



Les Maxence

Documentation
Equipe 2



Sommaire :

Projet Cub	1
Schéma - Infrastructure	2
Info. Réseaux & Composantes	3
Real. Tech. - Switch L3 - Cisco	4
Schéma - Switch L3 - Pare-feu	5
Real. Tech. - Stormshield	6
Real. Tech. - DHCP-LAN-1	10
Real. Tech. - DHCP-LAN-2	11
Real. Tech. - DNS-Récuratif	12
Real. Tech. - DNS-Autoritaire	13
Real. Tech. - Zabbix	14
Real. Tech. - GLPI	16
Real. Tech. - HA-Proxy	18
Real. Tech. - Serv-Web	19

Projet CUB

Le projet CUB (Configuration d'une Unité de Base) s'inscrit dans le cadre des épreuves professionnelles de deuxième année du BTS SIO – option SISR. Il s'agit d'un projet complet et évalué, visant à mettre en œuvre une **infrastructure réseau fonctionnelle** et sécurisée, tout en appliquant les **compétences techniques acquises** durant la **formation**.

L'objectif principal de ce projet est de concevoir, **déployer** et **administrer** un **système d'information interne**, en **simulant** un **environnement d'entreprise**. Les activités réalisées couvrent l'ensemble des compétences attendues : **configuration des équipements réseau, virtualisation, administration système, mise en service des services essentiels, supervision** et **résolution de pannes**.

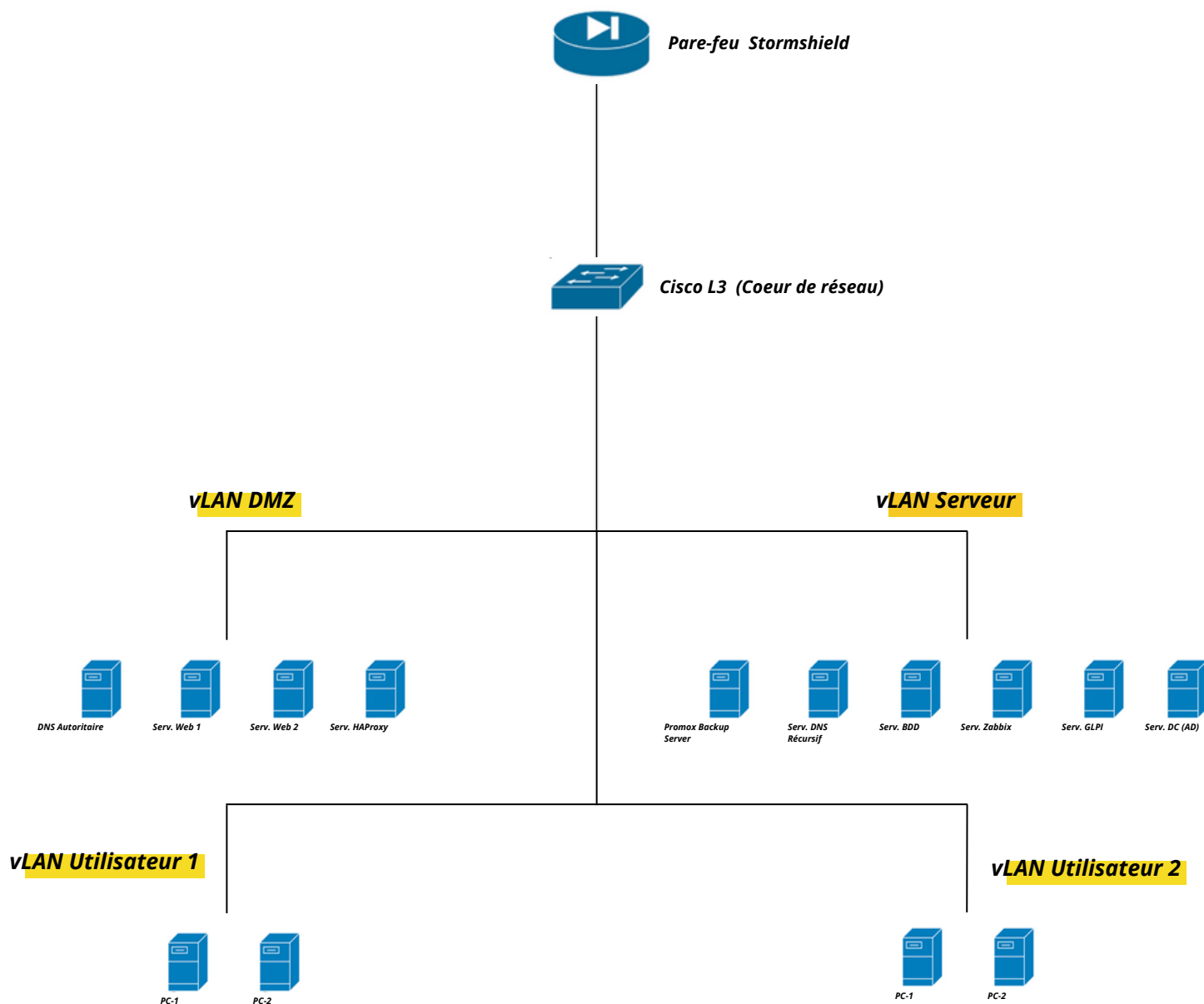
L'architecture technique repose sur les équipements et solutions suivants :

