BTS Services informatiques aux organisations - SISR

Session 2026

E4 – Support et mise à disposition de services informatiques

Coefficient 4

DESCRIPTION DE LA REALISATION PROFESSIONNELLE

NOM et prénom du candidat : Liana MEGAZZINI	

Contexte de la réalisation professionnelle

Dans le cadre de la réalisation de mon portfolio pour l'épreuve du BTS SIO, j'ai été amenée à déployer un site web personnel basé sur le CMS WordPress au sein de mon infrastructure de maquettage.

Ce projet avait pour objectif de rendre mon portfolio accessible depuis Internet tout en respectant les bonnes pratiques de sécurité et d'architecture réseau.

L'infrastructure étant entièrement simulée, j'ai dû concevoir une solution complète intégrant :

- Un serveur web sous Linux
- Une base de données
- Une segmentation réseau sécurisée
- Une publication via un reverse proxy

Plusieurs contraintes ont été identifiées :

· Nécessité d'exposer le site sur Internet tout en limitant les risques de sécurité
· Gestion des flux entre réseau interne et externe
· Mise en place d'un accès sécurisé en HTTPS
· Conflits potentiels entre services réseau (DNS interne et reverse proxy)

Le projet consistait donc à déployer un serveur WordPress fonctionnel, sécurisé et accessible publiquement, tout en respectant une architecture réseau de type professionnel.

Intitulé de la réalisation professionnelle :

<u>Déploiement et sécurisation d'un site WordPress</u>

Période de réalisation : **Lieu :**
Modalité : x Individuelle ☐ En équipe

Principale(s) activité(s) concernée(s) :

- **Développer la présence en ligne de l'organisation :**
  - Participer à la valorisation de l'image de l'organisation sur les médias numériques
  - Référencer les services en ligne de l'organisation et mesurer leur visibilité
  - Participer à l'évolution d'un site Web exploitant les données de l'organisation
  - Veiller à la qualité de l'expérience utilisateur

- **Organiser son développement professionnel :**
  - Mettre en place son environnement d'apprentissage personnel
  - Mettre en œuvre des outils et stratégies de veille informationnelle
  - Gérer son identité professionnelle
  - Développer son projet professionnel

- **Mettre à disposition des utilisateurs un service informatique :**
  - Réaliser les tests d'intégration et d'acceptation d'un service
  - Déployer un service

Conditions de réalisation :

- **Ressources présentes :**
  - Infrastructure virtualisée (VM Debian)
  - Reverse proxy Traefik
  - DNS interne (Bind)
  - Réseau segmenté avec VLAN
  - Accès Internet

- **Résultats attendus :**
  - Déploiement d'un site Wordpress fonctionnel
  - Mise en place d'une base de données MariaDB
  - Intégration du serveur dans une DMZ
  - Accessibilité du site depuis Internet
  - Mise en place d'un certificat HTTPS (Let's Encrypt)
  - Absence de conflits entre les différents services réseau

- **Durée de réalisation**
  - Janvier 2026 – Février 2026

Modalités d'accès à cette réalisation professionnelle.

URL : https://pllm.fr Mot de passe : sioBTSsisr26

Partie 1 – Procédure de mise en œuvre.

Outils et technologies utilisés :

- Debian (VM)
- WordPress
- MariaDB
- PHP
- Treafik (reverse proxy)
- Let's Encrypt (certificat SSL)
- VLAN / DMZ
- DNS Bind

Organisation du projet (travail autonome) :

- Travail réalisé en autonomie dans un environnement de maquettage
- Recherches de documentations techniques (WordPress, Traefik, HTTPS)
- Tests et validation progressive des configurations

