SAÉ S4.Deploi.01 : Documentation Vortex Network

Nolan Dupont, Nils Rayot, Kylian Metayer, Jess Rousselard, Flavien Riondet

6 avril 2025

Table des matières

Liste des tableaux

Table des figures

1 Inventaire des ressources

1.1 Restriction des hyperviseur

L'infrastructure se compose d'un cluster de 5 hyperviseurs (appelés nœuds), chacun disposant de 20 Go d'espace de stockage local, pour un total de 100 Go répartis sur l'ensemble du cluster. Les machines virtuelles (VM) sont réparties de manière stratégique entre les différents nœuds, de façon à optimiser l'utilisation de la mémoire vive disponible et garantir une meilleure tolérance aux pannes ainsi qu'une répartition équilibrée de la charge.

Le SDN est mis en place pour assurer la connectivité réseau entre toutes les VM, même lorsqu'elles sont hébergées sur des nœuds différents, tant qu'elles appartiennent au même segment réseau. Pour cela, nous utilisons des zones , permettant l'isolation et la segmentation réseau tout en maintenant une communication efficace entre les VM au sein de la même zone.

L'ensemble du cluster fonctionne actuellement sous Proxmox VE 8.2.7, une version stable et supportée par notre environnement d'hébergement. Bien que des versions plus récentes de Proxmox soient disponibles, nous avons volontairement évité les mises à jour vers la dernière version, celle-ci n'étant pas encore pleinement compatible avec notre datacenter. Une telle mise à jour pourrait entraîner des instabilités, notamment au niveau du SDN, et potentiellement compromettre la disponibilité du cluster.

1.2 Listing des machines virtuelles

Alias	Nom	IP
dns	srv-dns	192.168.1.10
dhcp-d	dhcp-datacenter	192.168.1.20
web	web-datacenter	192.168.1.30
sgbd-d	sgbd-datacenter	192.168.1.40
file	file-datacenter	192.168.1.50
ldap	ldap-datacenter	192.168.1.90
superviseur	superviseur-vortex	192.168.1.100
dhcp-a	dhcp-admin	192.168.2.3
dhcp-p	dhcp-post	192.168.10.3
post1	post1-post	192.168.10.5
web-dmz	web-dmz	192.168.0.3
dns-dmz	dns-dmz	192.168.0.2
routeur-NAT-p	routeur-private	192.168.3.10
routeur-NAT-d	routeur-dmz	192.168.3.11

TABLE 1 – Tableau des machines et leurs adresses IP

2 Network

2.1 Configuration Proxmox

Proxmox permet de configurer des réseaux facilement avec son système [1], ce qui nous permet de créer des [3] pour assurer la communication entre les machines virtuelles sur différents hyperviseurs. Aussi nous avons créent un réseau Simple [2] ou nous avons connecter les routeurs pour que le réseau est accès à internet. Pour visualiser le réseau, voir Infrastructure ci-dessus.

Alias Zone	Type zone	Alias schéma	Vnet	Subnet
vXvortex	VXLAN	Réseau de serveur	vnet1	192.168.1.0/24
vXvortex	VXLAN	Réseau des administrateurs	vnet2	192.168.2.0/24
vXvortex	VXLAN	Réseau de poste	vnet10	192.168.10.0/24
vXvortex	VXLAN	DMZ	vnet20	192.168.0.0/24
private	Simple	Réseau Internet	vnet0	192.168.3.0/24

TABLE 2 – Tableau configuration du réseau

2.2 Configuration hyperviseur

Pour que les machines virtuelles puissent communiquer avec Internet sans installer de serveur DNS dans le réseau, utilisez l'hyperviseur comme serveur DNS avec **dnsmasq**.

```
apt install dnsmasq
systemctl disable --now dnsmasq
nano /etc/network/interfaces
#ecrire tout en bas
        source /etc/network/interfaces.d/sdn
```

2.3 Configuration routeur NAT

Les [3] n'ayant aucune configuration par défaut, contrairement à un réseau Simple [2] qui offre une passerelle, un [5] et toute la base nécessaire pour utiliser un réseau, nous sommes alors obligés de créer notre propre routeur [6]. Ce routeur NAT utilise le framework netfilter implémentant un pare-feu avec Nf_table, un sous-système du noyaux Linux fournissant le filtrage des paquets. Les routeur sont sur la version `v1.0.6` de netfilter.