- Un switch **L3 Cisco**, configuré pour assurer le routage inter-VLAN et la segmentation du réseau.
- Un **pare-feu Stormshield**, assurant la sécurité des flux entrants et sortants, avec des règles de filtrage précises.
- Un serveur de **virtualisation Proxmox**, hébergeant l'ensemble des machines virtuelles nécessaires : contrôleur de domaine, serveurs web, DNS, base de données, supervision, etc.
- Ce projet permet de mettre en pratique des scénarios **réalistes d'administration réseau et système**, tout en intégrant des problématiques de **sécurité, de haute disponibilité** et de **dépannage**.

Schéma - Infrastructure

Le schéma ci-dessous représente l'architecture globale mise en place dans le cadre du projet CUB. Il illustre les différentes zones réseau (VLANs), le rôle du switch L3 Cisco en tant que cœur de réseau, ainsi que la position du pare-feu Stormshield en frontal pour contrôler les flux inter-VLAN et vers Internet.

Chaque machine virtuelle ou physique est placée dans le VLAN correspondant à son usage (DMZ, Serveur, Utilisateur), assurant ainsi une segmentation claire, une meilleure sécurité et une gestion optimisée du trafic.



Info. Réseaux & Composantes

Dans le cadre de la mise en place de notre infrastructure réseau, nous avons configuré l'ensemble des machines virtuelles, serveurs et équipements réseau nécessaires. Chaque composant a été identifié, associé à un mot de passe par défaut, un VLAN spécifique et une adresse IPv4 fixe, permettant une gestion centralisée, sécurisée et claire du système d'information.

La table ci-dessous récapitule tous les accès aux différents serveurs et équipements physiques/virtuels du réseau, incluant les identifiants d'administration, les VLANs associés et les plages IP utilisées :

Serveur	ID défaut :	MDP défaut :	ID secondaire :	MDP secondaire :	vLAN	IPv4/24
Windows-DC	Administrateur	Lesmaxence8716&	Aucun	Aucun	Serveur	172.16.22.220 (A vérifier)
SNMP-Centreon	glpi	Lesmaxence	Aucun	Aucun	DMZ	172.16.22.106
GLPI	root	Lesmaxence	Aucun	Lesmaxence/glpi/user	Serveur	172.16.22.104
Proxy-HA-1	root	Lesmaxence	maxence	Lesmaxence	DMZ	172.16.2.190
HAProxy	root	Lesmaxence	Aucun	Lesmaxence/glpi/user	Serveur	172.16.22.120
Web-1	root	Lesmaxence	maxence	Lesmaxence	DMZ	172.16.2.111
Web-2	root	Lesmaxence	maxence	Lesmaxence	DMZ	172.16.2.112
BDD-1-PC	root	Lesmaxence	maxence	Lesmaxence	Serv	172.16.22.103
Zabbix	root	ID : Admin: MDP : zabbix	Aucun	Aucun	Aucun	172.16.22.102
DNS-NOM	root	Lesmaxence	Aucun	Aucun	DMZ	172.16.2.100
DNS-Recursif	root	lesmaxence	Aucun	Aucun	Serveur	172.16.22.99
DHCP-LAN-1-2-Serv	root	maxence	Aucun	Aucun	Serveur	172.16.22.9

Routeur :	ID Défaut :	MDP	IPv4/24
Stormshield Physique	admin	Lesmaxence	192.168.229.9
Stormshield Virtuelle HA	admin	Lesmaxence8716&@#	192.168.229.9

Switch	ID Défaut :	MDP	IPv4/24
Switch L3	admin	Lesmaxence	172.16.22.253 (sur VLAN 311)

Real. Tech. - Switch L3 - Cisco

Dans le cadre de la mise en place de l'infrastructure réseau du projet CUB, plusieurs équipements clés ont été configurés afin d'assurer une communication fluide, sécurisée et segmentée entre les différentes machines virtuelles et physiques.

Pare-feu Stormshield :

Le pare-feu Stormshield a été configuré de manière à assurer une séparation stricte des flux réseau.

Chaque interface physique du pare-feu a été associée à un VLAN spécifique, correspondant à une zone de l'infrastructure (LAN Serveur, DMZ, LAN Utilisateur, etc.).

Cela permet d'appliquer des politiques de sécurité précises selon l'origine et la destination des flux.