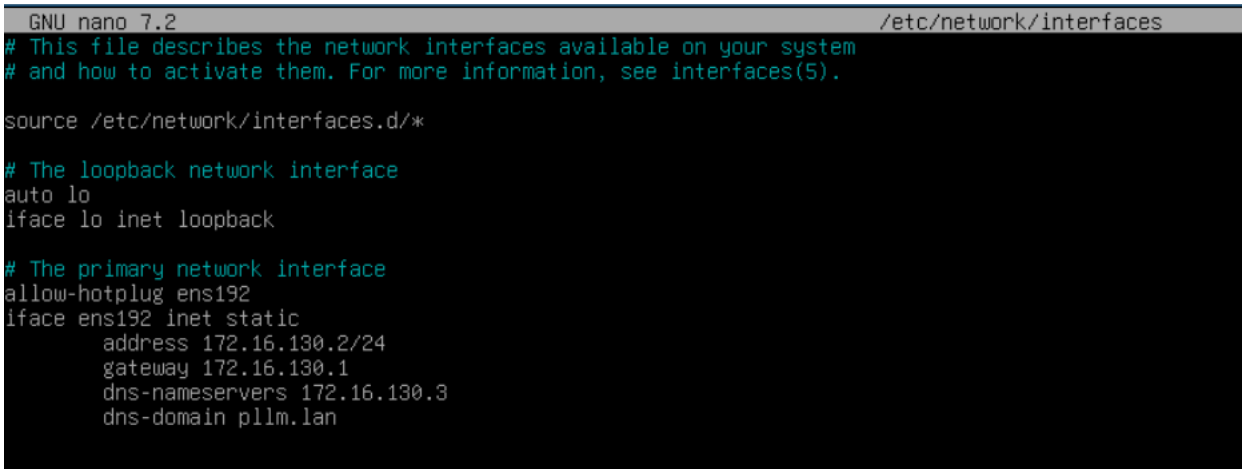
Phases de mise en œuvre :

1. **Création de la machine virtuelle Debian**

Afin d'héberger mon site j'ai créé une machine virtuelle debian via une ISO. Puisque cette VM était destinée à exécuter un WordPress, lors de l'installation sans interface graphique j'ai ajouté la prise en charge du SSH pour l'administration du serveur et l'option de serveur web pour définir sa fonction.

2. **Configuration réseau**

J'ai commencé par modifier le fichier de configuration réseau de la machine pour lui attribuer une IP fixe, définir son DNS et sa gateway et j'ai choisi de la placer dans un VLAN de type DMZ afin d'isoler le serveur du réseau interne et limiter les risques de sécurité. Également pour des raisons de sécurité et de continuité de service, le DNS n'a pas été installé directement sur la VM du site mais dans une VM Bind à part se trouvant également sur le VLAN DMZ.



3. **Installation de l'environnement web**

J'ai ensuite poursuivi en mettant à jour la VM, puis j'ai installé les paquets essentiels pour pouvoir construire et faire fonctionner mon site. J'ai donc installé les paquets PHP pour la gestion du HTML du site, puis j'ai installé Apache pour le rôle de serveur web.

4. **Déploiement de la base de données**

J'ai ensuite installé les paquets de la base de données MariaDB et configuré celle-ci en créant un utilisateur dédié et une base pour le site :

```
Thanks for using MariaDB!
root@wordpress:~# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.11.14-MariaDB-0+deb12u2 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE wp_portfolio;
Query OK, 1 row affected (0,015 sec)

MariaDB [(none)]> CREATE USER 'wpadmin'@'localhost' IDENTIFIED BY 'btssio89$';
Query OK, 0 rows affected (0,010 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wp_portfolio.* TO 'wpadmin'@'localhost';
Query OK, 0 rows affected (0,004 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,004 sec)

MariaDB [(none)]> EXIT
Bye
root@wordpress:~#
```

5. **Installation et configuration de WordPress**

Par la suite, j'ai téléchargé, dézippé puis installé le logiciel WordPress, sans oublier d'attribuer les droits nécessaires pour pouvoir modifier le fichier installé.