Nous avons listé les ports autorisés suivant les réseaux :

Pour que le forwarding soit actif il est nécéssaire de modifié une variable dans la configuration du routeur

Pour que le forwarding soit actif il est nécessaire de modifié une variable dans la configuration du routeur

Routeur	Code Réseau	Nom du Réseau	Plage IP	Interface routeur
private	1	Réseau de serveur	192.168.1.0/24	ens19
private	2	Réseau des administrateurs	192.168.2.0/24	ens20
private	3	Réseau de poste	192.168.10.0/4	ens21
private	4	DMZ	192.168.0.0/24	ens19
private/DMZ	5	Réseau Internet	192.168.3.0/24	ens18
private/DMZ	6	Internet	Adresse client	ens18

TABLE 3 – Légende des référence de nom des réseaux

```
# autoriser le forward sur la machine
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Dans tous les tableaux présentés ci-dessous, les règles DROP sont omises, car la politique de filtrage par défaut de nos routeurs NAT est déjà définie sur DROP. Cela signifie que toutes les connexions non explicitement autorisées par une règle (connexions ne correspondant pas à un critère de filtrage) seront rejetées de manière implicite. Ainsi, les règles de type DROP n'ont pas besoin d'être explicitement incluses dans les tableaux car le comportement par défaut du routeur, sans règles spécifiques, rejettera toute connexion non autorisée. Cela simplifie la configuration des tables, car nous n'avons pas besoin de mentionner à chaque fois les connexions qui sont rejetées par défaut. Cela permet de se concentrer uniquement sur les connexions explicitement acceptées dans les règles input, output, forward, et nat, sans répéter la logique de rejet pour chaque situation.

Port/Protocol	Input/Output/Forward	État	Source	Destination
*	Input	Established	*	*
*	Output	Established	*	*
ICMP	Input & Output rate limit 5/second	*	*	*
22/TCP	Input	new	*	5
22/TCP	Output	new	5	1/2/3
53/TCP & 53/UDP	Input	*	*	5
53/TCP & 53/UDP	Output	*	5	*
123/UDP	Input	new	6	*
123/UDP	Output	new	5	6
10050/TCP & 10051/TCP	Input	new	1/2/3	5
10050/TCP & 10051/TCP	Output	new	5	1/2/3
80/TCP & 443/TCP	Input	new	*	5
80/TCP & 443/TCP	Output	new	5	*

TABLE 4 – Table firewall du routeur NAT réseau privé Input & Output, accept

Port/Protocol	Input/Output/Forward	État	Source	Destination
*	*	Established	*	*
ICMP	rate limit 5/second	*	*	*
22/TCP	Forward	new	*	1/ 2/ 3/ 4/ 5
10050 & 10051/TCP	Forward	new	1	2
10050 & 10051/TCP	Forward	new	2	1
443/TCP	Forward	new	2	1
389/TCP	Forward	new	3	1
389/TCP	Forward	new	1	3
123/UDP	Forward	new	1	*
63/TCP	Forward	*	1	2
63/TCP	Forward	*	2	1
53/TCP & 53/UDP	Forward	*	1	*
53/TCP & 53/UDP	Forward	*	*	1
67/UDP & 68/UDP	Forward	*	1	2/ 3
67/UDP & 68/UDP	Forward	*	2/ 3	1
80/TCP & 443/TCP	Forward	new	1/ 2/ 3	*

TABLE 5 – Table firewall du routeur NAT réseau privé forwarding, accept

Interface Entrante	Interface Sortante	Action
6	1	Masquerade
6	2	Masquerade
6	3	Masquerade