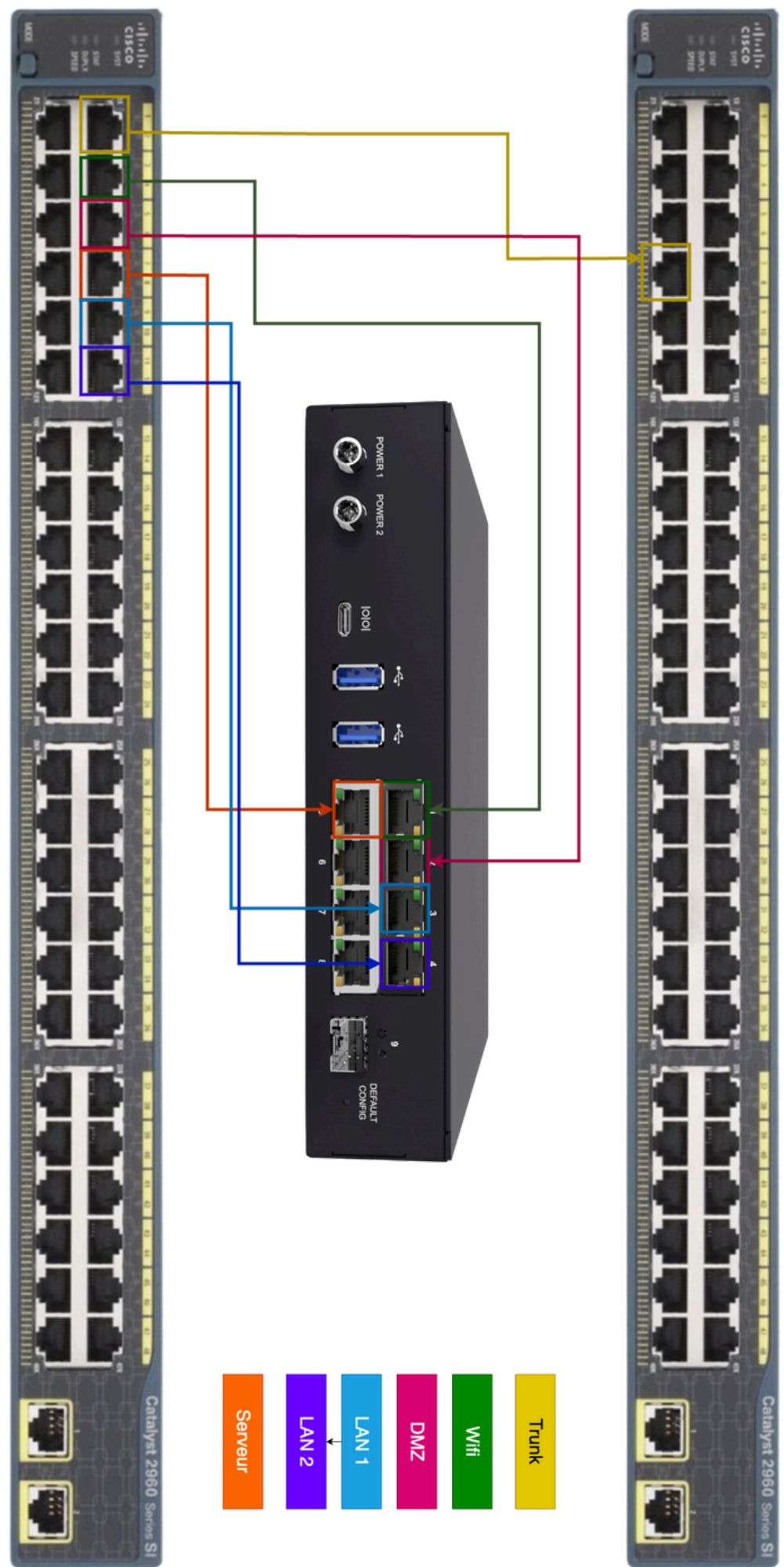
Switch Cisco L3

Le switch Cisco de niveau 3 joue un rôle central dans le routage inter-VLAN et la distribution du trafic. Voici les principales configurations réalisées :

- Mise en place d'un lien trunk sur le port 1, allant jusqu'à l'Aruba central.
- Utilisation de la commande `switchport trunk encapsulation dot1q` pour gérer le marquage VLAN.
- Attribution des VLANs nécessaires au bon fonctionnement du réseau : 134, 302, 310, 311, 312, 313.
- Association des interfaces 37 à 42 au VLAN 134, utilisé comme VLAN Internet.
- Désactivation du VLAN 1 (vlan par défaut) pour des raisons de sécurité : soit par la commande `shutdown`, soit en retirant son adresse IP.

Schéma page suivante.

