BTS SIO

DOCUMENTATION TECHNIQUE FICHE n°2

Épreuve E6

Alexandre BONGRAND
07/05/2025

Suivi des modifications

Version	*Référence*	*Auteur*	*Date*	*Commentaires*
A	Documentation technique fiche n°2	Alexandre	30/04/2025	Création

Objet :

Documentation technique fiche n°2 - Projet E6

Diffusion :

BTS SIO – Étudiants BTS SIO.

Développement :

Table des matières

PFSENSE

Ajout du VLAN 50 DMZ

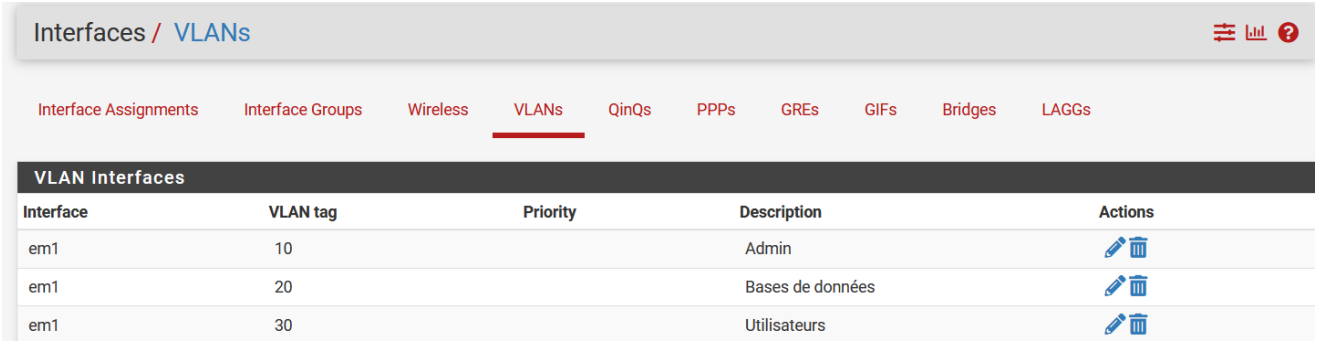
Dans l'infrastructure, 3 VLAN étaient déjà paramétré. Pour l'ajout de la DMZ, j'ai créé un nouveau VLAN, qui ne sera pas utilisable en dehors des services web installés dessus. Les règles du pare feu laissent passer uniquement les ports pour les bases de données, ou le trafic HTTPS. Lorsqu'une requête HTTPS arrive à mon pfSense, elle est directement redirigée vers mon reverse proxy qui est dans ma DMZ et que sécurise mon réseau.

Ancienne configuration :

VLAN 10 - Administration : regroupe le serveur Windows Server ainsi que le poste de l'administrateur.

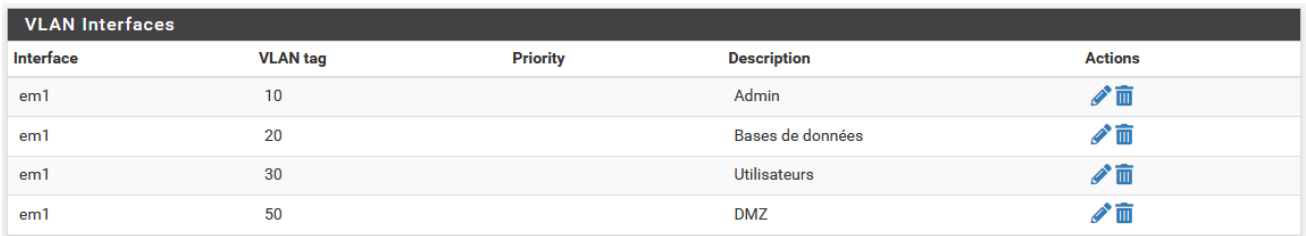
VLAN 20 - Bases de données : contient le serveur Debian hébergeant les bases de données ainsi que les services internes tels que Wazuh et Zabbix.

VLAN 30 - Utilisateurs : dédié aux postes clients des salariés de l'entreprise BONGRAND.



Ajout du nouveau VLAN 50 :

VLAN 50 – DMZ : Dédié à mes serveurs et services web.



Mise en place des règles de Pare-feu

VLAN 10

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓	0/6.93 MiB	*	*	*	VLAN10 Address	443 80	*	*		Anti-Lockout Rule	⚙
✓	0/0 B	IPv4 TCP	192.168.50.250	80 - 443	VLAN10 subnets	*	*	none			
✓	0/0 B	IPv4 TCP/ UDP	VLAN10 subnets	*	192.168.10.250	53 (DNS)	*	none			
✓	0/0 B	IPv4 TCP	VLAN30 subnets	*	192.168.10.250	445 (MS DS)	*	none		partages windows	
✓	0/0 B	IPv4 TCP	VLAN10 subnets	*	192.168.20.250	1514	*	none		agent wazzuh	
✓	0/0 B	IPv4 TCP	VLAN10 subnets	*	192.168.20.250	10051	*	none		agent zabbix	
✓	0/0 B	IPv4 TCP	VLAN10 subnets	*	192.168.20.250	62354	*	none		agent glpi	
✓	0/0 B	IPv4 TCP	VLAN10 subnets	*	192.168.20.250	3306	*	none		bases de données	
✓	0/47 KiB	IPv4 TCP	VLAN10 subnets	*	192.168.50.250	443 (HTTPS)	*	none			
✓	11/4.11 GiB	IPv4 *	VLAN10 subnets	*	*	*	*	none		Default allow LAN to any rule	
✓	0/0 B	IPv4 ICMP any	*	*	*	*	*	none			
✓	0/0 B	IPv6 *	VLAN10 subnets	*	*	*	*	none		Default allow LAN IPv6 to any rule	
✗	0/640 B	IPv4 *	*	*	*	*	*	none			

