar.gz`
- supprimer l'archive `rm latest-fr_FR.tar.gz`

2. Configuration ::

- Installation de modules php avec `apt-get -y install php-cli php-mysql php-curl php-gd php-intl` :
 - PHP PDO est un connecteur pour MariaDB
 - PHP CURL est nécessaire pour certains plugins
 - PHP GD permettra d'opérer sur les images
 - PHP INTL est un support de l'internationalisation
- Vérification de la version de php avec `php -v`

```
root@SrvWeb2:/var/www# php -v
PHP 7.3.27-1~deb10u1 (cli) (built: Feb 13 2021 16:31:40) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.27, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.3.27-1~deb10u1, Copyright (c) 1999-2018, by Zend Technologies
root@SrvWeb2:/var/www#
```

• PHP FPM :

- Installation du module php-fpm permet la communication entre le serveur Nginx et PHP. On utilise `apt-get install -y php-fpm`
- Un **pool** est un fichier de configuration qui va dicter le nombre de thread à lancer, les permissions, etc...
On crée donc un pool wordpress : `/etc/php/7.3/fpm/pool.d/wordpress.conf`

```
[wordpress]
listen = /var/run/wordpress.sock

listen.owner = wordpress
listen.group = www-data

user = wordpress
group = www-data

pm = ondemand
pm.max_children = 10
pm.process_idle_timeout = 60s
pm.max_requests = 500
```

`listen` est l'interface d'écoute des requêtes. On utilisera donc un socket qui va nous permettre d'interfacer des processus entre eux sans passer par la couche réseau du système.

- Création du fichier `/etc/nginx/sites-available/wordpress.conf`

```

upstream php-wp {
server            unix:/var/run/wordpress.sock;
}

server {
    listen        80;
    listen        [::]:80;
    server_name    projet-infra.fr;

    root          /var/www/wordpress;

    index          index.php;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location = /favicon.ico {
        log_not_found off;
        access_log    off;
    }

    location = /robots.txt {
        allow          all;
        log_not_found off;
        access_log    off;
    }

    location ~ .php$ {
        include        fastcgi.conf;
        fastcgi_pass    php-wp;
    }

    location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {
        expires        max;
        log_not_found off;
    }
}

```

3. Activation :

Redémarrer le service nginx : `systemctl restart nginx`

si une erreur apparait de type `nginx.service: Failed to read PID from file /run/nginx.pid: Invalid argument`
lancer les commandes suivantes:

- `mkdir /etc/systemd/system/nginx.service.d`
- `printf "[Service]\nExecStartPost=/bin/sleep 0.1\n" > /etc/systemd/system/nginx.service.d/override.conf`
- `systemctl daemon-reload`
- `systemctl restart nginx`

4. Mise en route de Wordpress :

- Si vous avez mis en place votre reverse proxy, rendez vous sur votre navigateur (hors serveur) à l'adresse <http://www.projet-infra.fr>.
- Soit vous activez une redirection depuis un port non utilisé de pfSense (ex : 3222) vers le port 80 de votre cluster Heartbeat. Ainsi vous pouvez, depuis votre navigateur vous rendre sur wordpress avec 192.168.5.50:3222.

Vous arrivez sur cette page :



Vous devez saisir ci-dessous les détails de connexion à votre base de données. Si vous ne les connaissez pas, contactez votre hébergeur.

Nom de la base de données

Le nom de la base de données avec laquelle vous souhaitez utiliser WordPress.

Identifiant

Votre identifiant MySQL.

Mot de passe

Votre mot de passe de base de données.

Adresse de la base de données

Si localhost ne fonctionne pas, demandez cette information à l'hébergeur de votre site.

Préfixe des tables

Si vous souhaitez faire tourner plusieurs installations de WordPress sur une même base de données, modifiez ce réglage.

Entrer les informations créés dans MariaDB plus haut.

Si lorsque vous cliquez sur **Envoyer** il se produit une erreur, vérifier que vous avez bien configuré MariaDB.