Schéma - Switch L3 - Pare-feu



Real. Tech. - Stormshield















Nous avons en premier commencé par créer des objets réseaux afin de les assigner aux interfaces du routeur afin d'en faire les passerelles. Voic les objets :










- Objet VLAN DMZ : 172.16.2.0/24
- Objet VLAN Serveur : 172.16.22.0/24
- **Objet VLAN Utilisateur 1 : 192.168.2.0/24**
- **Objet VLAN Utilisateur 2 : 192.168.22.0/24**

Ou encore renseigner un objet pour l'interface du routeur côté WAN :

- **Objet Interface WAN : 192.168.1.229.9/26**

Voici des captures d'écran pour illustrer :

		Firewall_WAN	192.168.229.9 / static
		Firewall_CUB_DMZ1	172.16.2.254 / static
		Firewall_LAN1	192.168.2.254 / static
		Firewall_LAN2	192.168.22.254 / static
		Firewall_LAN-Serv	172.16.22.254 / static
		Firewall_HA	10.10.0.5 / static
		Firewall_bridge	10.0.0.254 / static

		GATEWAY-CUB	192.168.229.1 / static
		DHCP	172.16.22.9 / static
		ewc-sns.stormshieldcs....	34.120.68.241 / dynamic
		DNS-Privé	172.16.2.100 / static
		LAN-1	192.168.2.0 / static
		LAN-2	192.168.22.0 / static
		DNS-Recursif	172.16.22.99 / static
		Radius	172.16.2.101 / static
		Serv-Web	172.16.2.102 / static

Real. Tech. - Stormshield

L'ensemble du trafic entrant et sortant sur notre infrastructure est strictement contrôlé par un pare-feu configuré avec des règles de filtrage précises. Ces règles assurent la sécurité, la segmentation réseau ainsi que l'accès contrôlé aux services essentiels (DNS, Web, base de données, etc.).

Les règles sont classées par catégories (DNS, Web, Admin, WAN, etc.) et appliquées selon l'interface, l'adresse source/destination, le protocole et le port. Une règle de blocage final vient systématiquement fermer toute communication non explicitement autorisée.

La capture suivante présente l'ensemble des règles actuellement appliquées :

FILTERING NAT									
Searching...									
	Status	Action	Source	Destination	Dest. port	Protocol	Security inspection	Comments	
Section 1 - Règles d'autorisation à destination du pare-feu (contains 3 rules, from 1 to 3)									
1			Network_WAN	Firewall_WAN	https			Log in Firewall By HTTPS	
2			Internet interface: WAN	Firewall_WAN	dns				
3			Internet interface: WAN	Firewall_WAN DNS-Privé	dns			Created on 2025-04-16 12:53:44 by admin (192.168.229.99)	
Règles d'autorisation des flux DNS et internet (contains 5 rules, from 4 to 8)									
4			Network_CUB_DMZ1	Any	dns			DMZ à joindre all via DNS	
5			DNS-Recursif	Internet	dns			Created on 2024-11-08 11:45:49 by admin (192.168.229.60)	
6			DNS-Recursif	DNS-Privé	dns			Created on 2025-01-17 17:21:09 by admin (192.168.229.122)	
7			Network_LAN1 Network_LAN2 Network_LAN-Serv DNS-Privé	DNS-Recursif	dns			Created on 2024-11-08 11:17:18 by admin (192.168.229.60)	
8			Network_LAN1 Network_LAN2 Network_LAN-Serv Network_CUB_DMZ1	Internet	http https				
Règles Admin Poste & Ping (contains 2 rules, from 9 to 10)									
9			Win-Admin	Network_CUB_DMZ1	Any			Created on 2025-03-12 16:35:00 by admin (192.168.229.176)	
10			Network_LAN-Serv	Any	icmp			Created on 2024-11-08 13:57:35 by admin (192.168.229.60)	
Règles Web Server (contains 2 rules, from 11 to 12)									
11			Serv-Web Serv-Web2	Serv-800	mysql			Created on 2025-03-19 12:19:01 by admin (192.168.229.176)	
12			Network_LAN1 Network_LAN2 Network_LAN-Serv Network_CUB_DMZ1	GLPI	Any			Created on 2025-04-15 12:21:31 by admin (192.168.229.124)	
Web depuis Cub WAN (contains 4 rules, from 13 to 16)									
13			Internet interface: WAN	Firewall_WAN HA-Proxy-Web	Web2-8081 https			Created on 2025-04-18 14:20:08 by admin (192.168.229.99)	
14			Internet interface: WAN	Firewall_WAN Serv-Web2	https			Created on 2025-04-18 14:14:50 by admin (192.168.229.99)	
15			Internet interface: WAN	Firewall_WAN Serv-Web	web_8080 https			Created on 2025-04-18 13:35:52 by admin (192.168.229.99)	
16			Any interface: WAN	Firewall_WAN Serv-Web	web-Certificat ssh			Created on 2025-01-08 10:50:22 by admin (192.168.229.122)	
Backup DNS - PRS (contains 1 rules, from 17 to 17)									
17			DNS-Privé	Network_LAN-Serv	BACKUP			Created on 2025-03-12 11:58:42 by admin (192.168.229.131)	
Règle d'interdiction Finale (contains 1 rules, from 18 to 18)									
18			Any	Any	Any			Règle blocage par défaut / Log	

Explication - Règles filtrage

Afin de mieux comprendre la logique de filtrage mise en place sur notre pare-feu, nous présentons ci-dessous un résumé clair et concis de chaque règle de sécurité configurée, en précisant son rôle et sa fonction dans l'infrastructure réseau.