6. **Configuration serveur HTTP Apache2**

Pour que le site soit accessible en ligne j'ai configuré le serveur HTTP Apache2. Pour cela, j'ai commencé par créer un fichier de configuration de mon WordPress dans le répertoire ***/etc/apache2/sites-available*** que j'ai ensuite configuré de la façon suivante :

```
 GNU nano 7.2                                    /etc/apache2/sites-available/pllm.fr.conf
<VirtualHost *:80>
        ServerName pllm.fr
        ServerAlias www.pllm.fr

        DocumentRoot /var/www/wp-portfolio

        <Directory /var/www/wp-portfolio>
                Options FollowSymLinks
                AllowOverride All
                Require all granted
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/pllm.fr_error.log
        CustomLog ${APACHE_LOG_DIR}/pllm.fr_access.log combined
RewriteEngine on
RewriteCond %{SERVER_NAME} =pllm.fr [OR]
RewriteCond %{SERVER_NAME} =www.pllm.fr
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>
```

Une fois ce fichier rempli, j'ai exécuté une commande pour supprimer le fichier de configuration ***000-default.conf*** qui existe par défaut à l'installation de Apache2 pour éviter les conflits éventuels entre celui-ci et celui de mon site. Ensuite j'ai rendu actif mon site et démarré le service Apache2.

7. **Publication via reverse proxy Traefik**

La VM étant placée dans une DMZ, afin que le site soit accessible publiquement sur Internet, j'ai géré la redirection web en mettant en place un reverse proxy basé sur Traefik.

J'ai donc créé une VM debian sur laquelle j'ai installé Traefik avant de lui attribuer ses paramètres réseau en le plaçant également dans le VLAN DMZ :

```
  GNU nano 7.2                                                /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens192
iface ens192 inet static
        address 172.16.130.5
        netmask 255.255.255.0
        gateway 172.16.130.1
        dns-nameservers 172.16.110.2 8.8.8.8
```

J'ai ensuite configuré le fichier de Traefik lui-même pour définir les ports d'écoute (80 en HTTP et 443 en HTTPS) :

```
  GNU nano 7.2                                                        traefik.yml
log:
 level: INFO

entryPoints:
 web:
  address: ":80"
  http:
   redirections:
    entryPoint:
     to: websecure
     scheme: https

 websecure:
  address: ":443"

providers:
 file:
  filename: /etc/traefik/dynamic.yml
  watch: true

certificatesResolvers:
 letsencrypt:
  acme:
   email: admin@pllm.fr
   storage: /etc/traefik/acme.json
   httpChallenge:
    entryPoint: web

api:
 dashboard: true
 insecure: true
```

Puis j'ai paramétré le fichier dynamic.yml pour définir les services qui seront pris en charge par le traefik et leurs routes respectives en déclarant mon site WordPress :

```
liana-service:
 loadBalancer:
  servers:
   - url: "https://172.16.130.2"
  serversTransport: wordpress-transport_
```

Le Traefik que j'ai mis en place gère plusieurs services, mais ici je ne détaillerais que la partie site Web.

8. **Mise en place du HTTPS avec Let's Encrypt**

Afin de sécuriser la page Web de mon Wordpress, j'ai effectué la configuration d'un certificat SSL permettant de sécuriser les échanges en HTTPS. Pour cela, j'ai choisi d'utiliser Let's Encrypt pour générer un certificat valide.

9. **Résolution d'un problème de configuration réseau**

Lors de la mise en place de Traefik, un conflit de résolution est apparu entre le DNS interne (Bind) et le reverse proxy.

Pour résoudre ce problème, j'ai configuré un serveur de transport dédié dans le fichier dynamic.yml de Traefik, permettant d'assurer une communication correcte entre les services et d'éviter les erreurs liées aux certificat Let's Encrypt que j'ai mis en place et les propres certificats de Traefik :

```
 GNU nano 7.2                                                dynamic.yml *
http:
 serversTransports:
  wordpress-transport:
   insecureSkipVerify: true
```

La sécurité a été prise en compte via la segmentation réseau, l'utilisation d'un reverse proxy et le chiffrement des communications en HTTPS.