Une fois connectée, vous arrivez sur cette page, vous pouvez à présent lancer l'installation.



C'est parfait ! Vous avez passé la première partie de l'installation. WordPress peut désormais communiquer avec votre base de données. Préparez-vous, il est maintenant temps de...

Donner un titre à votre site, saisissez un identifiant, un mot de passe et votre e-mail. Et c'est parti !



Bienvenue

Bienvenue dans la très célèbre installation en 5 minutes de WordPress ! Vous n'avez qu'à remplir les informations demandées ci-dessous et vous serez prêt à utiliser la plus extensible et puissante plateforme de publication de contenu au monde.

Informations nécessaires

Veuillez renseigner les informations suivantes. Ne vous inquiétez pas, vous pourrez les modifier plus tard.

Titre du site

Identifiant

Les identifiants ne peuvent utiliser que des caractères alphanumériques, des espaces, des tirets bas ("_"), des traits d'union ("-"), des points et le symbole @.

Mot de passe

[Masquer](#)

Forte

Important : Vous aurez besoin de ce mot de passe pour vous connecter. Pensez à le stocker dans un lieu sûr.

Votre e-mail

Vérifiez bien cette adresse e-mail avant de continuer.

Visibilité par les moteurs de recherche

☐ Demander aux moteurs de recherche de ne pas indexer ce site

Certains moteurs de recherche peuvent décider de l'indexer malgré tout.

Installer WordPress

Wekan

Nous allons à présent installer Wekan afin d'avoir un outil d'organisation de tâches. Cet outil va nous permettre de pouvoir créer des tâches (sous forme de cartes) et de les déplacer entre plusieurs colonnes afin d'établir un état d'avancement de celles-ci ou encore même, de les assigner à des membres spécifique.

1. On commence par installer **Node.js** avec `apt install curl git gcc g++ make` puis on installe le référentiel Node.js avec `curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -`.

Une fois effectuée on peut installer nodejs `apt-get install nodejs` et vérifier la version avec `node -v`

```
root@SrvWeb1:/home/user# node -v
v12.22.1
```

2. Puis on installe **Wekan** avec

```
wget https://github.com/wekan/wekan/releases/download/v0.63/wekan-0.63.tar.gz $ tar xf wekan-0.63.tar.gz
```

3. On se place dans le répertoire `cd /opt/bundle/programs/server` et on installe les dépendances Wekan avec `npm install`

4. A présent on se rend dans le répertoire `/opt/bundle` et on exécute l'application Wekan Node.js : `node main.js`
5. Configuration de Wekan SystemD Service avec `nano /etc/systemd/system/wekan.service` , on y rentre les instructions suivantes :

```
[Unit]
Description=Wekan Server
After=syslog.target
After=network.target

[Service]
Type=simple
Restart=always
StandardOutput=syslog
SyslogIdentifier=Wekan
User=wekan
Group=wekan
Environment=MARIADB_URL=mariadb://192.168.2.5/wekan
Environment=ROOT_URL=https://wekan.com
Environment=PORT=8000
WorkingDirectory=/bundle
ExecStart=/usr/bin/node /bundle/main.js

[Install]
WantedBy=multi-user.target
```

Haute disponibilité

Nous utilisons Heartbeat pour la mise en place de la haute disponibilité. En effet nous allons créer un clustering avec nos deux serveurs web qui partageront la même ip. Ainsi si le serveur primaire est arrêté, l'adresse virtuelle du cluster sera automatiquement redirigée sur le second serveur.

Les étapes ci-dessous devront être réalisées sur les deux serveurs.