Section 1 – Règles d'autorisation à destination du pare-feu

Règle 1 : Autorise l'accès HTTPS à l'interface de gestion du pare-feu depuis le WAN.

Règle 2 : Autorise les requêtes DNS depuis Internet vers le pare-feu.

Règle 3 : Autorise les requêtes DNS du pare-feu vers un serveur DNS privé.

Section 2 – Flux DNS et Internet

Règle 4 : Autorise les serveurs de la DMZ à joindre n'importe quel DNS.

Règle 5 : Permet à DNS-Recursif d'accéder à Internet.

Règle 6 : Autorise DNS-Recursif à interroger DNS-Privé.

Règle 7 : Permet aux LANs et aux serveurs de requêter DNS-Recursif.

Règle 8 : Autorise tous les réseaux internes à accéder à Internet via HTTP/HTTPS.

Section 3 – Admin & Ping

Règle 9 : Permet à la machine admin (Win-Admin) d'administrer la DMZ.

Règle 10 : Autorise les pings entre tous les réseaux internes depuis LAN Serveur (debug, monitoring).

Section 4 – Web Server

Règle 11 : Autorise les accès MySQL depuis les serveurs web vers le serveur BDD.

Règle 12 : Permet aux serveurs internes d'accéder à GLPI via HTTP/HTTPS.

Section 5 – Web depuis WAN

Règle 13 : Permet au WAN d'accéder à l'interface HAProxy (port 8082).

Règle 14 : Autorise l'accès à Web-2 via HTTPS depuis l'extérieur.

Règle 15 : Ouvre le port 8080 vers Web-1 depuis Internet.

Règle 16 : Permet les connexions HTTPS et SSH vers les serveurs web.

Explication - Règles filtrage

Section 6 – Backup DNS PBS

Règle 17 : Autorise le serveur DNS-Privé à se connecter au serveur de sauvegarde via le port BACKUP.

Section 7 – Règle de blocage finale

Règle 18 : Bloque tout le trafic non explicitement autorisé, avec log activé.

Real. Tech. - DHCP-LAN-1

Nous avons commencé par créer une machine virtuelle sous Debian et déployé dans le VLAN Serveur 311. Ensuite, nous avons installé le service DHCP : `isc-dhcp` puis édité le fichier de configuration suivant : `/etc/dhcp/dhcpd.conf` avec la configuration ci-dessous :

```
subnet 192.168.2.0 netmask 255.255.255.0
range 192.168.2.11 192.168.2.99;
option domain-name-servers 8.8.8.8;
option domain-name "bastia.cub.fr";
option routers 192.168.2.254;
option broadcast-address 192.168.2.254;
default-lease-time 600;
max-lease-time 7200;
```

Après cela, nous sauvegardons la configuration, puis nous réalisons un `ipconfig` puis `/release` et enfin un `/renew`.

Voici un aperçus du résultat :

Carte Ethernet Ethernet :

```
Suffixe DNS propre à la connexion. . . . : agence.cubbastia.fr
Adresse IPv6 de liaison locale. . . . . : fe80::ef25:d68c:104f:4383%24
Adresse IPv4. . . . . : 192.168.2.11
Masque de sous-réseau. . . . . : 255.255.255.0
Passerelle par défaut. . . . . : 192.168.2.254
```

Photo du résultat obtenus après le test**

Real. Tech. - DHCP-LAN-2

Nous avons réutiliser le même serveur DHCP que pour le LAN 192.168.2.0/24. Nous avons simplement édité le fichier : **/etc/dhcp/dhcpd.conf** et mis la configuration suivante :

```
subnet 192.168.22.0 netmask 255.255.255.0 {  
  range 192.168.22.11 192.168.22.99;  
  option domain-name-servers 8.8.8.8;  
  option domain-name "bastia.cub.fr";  
  option routers 192.168.22.254;  
  option broadcast-address 192.168.22.254;  
  default-lease-time 600;  
  max-lease-time 7200; }
```

Après cela, nous sauvegardons la configuration, puis nous réalisons un **ipconfig** puis **/release** et enfin un **/renew** sur le pc d'un utilisateur :

Voici un aperçu du résultat :

Carte Ethernet Ethernet :

```
Suffixe DNS propre à la connexion. . . . : agence.cubbastia.fr  
Adresse IPv6 de liaison locale. . . . . : fe80::ef25:d68c:104f:4383%24  
Adresse IPv4. . . . . : 192.168.22.11  
Masque de sous-réseau. . . . . : 255.255.255.0  
Passerelle par défaut. . . . . : 192.168.22.254
```

Photo du résultat obtenus après le test**

Enfin, les manœuvres ont été répétées sur le LAN Serveur et DMZ.

Real. Tech. - DNS-Récuratif

Nous avons réutilisé une machine virtuelle dédiée pour le réseau 172.16.22.0/24, à laquelle nous avons attribué l'adresse IP suivante : 172.16.22.99/24.