1	6	Masquerade
1	2	Masquerade
1	3	Masquerade
2	6	Masquerade
2	1	Masquerade
3	6	Masquerade
3	1	Masquerade

TABLE 6 – Table masquerade du routeur NAT

Port/Protocol	Input/Output/Forward	État	Source	Destination
*	Input	established, related	*	*
22/TCP	Input	new	5/2	*
ICMP echo-request	Input rate limit 5/second	new	*	*
ICMP echo-reply	Input rate limit 5/second	new	*	*
*	Output	*	*	*
*	Forward	established, related	*	*
443/TCP	Forward	new	5	*
22/TCP	Forward	new	5/2	*
53/UDP	Forward	new	5	*
ICMP echo-request	Forward rate limit 5/second	new	*	*
ICMP echo-reply	Forward rate limit 5/second	new	*	*

TABLE 7 – Table firewall basée sur les règles nftables du routeur DMZ (inet filter)

Port/Protocol	Hook	Interface Entrante	Destination (DNAT)
443/TCP	prerouting	6	4 (192.168.0.4)
80/TCP	prerouting	6	4 (192.168.0.4)

TABLE 8 – Table NAT - redirection du routeur DMZ (DNAT)

Interface Entrante	Interface Sortante	Action
1	6	Masquerade
6	1	Masquerade

TABLE 9 – Table masquerade du pare-feu du routeur DMZ (nftables postrouting)

10050 & 10051/TCP	Forward	new	2	1
443/TCP	Forward	new	2	1
389/TCP	Forward	new	3	1
389/TCP	Forward	new	1	3
123/UDP	Forward	new	1	*
63/TCP	Forward	*	1	2
63/TCP	Forward	*	2	1
53/TCP & 53/UDP	Forward	*	1	*
53/TCP & 53/UDP	Forward	*	*	1
67/UDP & 68/UDP	Forward	*	1	2/ 3
67/UDP & 68/UDP	Forward	*	2/ 3	1
80/TCP & 443/TCP	Forward	new	1/ 2/ 3	*

TABLE 10 – Table firewall du routeur NAT forwarding, accept

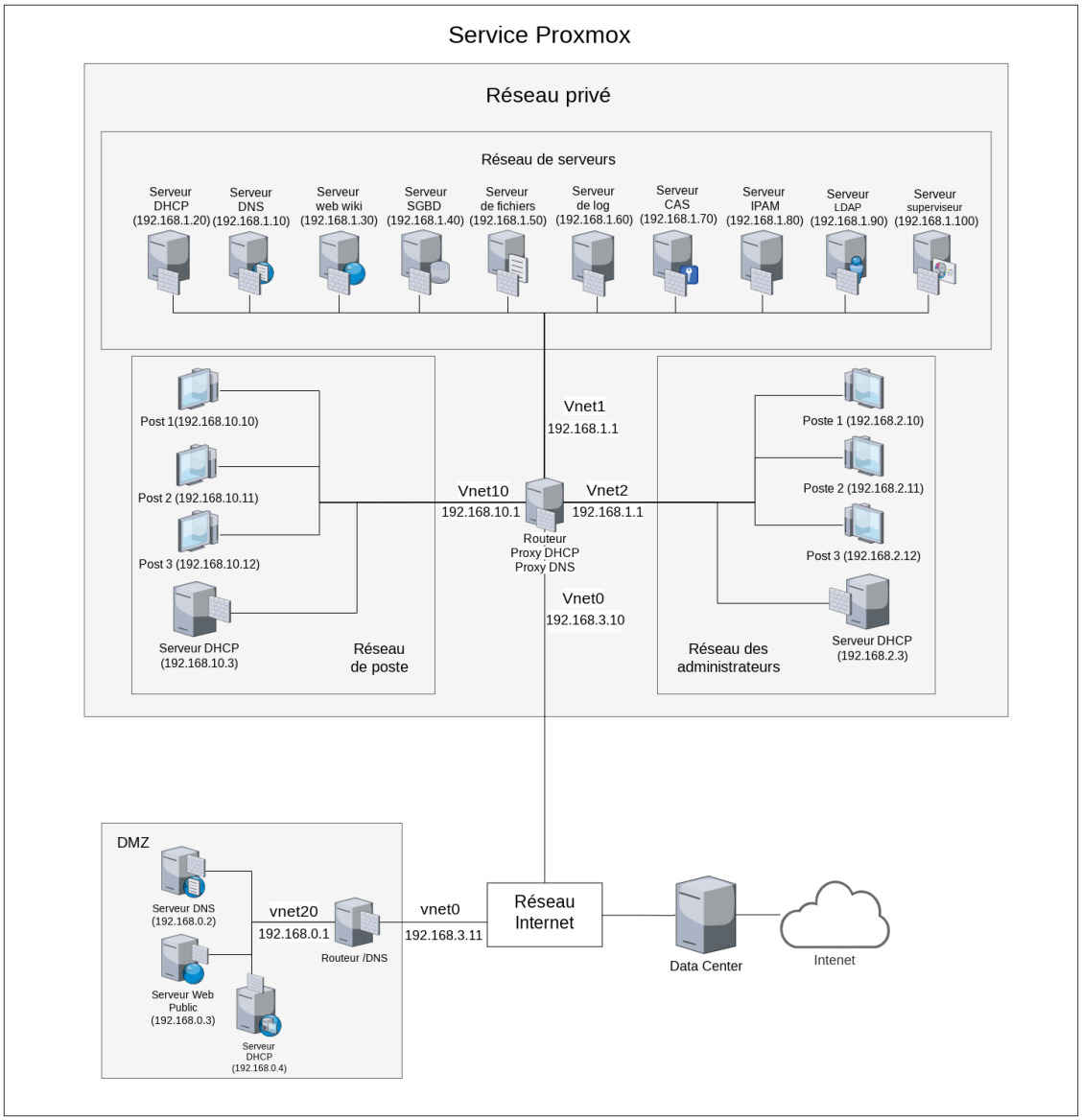
Port/Protocol	Input/Output/Forward	État	Source	Destination
*	Input	established	*	*
*	Output	established	*	*
ICMP	Input & Output rate limit 5/second	*	*	*
22/TCP	Input	new	*	5
22/TCP	Output	new	5	1/2/3
53/TCP & 53/UDP	Input	*	*	5
53/TCP & 53/UDP	Output	*	5	*
123/UDP	Input	new	6	*
123/UDP	Output	new	5	6
10050/TCP & 10051/TCP	Input	new	1/2/3	5
10050/TCP & 10051/TCP	Output	new	5	1/2/3
80/TCP & 443/TCP	Input	new	*	5
80/TCP & 443/TCP	Output	new	5	*

TABLE 11 – Table firewall du routeur NAT Input & Output, accept

Interface Entrante	Interface Sortante	Action
6	1	Masquerade
6	2	Masquerade
6	3	Masquerade
1	6	Masquerade
1	2	Masquerade
1	3	Masquerade
2	6	Masquerade
2	1	Masquerade
3	6	Masquerade
3	1	Masquerade

TABLE 12 – Table mascarade du routeur NAT

2.4 Infrastructure



2.5 Gestion accès

Pour établir une connexion SSH vers les machines du réseau, il est nécessaire de passer par un *jump* SSH via l'hyperviseur `Proxmox-dupontno-283`. Cet hyperviseur joue un rôle central, car il est le seul hôte accessible dans la zone simple où se trouvent les routeurs. Cette configuration permet de sécuriser l'accès aux machines internes en limitant les connexions entrantes directes.

Une fois connecté à l'hyperviseur, l'authentification aux machines du réseau s'effectue exclusivement par clé SSH. L'accès peut être établi en utilisant soit l'adresse IP du poste cible, soit son nom de domaine, selon la configuration réseau et le service de résolution de noms en place.