VLAN 20

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓	0/121.12 MiB	IPv4 TCP	192.168.20.250	*	*	80 - 443	*	none			
✓	0/0 B	IPv4 TCP	VLAN30 subnets	*	192.168.20.250	3306	*	none		bases de données	
✓	0/0 B	IPv4 TCP	192.168.50.250	*	192.168.20.250	3306	*	none			
✓	0/0 B	IPv4 TCP	192.168.20.250	*	192.168.50.250	3306	*	none			
✓	0/227 KiB	IPv4 TCP	192.168.20.250	*	*	10050 - 10051	*	none			
✓	0/0 B	IPv4 TCP	VLAN30 subnets	1514 - 1515	192.168.20.250	1514 - 1515	*	none			
✓	0/0 B	IPv4 TCP	192.168.50.250	1514 - 1515	192.168.20.250	1514 - 1515	*	none			
✓	0/0 B	IPv4 TCP	VLAN10 subnets	1514 - 1515	192.168.20.250	1514 - 1515	*	none			
✓	0/428 KiB	IPv4 TCP/UDP	192.168.20.250	*	This Firewall (self)	53 (DNS)	*	none			
✓	0/0 B	IPv4 ICMP any	*	*	*	*	*	none			
✗	0/1.12 MiB	IPv4 *	*	*	*	*	*	none			

VLAN 30

Floating | WAN | VLAN10 | VLAN20 | VLAN30 | VLAN50

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓	0/15 KiB	IPv4 TCP/UDP	VLAN30 subnets	*	VLAN10 subnets	88	*	none			
✓	0/0 B	IPv4 TCP	192.168.50.250	80 - 443	VLAN30 subnets	*	*	none			
✓	0/5.91 MiB	IPv4 TCP/UDP	VLAN30 subnets	*	192.168.50.250	*	*	none			
✓	0/325 KiB	IPv4 TCP	VLAN30 subnets	*	192.168.10.250	389 (LDAP)	*	none			
✓	4/131 KiB	IPv4 UDP	VLAN30 subnets	*	192.168.10.250	53 (DNS)	*	none			
✓	4/68.30 MiB	IPv4 TCP/UDP	VLAN30 subnets	*	*	443 (HTTPS)	*	none			
✓	0/0 B	IPv4 TCP	VLAN30 subnets	*	192.168.50.250	443 (HTTPS)	*	none			
✓	0/0 B	IPv4 UDP	VLAN30 subnets	*	This Firewall (self)	67 - 68	*	none			
✓	0/0 B	IPv4 TCP	VLAN30 subnets	*	192.168.20.250	3306	*	none			
✓	0/0 B	IPv4 TCP	VLAN30 subnets	*	192.168.20.250	10051	*	none			
✓	0/0 B	IPv4 TCP	VLAN30 subnets	*	192.168.20.250	62354	*	none			
✓	0/108 KiB	IPv4 TCP	VLAN30 subnets	*	192.168.10.250	445 (MS DS)	*	none			
✓	0/0 B	IPv4 TCP	VLAN30 subnets	*	192.168.20.250	1514	*	none			
✓	0/3 KiB	IPv4 ICMP any	*	*	*	*	*	none			
✗	0/621 KiB	IPv4 *	*	*	*	*	*	none			

VLAN 50

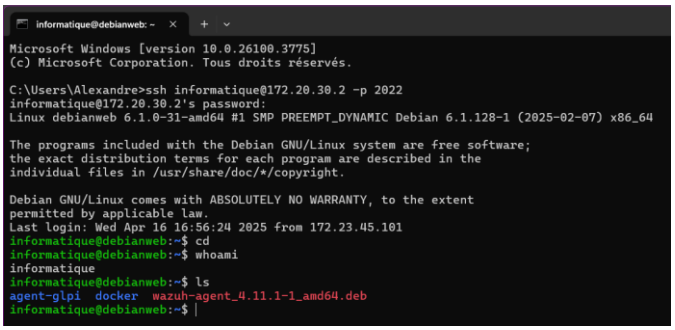
Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓	0/0 B	IPv4 TCP/UDP	*	*	192.168.50.250	80 - 443	*	none			
✓	0/35.53 MiB	IPv4 TCP	192.168.50.250	*	*	80 - 443	*	none		apt update	
✓	0/0 B	IPv4 TCP/UDP	192.168.50.250	80 - 443	*	*	*	none			
✓	0/232 KiB	IPv4 TCP/UDP	192.168.50.250	*	This Firewall (self)	53 (DNS)	*	none			
✓	0/57.99 MiB	IPv4 TCP	192.168.50.250	*	192.168.20.250	*	*	none			
✓	0/0 B	IPv4 TCP	192.168.20.250	62354	192.168.50.250	62354	*	none			
✓	0/0 B	IPv4 TCP	192.168.50.250	62354	192.168.20.250	62354	*	none			
✓	0/0 B	IPv4 TCP	192.168.20.250	10051	192.168.50.250	10051	*	none			
✓	0/0 B	IPv4 TCP	192.168.50.250	10051	192.168.20.250	10051	*	none			
✓	0/0 B	IPv4 TCP	192.168.50.250	3306	192.168.20.250	3306	*	none			
✓	0/0 B	IPv4 TCP	192.168.20.250	3306	192.168.50.250	*	*	none			
✓	0/0 B	IPv4 TCP/UDP	192.168.50.250	1514	192.168.20.250	1514	*	none			
✓	0/0 B	IPv4 TCP/UDP	192.168.20.250	1514	192.168.50.250	1514	*	none			
✓	0/15 KiB	IPv4 ICMP any	*	*	*	*	*	none			
✗	0/57 KiB	IPv4 *	*	*	*	*	*	none			

SERVEUR DEBIAN WEB

J'ai décidé d'installer un serveur Debian 12 afin d'y héberger les services web, Nginx, Nextcloud, le site de e-commerce PrestaShop et HAproxy.