Puis installé le service Unbound sur cette VM et l'avons associée au VLAN 311.

Enfin, nous avons simplement édité le fichier : /etc/unbound/unbound.conf et appliqué la configuration suivante :

```
server:
  verbosity: 1
  interface: 0.0.0.0
  access-control: 0.0.0.0/0
  allow do-not-query-localhost: no

  forward-zone:
    name: "bastia.cubfr"
    forward-addr: 172.16.2.100

  forward-zone:
    name: "."
    forward-addr: 8.8.8.8
    forward-addr: 8.8.4.4
```

Lorsque la configuration est correctement appliquée, la machine devient un serveur DNS récursif fonctionnel, capable de résoudre les requêtes du domaine interne bastia.cubfr via le DNS local (172.16.2.100), tout en assurant la résolution des domaines externes grâce aux serveurs publics de Google (8.8.8.8 et 8.8.4.4).

Les clients du réseau peuvent ainsi naviguer normalement sur Internet et interroger les services internes définis.

Real. Tech. - DNS-Autoritaire

Nous avons réutilisé une machine virtuelle dédiée pour le réseau 172.16.22.0/24, à laquelle nous avons attribué l'adresse IP suivante : 172.16.22.99/24.

Puis installé le service Bind9 sur cette VM et l'avons associée au VLAN 312.

Enfin, nous avons simplement créé le fichier : /etc/bind/bastia.cub.fr et appliqué la configuration suivante :

```
$TTL 604800
@ IN SOA beginningit.fr. root.beginningit.fr. (
    3 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL
;
@ IN NS ns1.beginningit.fr.
ns1 IN A 192.168.1.140
zabbix IN A 192.168.1.150
appweb IN A 192.168.1.160
files IN A 192.168.1.170
```

Nous avons ensuite validé la configuration avec la commande suivante :
`named-checkzone beginningit.fr /etc/bind/db.beginningit.fr`

Enfin, nous avons redémarré le service Bind9 afin de prendre en compte les modifications :
`systemctl restart bind9`

Ensuite il faut créer la zone DNS. Pour cela on crée le fichier : /etc/bind/named.conf.local :

```
zone "beginningit.fr" {
    type master; file "/etc/bind/db.beginningit.fr";
};
```

Puis on redémarre le service.

Une fois la configuration validée et le service Bind9 redémarré, le serveur DNS devient pleinement opérationnel en tant que serveur autoritaire pour la zone beginningit.fr.

Les noms définis dans la zone (comme zabbix.beginningit.fr ou files.beginningit.fr) peuvent être résolus par les clients du réseau, permettant une résolution DNS locale fiable et maîtrisée.

Real. Tech. - Zabbix

Nous avons utilisé une machine virtuelle Debian 12 dédiée pour héberger le serveur Zabbix.

Nous avons également préparé une ou plusieurs machines virtuelles supplémentaires pour y installer les agents Zabbix.

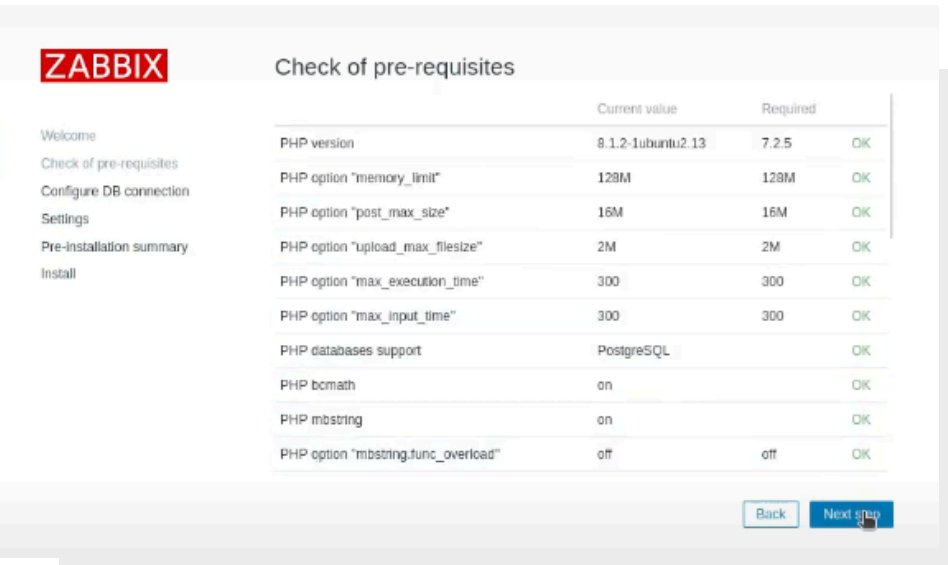
Un accès root ou via sudo est requis pour effectuer les différentes opérations.

Enfin, une stack LAMP (Apache, PHP, MariaDB) était déjà installée sur le serveur, sinon elle aurait dû être installée préalablement.