Pour garantir un déploiement homogène et sécurisé des clés SSH, un script d'automatisation est utilisé. Ce script se charge de générer, distribuer et installer les clés SSH sur l'ensemble des machines du réseau privé, garantissant ainsi un accès sécurisé et centralisé.

Mesures de sécurité renforcées : Dans un souci de renforcement de la sécurité, nous avons désactivé l'authentification SSH par mot de passe. Cette mesure empêche toute tentative de connexion via des identifiants classiques, réduisant ainsi les risques d'attaques par force brute et d'usurpation d'identité. Désormais, seules les connexions utilisant une clé SSH préalablement installée sont autorisées.

Grâce à cette infrastructure, nous assurons un accès sécurisé, automatisé et contrôlé aux machines du réseau, tout en minimisant les risques liés aux accès non autorisés.

2.6 DMZ

Cette zone est constituée d'un serveur web qui va afficher une page pour tous les utilisateurs qui vont se connecter de l'extérieur. La DMZ contient aussi sont routeur et son serveur DNS pour pouvoir résoudre le nom des serveurs à l'avenir comme un SGBD pour un site web plus complexe.

3 Système par défaut

3.1 Noyaux

3.2 Distribution

Nous avons utilisé Debian en ligne de commande pour tous les serveurs, et Debian avec le gestionnaire de fenêtres LXQt pour les machines des utilisateurs et des administrateurs. Nous avons choisi ce gestionnaire précisément car il est conçu pour être léger et performant, tout en offrant un visuel agréable.

3.3 Logiciel par défaut

4 Configuration des serveurs

4.1 logiciel

4.2 Supervision du réseau

Pour surveiller l'ensemble de l'infrastructure, nous avons choisi Zabbix. C'est un logiciel qui fait tourner un serveur sur une machine avec un tableau de bord regroupant toutes les informations. Un agent se déploie sur toutes les machines à surveiller pour pouvoir envoyer des données précises facilement. Le client se configure avec le fichier `/etc/zabbix/zabbix_agent2.conf`. Nous avons créé un script pour ajouter l'IP du serveur et le nom que l'agent aura.

4.3 Tableau des logiciels

Logiciel	hostname	fonction
KEA	DHCP-datacenter/DHCP-admin/DHCP-poste	Serveur d'adressage IP
Wikijs	`web-datacenter`	Serveur de documentation
Zabbix	`superviseur`	Surveillance du réseau
Bind9	`dns-datacenter/dns-dmz`	Serveur de résolution de nom
OpenLDAP	`ldap-datacenter`	Serveur d'annuaires LDAP
Samba	`file-datacenter`	Server de fichiers utilisant le
`libpam-ldapd/libnss-ldapd`	post-1	Client LDAP pour PAM et N
smbclient	`post-1`	Client SMB

TABLE 13 – Tableau configuration des logiciels

4.4 Serveur DHCP [4]

Pour mette en place notre serveur DHCP [4], nous avons utilisé le logiciel KEA logiciel OPEN-source DHCP [4].

Lors de l'installation du paquet 2 répertoires sont importants :