J'ai choisi Debian car il s'agit d'un système d'exploitation open source, reconnu pour sa légèreté et sa fiabilité, ce qui en fait un excellent choix pour un serveur.
L'installation est simple et rapide.



Une fois notre serveur en fonctionnement, je peux installer les services nécessaires pour l'entreprise BONGRAND en commençant par Docker.

Installation Docker et Portainer

L'installation de Docker se fait directement en ligne de commande sur notre machine hôte. C'est une installation simple qui nécessite que quelques lignes de commandes.

Une fois notre Debian mise à jour et curl d'installé, nous allons installer GPG de Docker à fin de permettre aux utilisateurs de communiqué de manière sûre avec le dépôt.

```
install -m 0755 -d /etc/apt/keyrings
```

```
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Nous ajoutons l'adresse du dépôt Docker dans les sources.

Nous devons refaire un `apt update` car nous avons ajouté un dépôt.

Une fois cette update faite, nous pouvons lancer l'installation de Docker et Docker-compose en plugin, il nous sera utile pour créer des multi-conteneurs, tel qu'un WordPress et sa base de données.

```
apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Ajouter notre utilisateur dans le groupe Docker.

```
usermod -aG docker « utilisateur »
```

Portainer

À présent que nous avons Docker d'installé, nous pouvons installer Portainer, ce qui va nous permettre de gérer nos machines virtuelles plus facilement et grâce à une interface web.

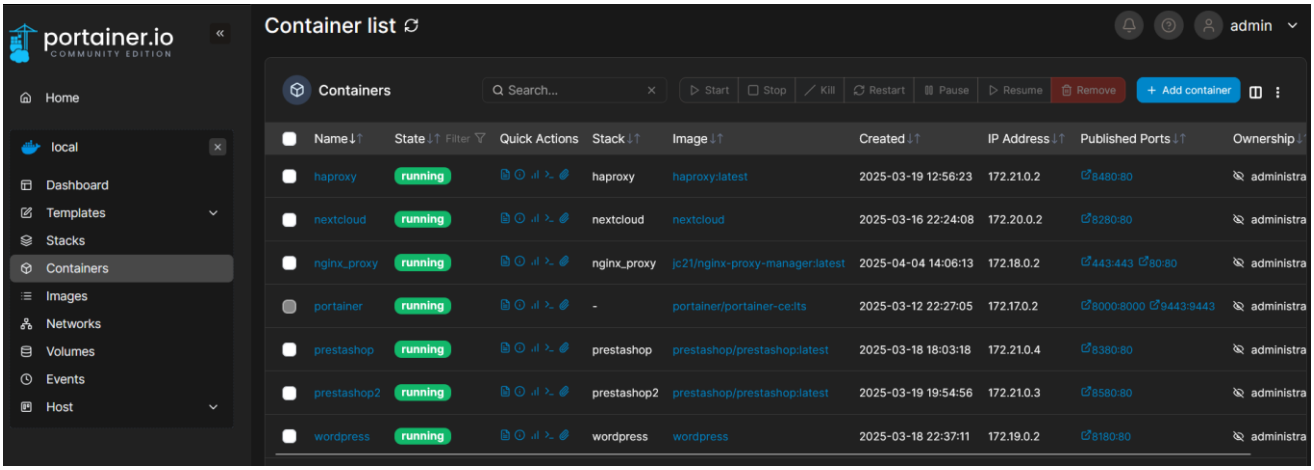
Nous allons créer un volume Docker nommé "portainer_data" pour stocker les données de Portainer.

```
Docker volume create portainer_data
```

Lancer le conteneur Portainer.

```
docker run -d --name=portainer --restart=always -p 8000:8000 -p 9443:9443 -v /var/run/docker.sock:/var/run/docker.sock -v
 portainer_data:/data portainer/portainer-ce:latest
```

Maintenant que nous avons Portainer d'installé, nous pouvons déployer des services plus simplement grâce à des stacks



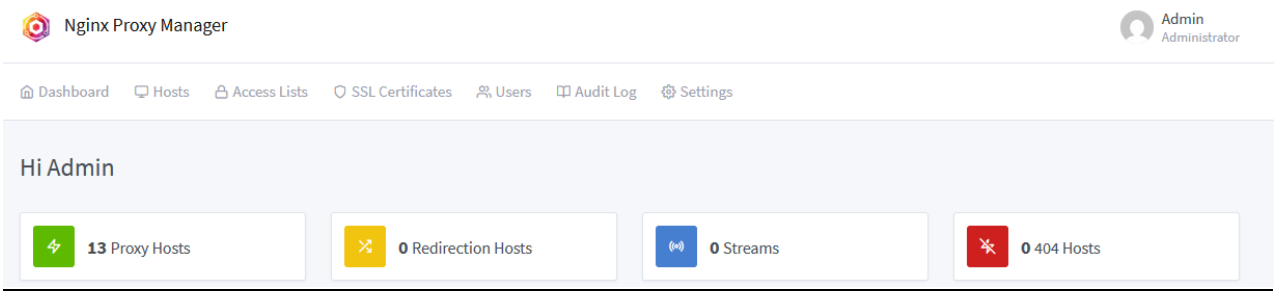
Nginx

L'installation de Nginx a été réalisée à l'aide d'un fichier Docker-compose.yml, ce qui permet de déployer la solution en seulement cinq minutes, le temps nécessaire pour télécharger l'image.