On y accède via l'ip : <ip>: 10003 et nous atterrissons directement sur la page d'installation de Zabbix :

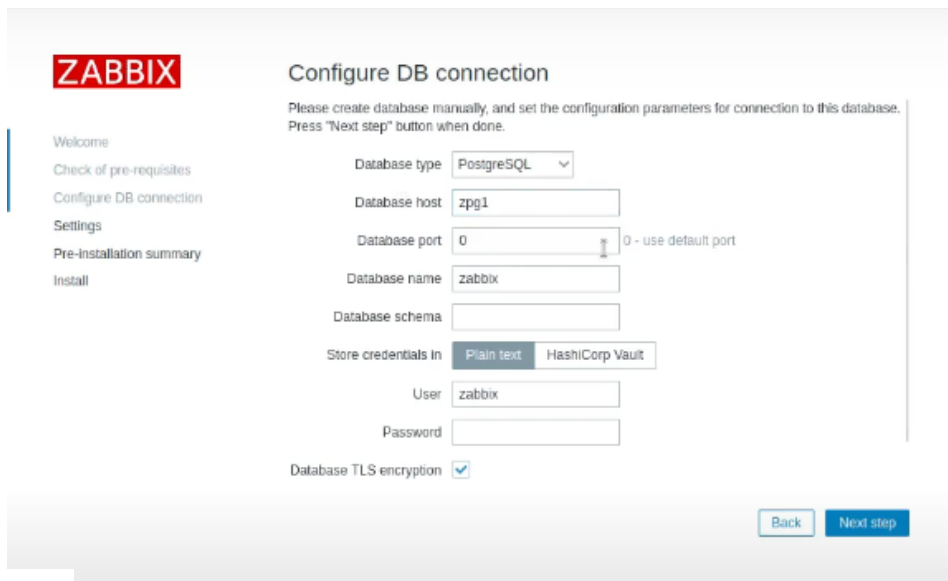


Cette fenêtre fait partie du processus d'installation de Zabbix Server via l'interface web. Plus précisément, c'est l'étape de vérification des prérequis PHP avant de continuer l'installation.



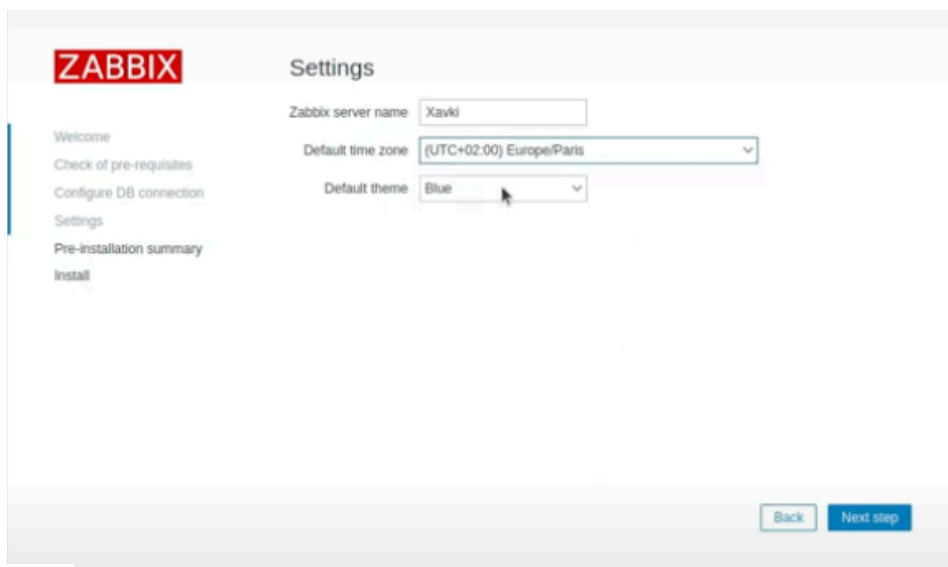
Real. Tech. - Zabbix

Ensuite, cette fenêtre correspond à la liason de la base de données mariadb avec l'interface Zabbix :



The screenshot shows the 'Configure DB connection' step in the Zabbix installation wizard. The left sidebar contains a navigation menu with the following items: Welcome, Check of pre-requisites, Configure DB connection (highlighted), Settings, Pre-Installation summary, and Install. The main content area is titled 'Configure DB connection' and includes the instruction: 'Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.' The configuration fields are as follows: Database type (PostgreSQL), Database host (zpg1), Database port (0, with a note '0 - use default port'), Database name (zabbix), Database schema (empty), Store credentials in (Plain text, HashiCorp Vault), User (zabbix), Password (empty), and Database TLS encryption (checked). At the bottom right, there are 'Back' and 'Next step' buttons.

Enfin, cette interface correspond au dernier réglages de Zabbix à configurer tel que l'heure ou la couleur de l'interface :



The screenshot shows the 'Settings' step in the Zabbix installation wizard. The left sidebar contains a navigation menu with the following items: Welcome, Check of pre-requisites, Configure DB connection, Settings (highlighted), Pre-Installation summary, and Install. The main content area is titled 'Settings' and includes the following configuration fields: Zabbix server name (Xavki), Default time zone ((UTC+02:00) Europe/Paris), and Default theme (Blue). At the bottom right, there are 'Back' and 'Next step' buttons.