1. On installe le package Heartbeat : `apt-get install heartbeat`
2. Seulement 3 fichiers de configurations sont nécessaires à la mise en place d'un cluster de serveur, ils se situent dans le dossier `/etc/heartbeat/`:
 - [ha.cf](#) :

```
# Indication du fichier de log
logfile /var/log/heartbeat.log

# Les logs heartbeat seront gérés par syslog, dans la catégorie daemon
logfacility daemon

# On liste tous les membres de notre cluster heartbeat (par les noms de préférences)
node SrvWeb1
node SrvWeb2

# On définit la périodicité de contrôle des nœuds entre eux (en seconde)
keepalive 1

# Au bout de combien de seconde un nœud sera considéré comme "mort"
deadtime 10

# Quelle carte réseau utiliser pour les broadcasts Heartbeat
bcast ens33

# Adresse LAN pour vérifier la connexion au net
ping 192.168.2.50

# Rebasculer automatiquement sur le primaire si celui-ci redevient vivant
auto_failback yes
```

- **authkeys** dont on sécurise les droits avec `chmod 600 /etc/heartbeat/authkeys` :

```
auth 1
1 sha1 SecretKey
```

- **haresources**, va permettre de définir l'action à effectuer lorsqu'un serveur passera de passif à actif:
SrvWeb1 192.168.2.40

3. Démarrage du serveur avec `systemctl start heartbeat`

Il est possible de vérifier la connexion dans `/var/log/heartbeat.log`

```
SrvWeb1 heartbeat: [15026]: info: Status update for node srvweb2: status active
1:58 harc(default)[15035]: info: Running /etc/ha.d/rc.d/status status
SrvWeb1 heartbeat: [15026]: info: Comm now up(): updating status to active
SrvWeb1 heartbeat: [15026]: info: Local status now set to: 'active'
SrvWeb1 heartbeat: [15026]: info: remote resource transition completed.
SrvWeb1 heartbeat: [15026]: info: remote resource transition completed.
SrvWeb1 heartbeat: [15026]: info: Local Resource acquisition completed. (none)
SrvWeb1 heartbeat: [15026]: info: srvweb2 wants to go standby [foreign]
SrvWeb1 heartbeat: [15026]: info: standby: acquire [foreign] resources from srvweb2
SrvWeb1 heartbeat: [15053]: info: acquire local HA resources (standby).
2:00 ResourceManager(default)[15067]: info: Acquiring resource group: srvweb1 192.168.2.40
2:00 /usr/lib/ocf/resource.d/heartbeat/IPaddr(IPaddr_192.168.2.40)[15094]: INFO: Resource is stopped
2:00 ResourceManager(default)[15067]: info: Running /etc/ha.d/resource.d/IPaddr 192.168.2.40 start
2:00 IPaddr(IPaddr_192.168.2.40)[15158]: INFO: Using calculated nic for 192.168.2.40: ens33
2:00 IPaddr(IPaddr_192.168.2.40)[15158]: INFO: Using calculated netmask for 192.168.2.40: 255.255.255.0
2:00 IPaddr(IPaddr_192.168.2.40)[15158]: INFO: eval ifconfig ens33:0 192.168.2.40 netmask 255.255.255.0 broadcast 192.168.2.255
2:00 /usr/lib/ocf/resource.d/heartbeat/IPaddr(IPaddr_192.168.2.40)[15146]: INFO: Success
SrvWeb1 heartbeat: [15053]: info: local HA resource acquisition completed (standby).
SrvWeb1 heartbeat: [15026]: info: Standby resource acquisition done [foreign].
SrvWeb1 heartbeat: [15026]: info: Initial resource acquisition complete (auto_failback)
SrvWeb1 heartbeat: [15026]: info: remote resource transition completed.
```

4. Notre ip c'est correctement changer en 192.168.2.40 sur le SrvWeb1 :

```

root@SrvWeb1:/# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:d3:0c:63 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.3/24 brd 192.168.2.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet 192.168.2.40/24 brd 192.168.2.255 scope global secondary ens33:0
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fed3:c63/64 scope link
        valid_lft forever preferred_lft forever
root@SrvWeb1:/#