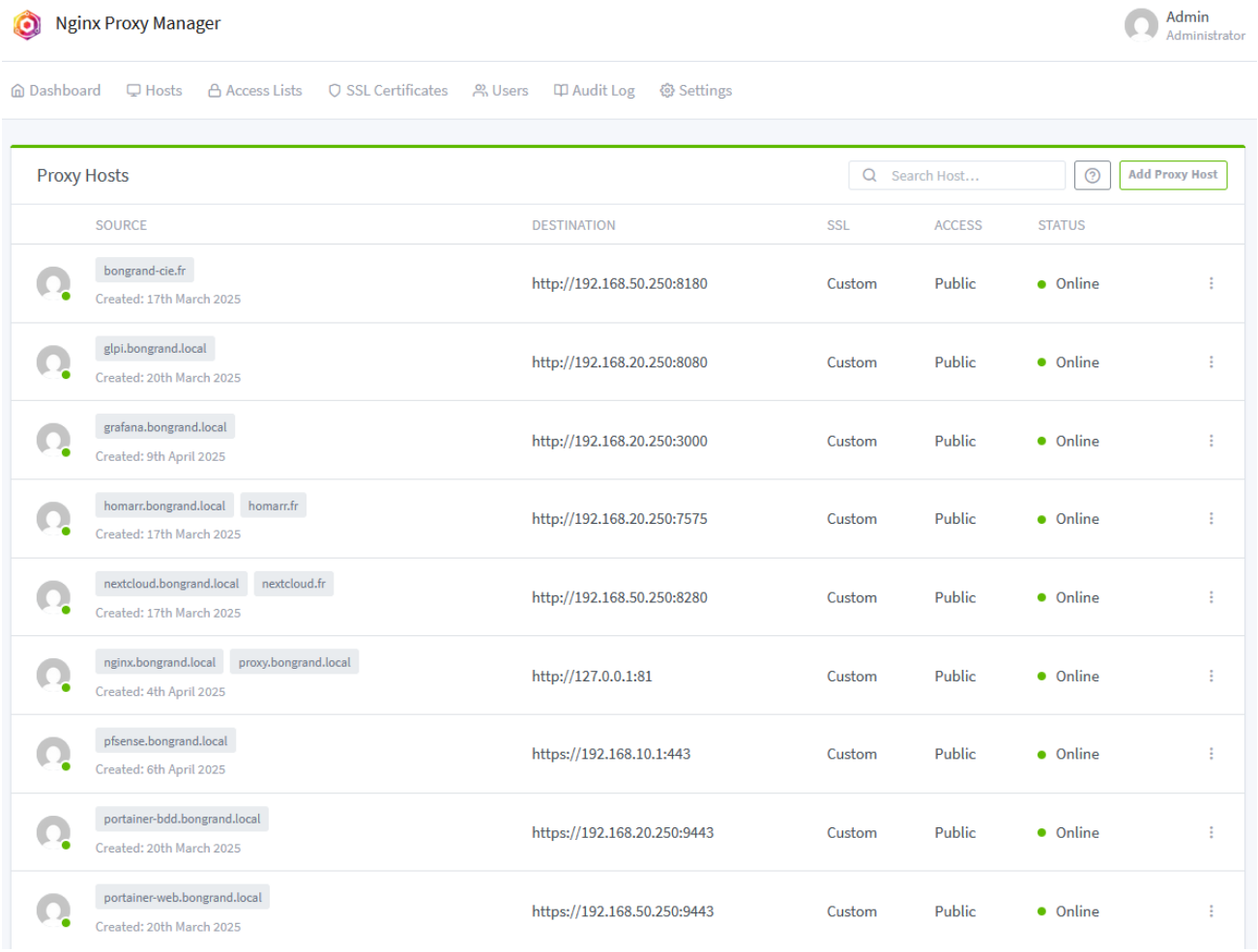
Docker-compose.yml :

```
 GNU nano 7.2                                                docker-com
services:
  app:
    image: 'jc21/nginx-proxy-manager:latest'
    container_name: nginx_proxy
    restart: unless-stopped
    ports:
      # These ports are in format <host-port>:<container-port>
      - '80:80' # Public HTTP Port
      - '443:443' # Public HTTPS Port
      #- '81:81' # Admin Web Port
      # Add any other Stream port you want to expose
      # - '21:21' # FTP

    environment:
      DB_MYSQL_HOST: "192.168.20.250"
      DB_MYSQL_PORT: 3306
      DB_MYSQL_USER: "nginx_user"
      DB_MYSQL_PASSWORD: "root"
      DB_MYSQL_NAME: "nginx_proxy"

    volumes:
      - ./data:/data
      - ./letsencrypt:/etc/letsencrypt
```

Interface web :



Configuration Nginx :



Cette configuration de mon Nginx Proxy Manager me permet de sécuriser tous mes sites en HTTPS, grâce aux certificats SSL configurés dessus. En revanche, les communications entre le proxy et mes serveurs internes se font en HTTP, afin de limiter l'utilisation des ressources et de simplifier la configuration interne.