```
1 cat /etc/kea/* # fichier de configuration
2 cat /var/lib/kea/* # fichier de sauvegarde des baux ↩
    DHCP
```

Nous avons décidé de configuré le réseau sur IPv4 et non pas IPv6 donc le fichier de configuration sera "/etc/kea/kea-dhcp4.conf", et les baux stocké ici "/var/lib/kea/kea-leases4.csv". La configuration possède 2 parties "Dhcp4" et "subnet4"

4.4.1 Dhcp4

Dans un premier temps faire une backup de ce fichier en renommant en [File].backup
Ensuite, nous allons ajouter une nouvelle configuration neuve conseillée par https://www.it-connect.fr/linux-installer-et-configurer-un-serveur-dhcp-kea-sur-debian/

La norme RFC 2131 - IPv4 ne spécifie aucune norme sur le temps de renouvellement de bail, la durée se détermine suivant la taille d'un réseau dans le cadre d'un réseau d'entreprise de petite taille, nous pouvons laisser la valeur par défaut de 8 jours avec une T1 (valid-lifetime) 50% de T3 et T2 87.5% de T3

Les informations des baux du dhcp sont stockés dans un fichier "type = mefile". Les options data écrit sur ce document ne sont pas celles sur le serveur DHCP 201.

Si une erreur de syntaxe est relevée dans le fichier de configuration dans journald et avec systemctl status nous pourrons voir les messages d'erreur. $DHCP4_INIT_FAILfailedtoinitializeKeaserver : configurationerrorusingfile'/etc/kea/kea-dhcp4.conf' : /etc/kea/kea-dhcp4.conf : 2.1-7 : syntaxerror$

Nous indiquons l'interface de lecture que le DHCP [4] dois utilisé pour répondre au requête dans le champ <Interface machine>.