10. **Architecture de l'infrastructure Wordpress**

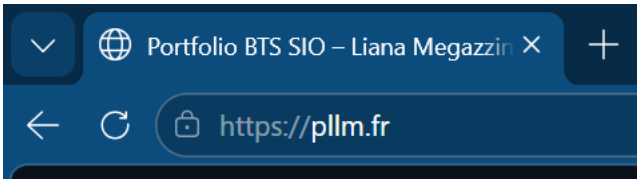
Partie 2 – Validation.

- Vérification de l'accessibilité du site depuis Internet et du bon fonctionnement du site WordPress :

En tapant l'URL du site dans un navigateur, celui-ci est bien accessible comme voulu.

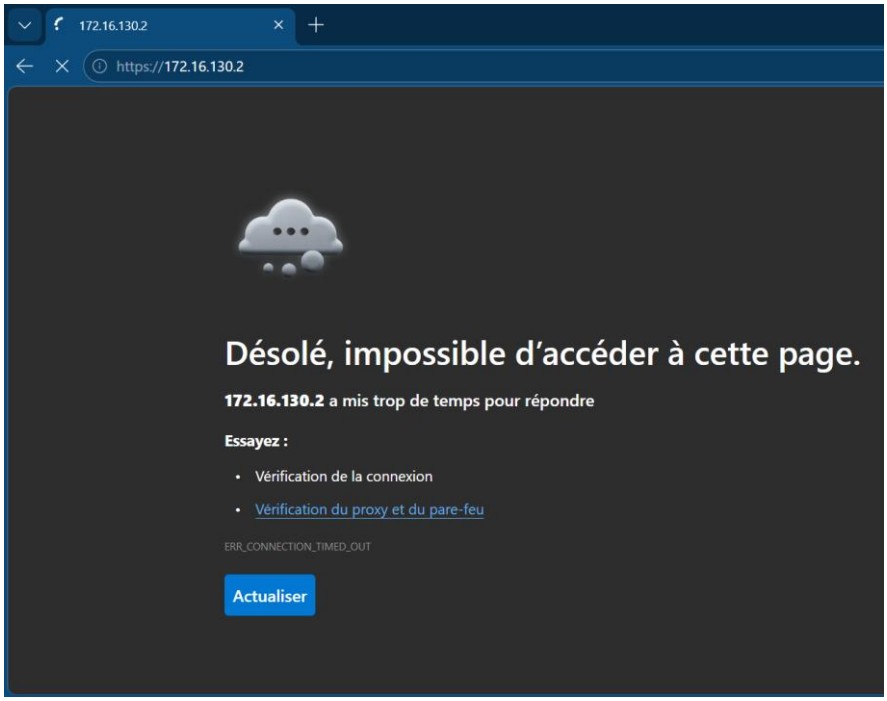
- Vérification du certificat HTTPS :

Même sans inclure dans l'URL du site le HTTPS en tapant seulement « pllm.fr » le site redirige bien sur une page en HTTPS grâce au certificat Let's Encrypt.



- Validation de la redirection via Traefik et vérification de l'isolation réseau de la VM WordPress :

Grâce à Traefik, le site est bien accessible depuis Internet mais n'expose pas la VM directement puisqu'elle à été placée en DMZ. Ainsi, en tapant l'adresse IP de la machine WordPress dans un navigateur, la page est inaccessible.



Ces tests ont permis de confirmer que le site est fonctionnel, sécurisé et accessible conformément aux objectifs initiaux.

Partie 3 – Veille technologique.

D’autres solutions auraient pu être envisagées :

- Hébergement sur un service externe à l’infrastructure maquette (OVH ou autre)
- Utilisation d’un autre reverse proxy que Traefik (Nginx ou autre)
- Déploiement via conteneurisation (Docker)

Ces solutions ont été écartées en raison :

- La volonté de maîtriser l’infrastructure complète
- L’objectif pédagogique de mise en place d’une architecture réseau centralisée
- De la complexité supplémentaire de certaines solutions

Le choix de Traefik et d’une infrastructure virtualisée permet de reproduire un environnement professionnel réaliste, tout en facilitant la gestion du HTTPS et du routage des services.