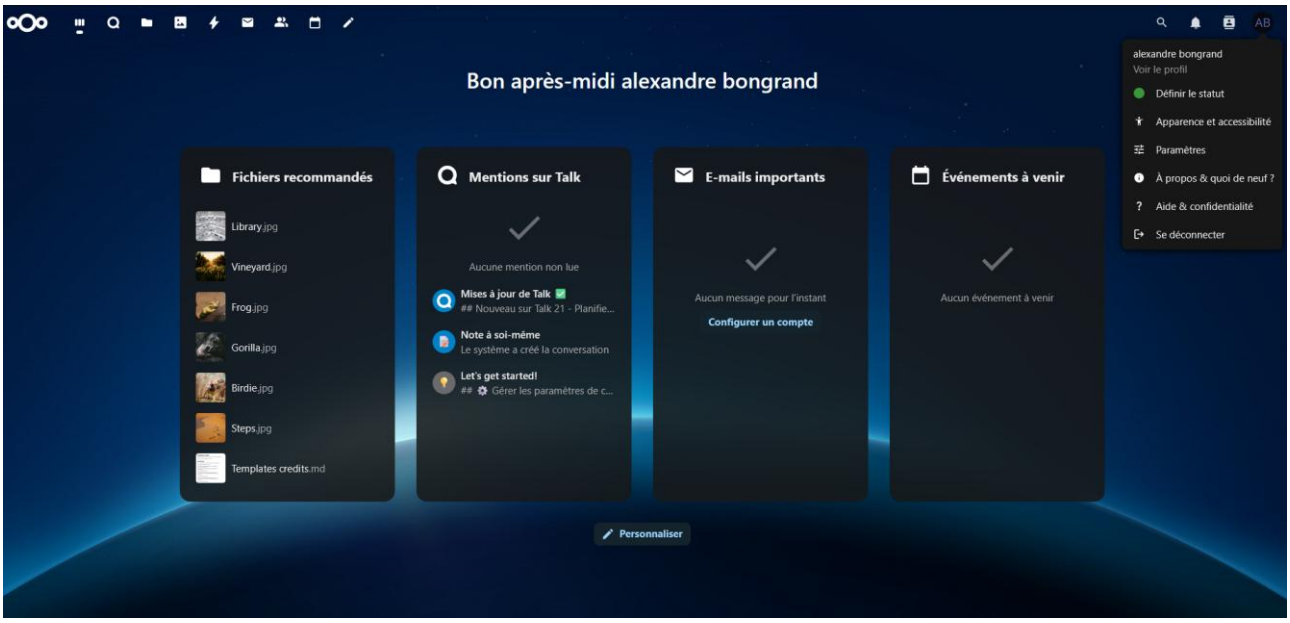
Nextcloud

Idem que pour Nginx, l'installation se fait avec un docker-compose.yml. Facilité d'installation et en cas de défaillance, on peut supprimer notre containeur, le refaire, nous retrouverons tout, étant donné que notre base de données est externalisée.

Docker-compose.yml :

```
services:
  nextcloud:
    image: nextcloud
    restart: always
    ports:
      - 8280:80
    volumes:
      - ./data:/var/www/html
    environment:
      - MYSQL_PASSWORD=root
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud_user
      - MYSQL_HOST=192.168.20.250
```

Interface web :



J'ai ajouté une connexion via LDAP/AD pour que les utilisateurs du domaine puisse avoir leur propre partage de fichiers sur le web.

Compte LDAP :

Nom d'affichage	Nom du compte	Mot de passe	E-mail	Groupes
JB Jean Bon	16588C7B-DCF7-4E36-95...			
N nextcloud	33BE509B-1601-4591-86...			
AB alexandre bongrand	5EB4C09C-C4EE-4CF8-9E...		alexandre-bongrand@bo...	
A admin	admin			admin

PrestaShop

L'entreprise BONGRAND souhaitait disposer d'un site e-commerce. Pour répondre à cette demande, j'ai installé une instance de PrestaShop avec 2 serveurs. Ces serveurs sont placés derrières un HAProxy, utilisé comme répartiteur de charge (load balancer), afin de prévenir toute surcharge du site et de garantir une accessibilité continue, même en cas de forte affluence.

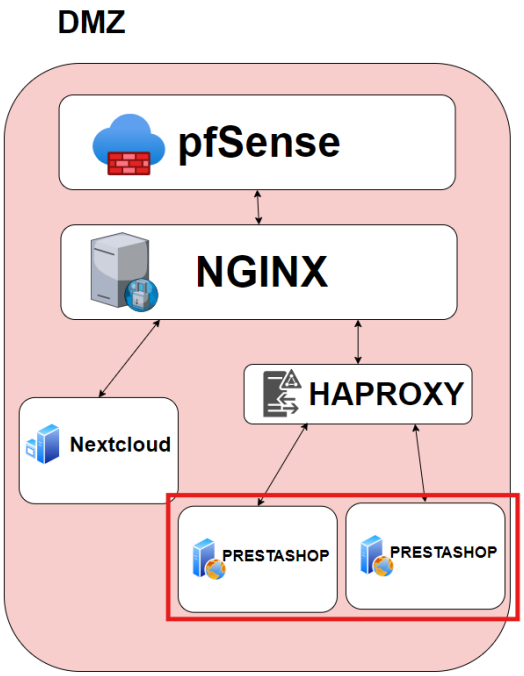
Serveurs PrestaShop sous Portainer :

Container list — admin

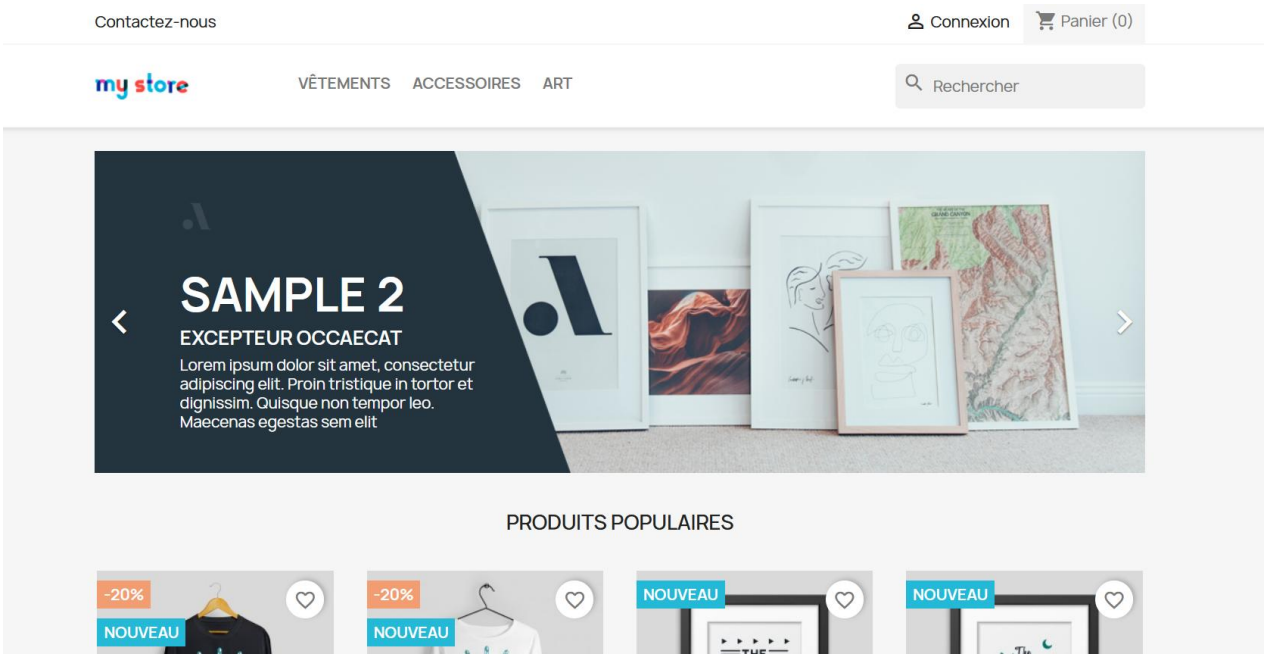
Containers — Search... — Start | Stop | Kill | Restart | Pause | Resume | Remove | + Add container