```
1            "Dhcp4": {
2                // Interface reseau sur laquelle le service DHCP doit etre en ecoute
3                "interfaces-config": {
4                        "interfaces": [<Interface machine>]
5                },
6
7                // Duree des baux DHCP (bail de 8 jours, renouvelable a partir de 4 jours)
8                "valid-lifetime": 691200, // timer de demande de renouvellement de bail par le DHCP source
9                "renew-timer": 345600, // timer de demande de renouvellement de bail par n'importe quelle autre DHCP
10               "rebind-timer": 604800,
11
12               // Serveur DHCP principal sur ce reseau local
13               "authoritative": true, // refus de reponse sur une plage ip non prise en charge par le DHCP
14
```

```
                // Configuration de la base des baux ↩
                   DHCP
"lease-database": { // base de bail dhcp↩
                    stocke en memoire et non pas dans ↩
                   une BD
                        "type": "memfile",
                        "persist": true,
                        "name": "/var/lib/kea/kea-↩
                           leases4.csv",
                        "lfc-interval": 3600
                },
```

Dans cette partie nous indiquons toute la partie adressage IP des machines de l'infrastructure :

- Plage d'adresse et masque de sous-réseau
- L'intervalles d'adressage autorisé pour le DHCP
- Les option spéciale comme là les adresses du DNS ou routeur
- La réservation d'ip fixe donné à certaines machine suivant leur adresse MAC

```
            // Suite de la configuration a ajouter ici
        "subnet4": [
           {
                "subnet": "XXX.XXX.XXX.XXX/XX",
                "pools": [
                        {"pool": "XXX.XXX.XXX.XXX - XXX.↩
                           XXX.XXX.XXX"}
                ],
                "option-data": [ // c'est dat sont des ↩
                   nom de domaine assigne par adresse
                        {
                                "name": "domain-name-↩
                                   servers",
                                "data": "XXX.XXX.XXX.XXX↩
                                   "
                        },
                        {
                                "name": "routers",
                                "data": "XXX.XXX.XXX.XXX↩
                                   "

```

```
                        }
                ],
                "reservations": [
                        {
                                "hw-address" : "XX:XX:XX:XX:XX:XX",
                                "ip-address" : "XXX.XXX.XXX.XXX",
                                "hostname" : "name"
                        },
                        {
                            ...
                        }
                ]
            }
        ]
    }
}
```

4.5 Annuaire LDAP

Pour mettre en place notre serveur LDAP, nous avons utilisé le logiciel OpenLDAP.

Lors de l'installation du paquet `slapd` (démon OpenLDAP), nous avons entré les options suivantes :

1. Domain name : `vortex-network.fr`
2. Orgnization name : `Vortex Network`
3. Delete database when purging ? `No`

4.5.1 Schéma Samba pour LDAP

Nous avons utilisé notre serveur LDAP pour l'authentification sur notre serveur Samba. Pour rendre cela possible, il a fallu ajouter le schéma de Samba pour LDAP, qui ajoute des nouveaux types d'entrées et d'attributs à la base de données du LDAP.

Pour récupérer le schéma de Samba pour LDAP, on peut installer Samba sur notre serveur LDAP, mais par soucis de stockage, nous avons préféré copier le schéma depuis une machine sur laquelle Samba était déjà installé : notre serveur de fichiers.