```

5. Si on éteint le SrvWeb1, et que l'on regarde les logs du côté du SrvWeb2, on observe que l'ip 192.168.2.40 à bien était transférer sur le SrvWeb2:

```

:02:40 SrvWeb2 heartbeat: [7635]: WARN: node srvweb1: is dead
:02:40 SrvWeb2 heartbeat: [7635]: WARN: No STONITH device configured.
:02:40 SrvWeb2 heartbeat: [7635]: WARN: Shared disks are not protected.
:02:40 SrvWeb2 heartbeat: [7635]: info: Resources being acquired from srvweb1.
:02:40 SrvWeb2 heartbeat: [7635]: info: Link srvweb1:ens33 dead.
:02:40 SrvWeb2 heartbeat: [8059]: info: No local resources [/usr/share/heartbeat/ResourceManager listkeys srvweb2] to acquire.
1_19:02:40 harc(default)[8058]: info: Running /etc/ha.d/rc.d/status status
1_19:02:40 mach_down(default)[8086]: info: Taking over resource group 192.168.2.40
1_19:02:40 ResourceManager(default)[8111]: info: Acquiring resource group: srvweb1 192.168.2.40
1_19:02:40 /usr/lib/ocf/resource.d/heartbeat/IPaddr(IPaddr_192.168.2.40)[8138]: INFO: Resource is stopped
1_19:02:40 ResourceManager(default)[8111]: info: Running /etc/ha.d/resource.d/IPaddr 192.168.2.40 start
1_19:02:40 IPaddr(IPaddr_192.168.2.40)[8202]: INFO: Using calculated nic for 192.168.2.40: ens33
1_19:02:40 IPaddr(IPaddr_192.168.2.40)[8202]: INFO: Using calculated netmask for 192.168.2.40: 255.255.255.0
1_19:02:40 IPaddr(IPaddr_192.168.2.40)[8202]: INFO: eval ifconfig ens33:0 192.168.2.40 netmask 255.255.255.0 broadcast 192.168.2.255
1_19:02:40 /usr/lib/ocf/resource.d/heartbeat/IPaddr(IPaddr_192.168.2.40)[8190]: INFO: Success
1_19:02:40 mach_down(default)[8086]: info: /usr/share/heartbeat/mach_down: nice_failback: foreign resources acquired
1_19:02:40 mach_down(default)[8086]: info: mach_down takeover complete for node srvweb1.
:02:40 SrvWeb2 heartbeat: [7635]: info: mach_down takeover complete.

```

6. On utilisera la commande `systemctl enable heartbeat` afin de lancer le processus automatiquement au démarrage du server.

Réplication des données : Il est possible de répliquer ses données de façon synchroniser entre les deux serveurs web avec [DRBD](#)

Installer une application web :

Pour installer une application dans notre serveur web, il suffit de :

1. Se placer dans le répertoire `cd /var/www`
2. Télécharger l'application voulu avec `wget nomapplication.tar.gz`
3. Extraire l'application avec `tar -xzf nomapplication.tar.gz`
4. Créer un pool php, on peut par exemple copier celui ci-dessous et le modifier en remplaçant :
 - *nomdusite* par le nom du site créé
 - *choisirUser* par le nom de l'utilisateur admin


```
[nomdusite]
listen = /var/run/nomdusite.sock

listen.owner = choisirUser
listen.group = www-data

user = choisirUser
group = www-data

pm = ondemand
pm.max_children = 10
pm.process_idle_timeout = 60s
pm.max_requests = 500
```

5. Mettre à jour la configuration nginx en créant un fichier dans */etc/nginx/sites-available/nomdusite*, on peut par exemple copier celui-ci et le modifier en remplaçant :
- *nomapplication* par le nom de l'application (identique à l'emplacement où vous avez téléchargé votre application dans */var/www/*)
 - *nomdedomaine* par le nom de domaine voulu ou que vous possédez.

```
server {
    listen 80;
    listen [::]:80;
    root /var/www/nomapplication;
    index index.html index.htm;
    server_name nomdedomaine;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

6. Pour finir relancer les services nginx avec `systemctl restart nginx`.

Contact

François BONNIN - francois.bonnin@ynov.com

Project Link: <https://github.com/Raimana92/Projet-Infra-SI>