Name	State	Quick Actions	Stack	Image	Created	IP Address	Published Ports	Ownership
haproxy	running		haproxy	haproxy:latest	2025-03-19 12:56:23	172.21.0.2	8480:80	administra
nextcloud	running		nextcloud	nextcloud	2025-03-16 22:24:08	172.20.0.2	8280:80	administra
nginx_proxy	running		nginx_proxy	jc21/nginx-proxy-manager:latest	2025-04-04 14:06:13	172.18.0.2	443:443 80:80	administra
portainer	running		-	portainer/portainer-ce:lts	2025-03-12 22:27:05	172.17.0.2	8000:8000 9443:9443	administra
prestashop	running		prestashop	prestashop/prestashop:latest	2025-03-18 18:03:18	172.21.0.4	8380:80	administra
prestashop2	running		prestashop2	prestashop/prestashop:latest	2025-03-19 19:54:56	172.21.0.3	8580:80	administra
wordpress	running		wordpress	wordpress	2025-03-18 22:37:11	172.19.0.2	8180:80	administra

Mise en place réseau :



Site Web :

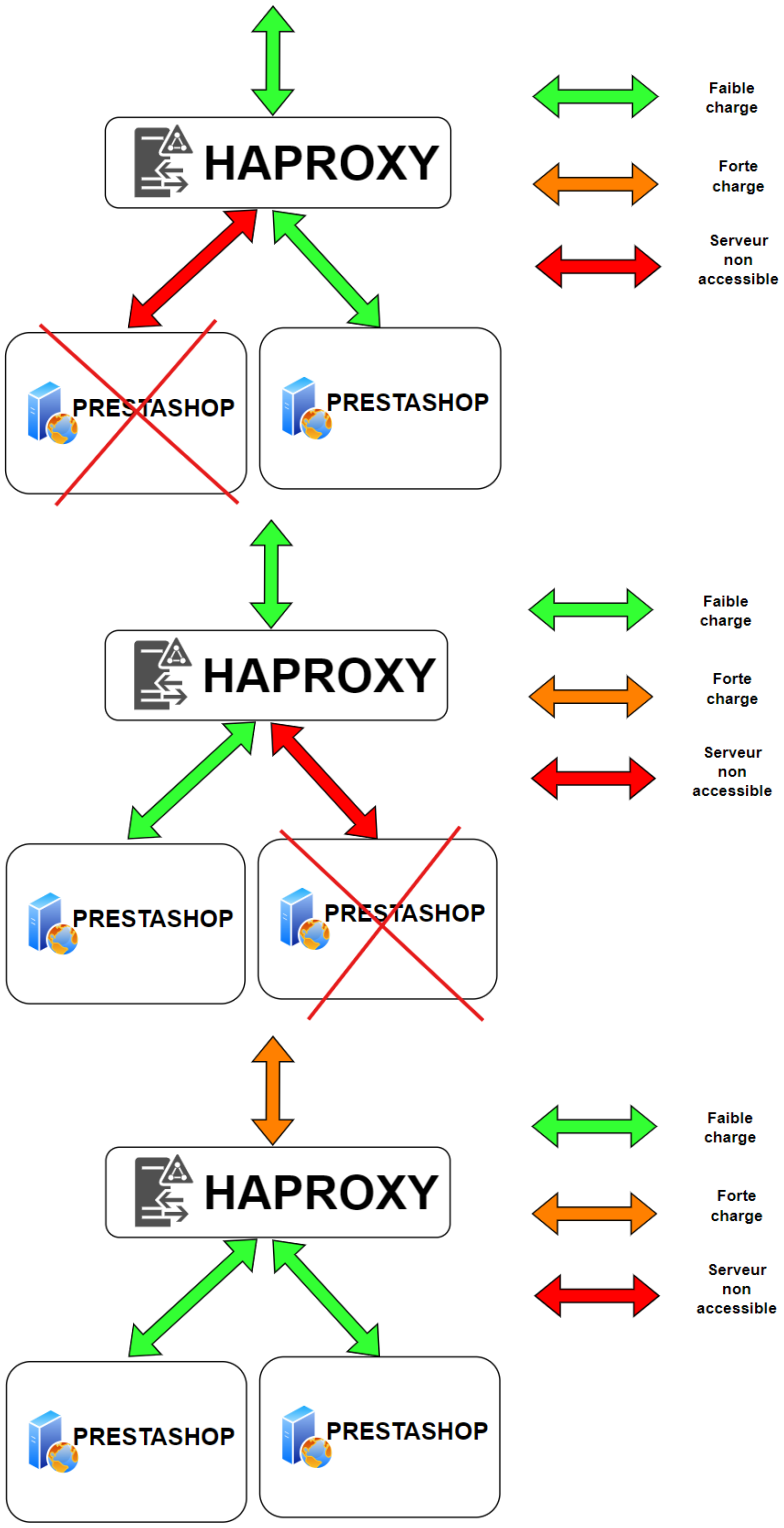


HAproxy

Mon HAproxy me permet d'assurer la répartition de charge. Concrètement, si l'un des serveurs tombe en panne, HAproxy redirige automatiquement les requêtes vers un autre serveur encore fonctionnel. Cela garantit une haute disponibilité du site web, même en cas de défaillance.