```
scp etu@file-datacenter:/usr/share/doc/samba/examples/LDAP/samba.schema .
```

Après avoir récupérer le schéma, nous l'avons converti au format LDIF, car le schéma actuel est écrit dans un format qui n'est plus supporté par OpenLDAP. Une fois le schéma converti dans le bon format, il suffit de l'ajouter à la base de données LDAP.

```
apt install schema2ldif
schema2ldif samba.schema > samba-schema.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -f samba-schema.ldif
```

4.5.2 Unités de l'organisation

Nous avons ajouté à notre LDAP deux unités d'organisation :
— l'unité des utilisateurs,
— l'unité des groupes.

Fichier LDIF 1: `base.ldif`

```
dn: ou=users,dc=vortex-network,dc=fr
objectClass: organizationalUnit
ou: users

dn: ou=groups,dc=vortex-network,dc=fr
objectClass: organizationalUnit
ou: groups
```

Il nous a suffit après d'ajouter le fichier LDIF à la base de données LDAP comme suit :

```
ldapadd -x -D "cn=admin,dc=vortex-network,dc=fr" -W -f ↩
    base.ldif
```

4.5.3 Entrées des groupes

Nous avons ensuite créé les groupes d'utilisateurs. Ces groupes servent de groupes Unix ainsi que groupes de permissions pour le Wiki.

Fichier LDIF 2: `groups.ldif`

```
dn: cn=Guests,ou=groups,dc=vortex-network,dc=fr
objectClass: top
objectClass: posixGroup
cn: Guests
gidNumber: 10000

dn: cn=Administrators,ou=groups,dc=vortex-network,dc=fr
objectClass: top
objectClass: posixGroup
cn: Administrators
gidNumber: 10001
```

```
1 ldapadd -x -D "cn=admin,dc=vortex-network,dc=fr" -W -f ↩
    groups.ldif
```

4.5.4 Entrées des utilisateurs

Puis, nous avons ajouté une entrée par utilisateur sur le réseau interne. Chaque entrée possède des attributs leurs permettant d'authentifier des utilisateurs Linux, et des utilisateurs Samba. De ce fait, dans chaque entrée il est nécessaire d'ajouter le **SID** de Samba. Il possible de le récupérer en exécutant cette commande :

```
1 ssh etu@file-datacenter "net getlocalsid"
```

Fichier LDIF 3: `user.ldif`

```
dn: uid=mon_utilisateur,ou=users,dc=vortex-network,dc=fr
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
objectClass: sambaSamAccount
cn: Mon Utilisateur
sn: Utilisateur
uid: mon_utilisateur
uidNumber: 10000
gidNumber: 10000
homeDirectory: /home/mon_utilisateur
loginShell: /bin/bash
mail: mon_utilisateur@vortex-network.fr
sambaSID: S-1-5-21-XXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX-1000
sambaPrimaryGroupSID: S-1-5-21-XXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX-513
sambaAcctFlags: [U]
```

```
1 ldapadd -x -D "cn=admin,dc=vortex-network,dc=fr" -W -f ↩
    user.ldif
```

Pour ajouter l'utilisateur à un groupe, il est nécessaire d'ajouter un attribut `memberUid` à ce même groupe, avec comme valeur l'identifiant de l'utilisateur.

Fichier LDIF 4: `add_user.ldif`

```
dn: cn=Administrators,ou=groups,dc=vortex-network,dc=fr
changetype: modify
add: memberUid
memberUid: mon_utilisateur
```

```
1 ldapadd -x -D "cn=admin,dc=vortex-network,dc=fr" -W -f ↩
    add\_user.ldif
```

4.6 Serveur de fichiers

Pour mettre en place notre serveur de fichiers, nous avons utilisé le protocole SMB avec un serveur Samba. Nous l'avons configuré de telle sorte à ce qu'il utilise notre serveur LDAP (voir Section 4.5) pour l'authentification des utilisateurs. Ainsi, les utilisateurs et les administrateurs du réseau pourront accéder à leurs fichiers depuis n'importe quelle machine, si ils ont un compte dans le serveur LDAP.

Dans la configuration du serveur Samba, il a fallu ajouter l'URI de notre serveur LDAP, ainsi que le suffixe `dc=vortex-network,dc=fr`, ajouter le dn du compte administrateur du LDAP.

Fichier de configuration 1: `/etc/samba/smb.conf`