Real. Tech. - GLPI

Nous avons utilisé une machine virtuelle Debian 12 dédiée pour héberger le serveur GLPI.

Nous avons également préparé une ou plusieurs machines virtuelles supplémentaires si besoin pour y connecter d'autres services ou extensions liés à GLPI.

Un accès root ou via sudo est requis pour effectuer les différentes opérations.

Enfin, une stack LAMP (Apache, PHP, MariaDB) était déjà installée sur le serveur, sinon elle aurait dû être installée préalablement.

On y accède via l'IP : 172.16.22.104 et nous atterrissons directement sur la page d'installation de GLPI :



Cette fenêtre fait partie du processus d'installation de GLPI via l'interface web. Ensuite, cette fenêtre correspond à la liaison de la base de données mariadb avec l'interface GLPI :



Real. Tech. - GLPI

Si la connexion est réussie, vous pourrez sélectionner la base de données glpidb pour l'installation :



GLPI SETUP

Étape 2

Test de connexion à la base de données

✓ Connexion à la base de données réussie

Veillez sélectionner une base de données :

Créer une nouvelle base ou utiliser une base existante :

☐

☒ db23_glpi

Continuer >

Valider les étapes jusqu'à la 6 et vous aurez l'accès à l'interface web :



GLPI SETUP

Étape 6

L'installation est terminée

Les identifiants et mots de passe par défaut sont :

- glpi/glpi pour le compte administrateur
- tech/tech pour le compte technicien
- normal/normal pour le compte normal
- post-only/postonly pour le compte postonly

Vous pouvez supprimer ou modifier ces comptes ainsi que les données initiales.

Utiliser GLPI

Real. Tech. - HA-Proxy

Nous avons mis en place un système de répartition de charge entre deux serveurs web à l'aide de HAProxy, configuré sur une machine Debian 12.

L'objectif est d'assurer une haute disponibilité des services web via une distribution équilibrée des requêtes entrantes. Le fichier de configuration principal utilisé est situé dans `/etc/haproxy/haproxy.cfg`.

Voici un exemple ci-dessous :

```
GNU nano 7.2 /etc/haproxy/haproxy.cfg
bind *:80
default_backend backend_webservers

backend backend_webservers
    balance roundrobin
#     server Web-A 172.16.2.121:80 check
#     server Web-B 172.16.2.122:80 check
    server Web-2 172.16.2.112:80 check
    server Web-1 172.16.2.111:80 check

listen stats
    bind *:8080
    stats enable
    stats uri /stats
    stats refresh 10s
    stats auth haproxy:admin
```

Dans cet exemple, nous utilisons une méthode de répartition de type roundrobin, et nous avons défini quatre serveurs web, dont deux sont commentés. L'écoute se fait sur le port 80 :

Nous avons également activé l'interface de statistiques d'HAProxy sur le port 8080, avec un rafraîchissement toutes les 10 secondes. Elle est protégée par une authentification basique (haproxy:admin) et accessible via `/stats`.

Real. Tech. - Apache2 - HTTP - HTTPS

Les serveurs web configurés en backend dans HAProxy sont des serveurs Apache hébergés sur des machines Debian 12. Chaque serveur possède une configuration similaire avec **HTTPS activé via certificat auto-signé avec OpenSSL**.

Voici un exemple de fichier de configuration site-ssl.conf pour Apache, sur le serveur ayant pour IP 172.16.2.112 :

Dans ce fichier, nous avons activé le module SSL (SSLEngine on) et indiqué les chemins vers les fichiers de certificat et de clé privée. Le répertoire racine du site est /var/www/htdocs avec toutes les permissions accordées.

Contenu du fichier virtualhost SSL :

```
GNU nano 7.2                                site-ssl.conf
<VirtualHost *:443>
    ServerName 172.16.2.112
    DocumentRoot /var/www/htdocs

    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/selfsigned.crt
    SSLCertificateKeyFile /etc/apache2/ssl/selfsigned.key

    <Directory /var/www/htdocs>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Nous avons également configuré une redirection automatique des requêtes HTTP vers HTTPS, pour garantir que toutes les connexions se fassent de manière sécurisée. Cela est réalisé dans le fichier web.conf :

Ici, la directive Redirect permanent /https://172.16.2.112/ assure que toute requête arrivant sur le port 80 est redirigée vers le port 443 en HTTPS. Le fichier reste très simple, mais essentiel pour renforcer la sécurité des accès.

```
GNU nano 7.2                                web.conf
<VirtualHost *:80>
    ServerName 172.16.2.112

    # Redirection permanente vers HTTPS
    Redirect permanent / https://172.16.2.112/

    # (Facultatif) Pour éviter les erreurs avec les répertoires :
    DocumentRoot /var/www/htdocs
</VirtualHost>
```