Mon HAproxy est mes serveurs PrestaShop sont dans le même réseau Docker à fin de faciliter les échanges entre eux.

Exemple réseau :



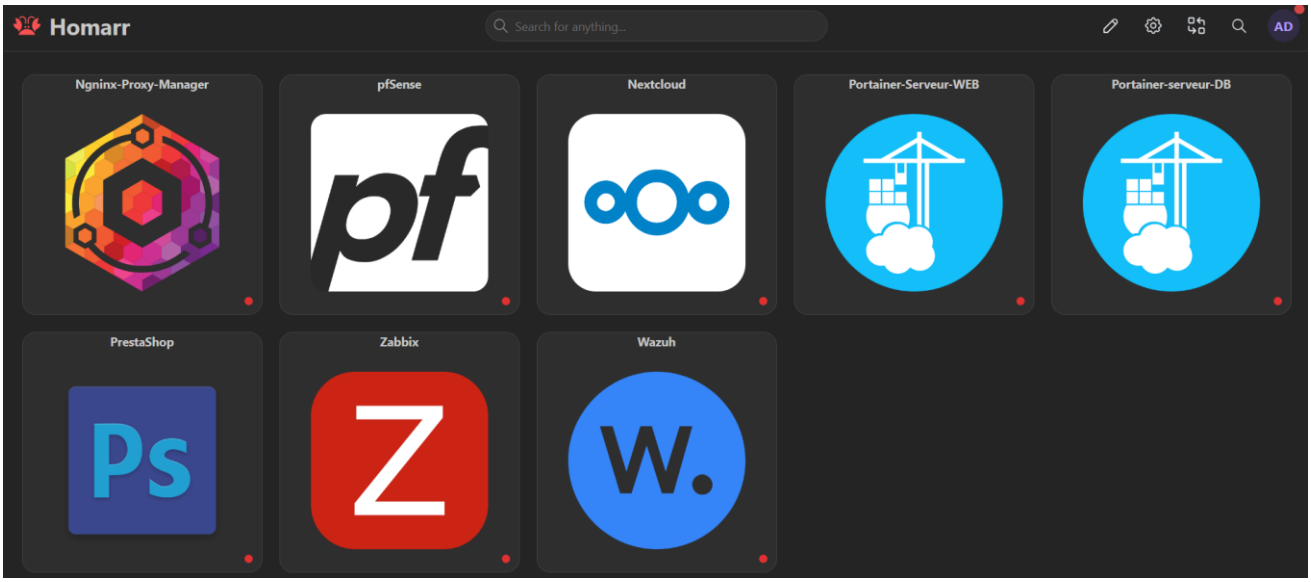
Grace à cette disposition, le site reste accessible. Attention à bien superviser le HAproxy, qui devient notre point de faiblesse (SPOF).

Homarr

Pour faciliter l'administration des serveurs et de leurs services, j'ai mis en place l'interface Homarr. Cet outil centralise l'accès à l'ensemble des applications et tableaux de bord via une seule page d'accueil personnalisable.

Les administrateurs peuvent ainsi se connecter à Homarr pour accéder rapidement aux différents services (comme PrestaShop, Nextcloud, ou les interfaces de supervision) et organiser leur interface selon leurs besoins. Cela améliore considérablement la visibilité, la réactivité et l'efficacité de la gestion quotidienne de l'infrastructure.

Interface Web :



SERVEUR DEBIAN BASE DE DONNÉES

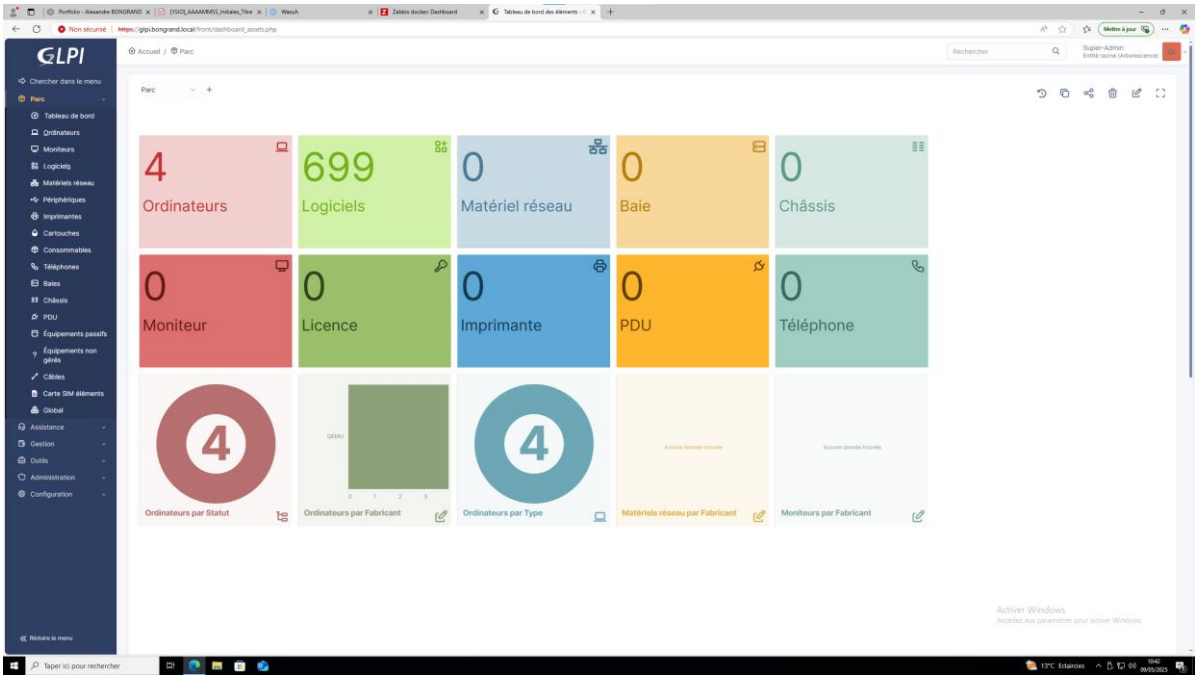
GLPI

Pour la gestion de l'inventaire et des tickets, l'entreprise souhaite avoir un logiciel dédié. Pour répondre à ce besoin, j'ai installé GLPI.

Comme ce logiciel est pour un usage interne, je l'ai installé sur le serveur " debiandatabases", il n'est pas accessible depuis internet.

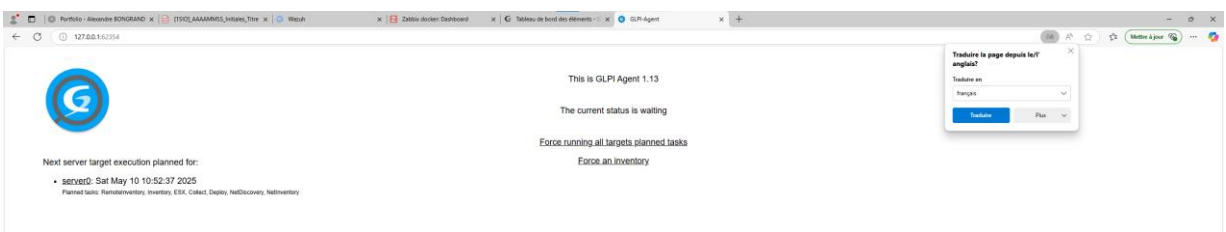
Je n'ai pas fait de liaison LDAP/AD pour s'identifier à GLPI, j'ai créé les utilisateurs à la main. Néanmoins, il est tout à fait possible de faire un script pour créer nos utilisateurs automatiquement, si nous avons beaucoup de compte à créer.

Interface Web :

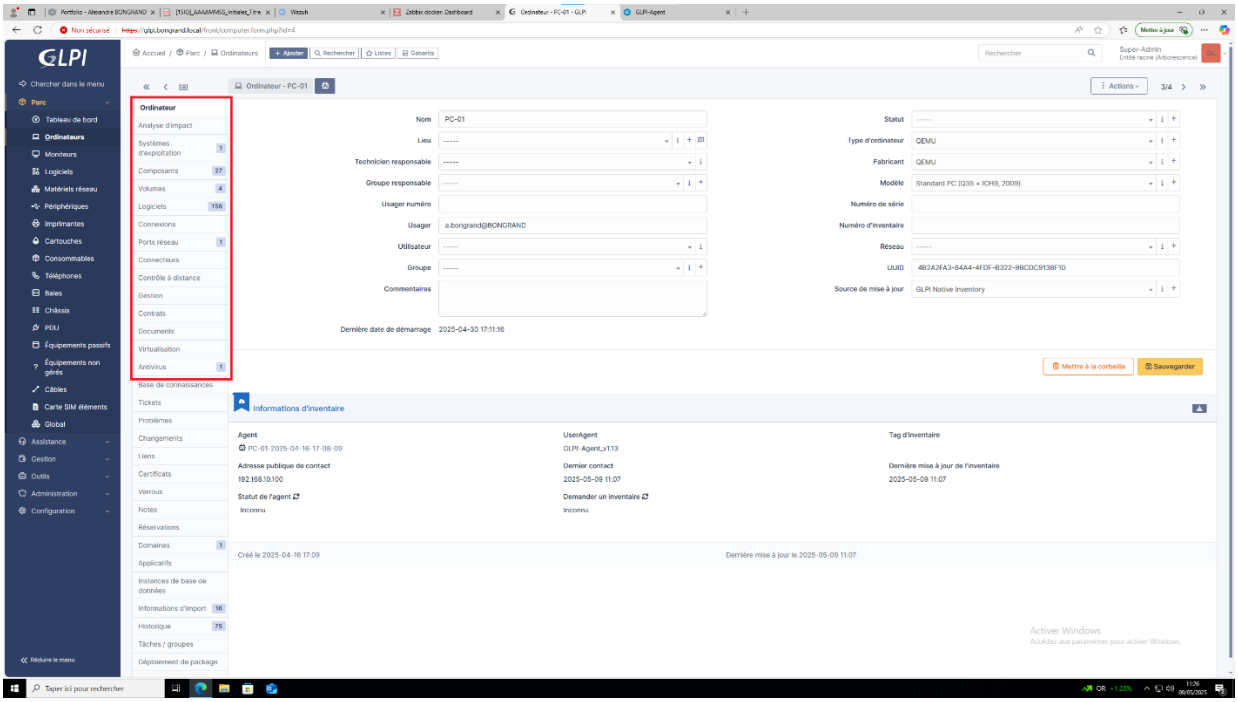


Pour l'inventaire de l'entreprise, j'ai installé l'agent GLPI sur les postes clients, ce qui permet de remonter, l'adresse IP, la version de l'OS, les logiciels, le matériel hardware et les écrans connectés

Agent GLPI :



Informations remontées :



Mise à jour Bases de données

Pour exploiter l'ensemble de ces nouveaux services installés, j'ai utilisé la base de données que j'avais créée auparavant. Le fait de centraliser les données dans une seule base facilite l'administration et permet de simplifier les sauvegardes.

Toutes les données sont stockées sur mon serveur MariaDB, installé précédemment.

```
+--------------------+
| Database           |
+--------------------+
| glpi               |
| information_schema |
| mysql              |
| nextcloud          |
| nginx_proxy        |
| performance_schema |
| presta             |
| sys                |
| wordpress          |
| zabbix             |
+--------------------+
```