```
[global]
        bind interfaces only = Yes
        interfaces = 127.0.0.1 192.168.1.0/24 lo ens18
        ldap admin dn = cn=admin,dc=vortex-network,dc=fr
        ldap group suffix = ou=groups
        ldap ssl = no
        ldap suffix = dc=vortex-network,dc=fr
        ldap user suffix = ou=users
        log file = /var/log/samba/log.%m
        logging = file
        map to guest = Bad User
        max log size = 1000
        obey pam restrictions = Yes
        panic action = /usr/share/samba/panic-action %d
        passdb backend = ldapsam:"ldap://ldap-datacenter"
        security = USER
        server role = standalone server
        usershare allow guests = Yes
        idmap config * : backend = tdb
        server smb encrypt = desired

[homes]
        browseable = No
        comment = Home Directories
        create mask = 0700
        directory mask = 0700
        path = /srv/samba/homes
        read only = No
        valid users = %S
```

Pour finir, nous avons mis un mot de passe au compte administrateur de Samba, et avons redémarré le service `smbd` pour appliquer les modifications.

```
smbpasswd -W
systemctl restart smbd
```

4.7 Wikijs / WEB-datacenter

Pour installer et mettre en place Wiki.js, nous avons suivi plusieurs étapes :

1. Installation de Wiki.js et des dépendances sur le serveur `web-datacenter`

```
mkdir /var/wiki && cd /var/wiki
```

Faire les commande ci-dessous dans le répertoire de la commande au dessus :
Installation de Wikijs :

```
wget https://github.com/Requarks/wiki/releases/latest/download/wiki-js.tar.gz
tar -xzf wiki-js.tar.gz
```

Installation des dépendances soit `nodejs` :

```
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo bash -
sudo apt install -y nodejs
```

2. Configuration de la base de données sur le serveur `sgbd-datacenter` et sur le serveur `web-datacenter`.

A faire sur le serveur SGBD :

```
create user wikijs_user with ENCRYPTED PASSWORD '<VOTRE-MDP>';
create database with owner=wikijs_user;
```

A faire sur le serveur web :

```
cd /var/www/wikijs
nano config.yml
```

Ajouter ou modifier dans le fichier :

```
db:
  type: postgres

  # PostgreSQL / MySQL / MariaDB / MS SQL Server only:
  host: sgbd-datacenter.vortex-network.fr
  port: 5432
  user: wikijs_user
  pass: etu
  db: wikijs
  ssl: false
```

3. Créer un service pour démarrer le service `node` automatiquement sur le serveur `web-datacenter`

Faire les commandes ci dessous :

```
nano /etc/systemd/system/wikijs.service
```

Puis écrire dans se fichier :

Fichier de configuration 2: `/etc/systemd/system/wikijs.service`

```
[Unit]
Description=Wiki.js
After=network.target

[Service]
Type=simple
ExecStart=/usr/bin/node server
Restart=always
User=nobody
Environment=NODE_ENV=production
WorkingDirectory=/var/www/wikijs

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

Enfin effectuer ces commandes :

```
systemctl daemon-reload
systemctl enable --now wikijs.service
```

4. Configurer apache comme *reverse proxy* et mettre en place SSL pour notre wiki sur le serveur `web-datacenter`

   En premier modifier activer les modules ci dessous :

```
a2enmod proxy
a2enmod proxy_http
a2enmod headers
a2enmod ssl
```

   Puis créer le fichier comme ci dessous :

```
nano /etc/apache2/sites-available/wikijs.conf
```

   Mettre à l'intérieur le contenu ci dessous :

```
<VirtualHost *:443>
    ServerName web-datacenter.vortex-network.fr
    ServerAlias wiki.vortex-network.fr
    ServerAlias 192.168.1.30

    DocumentRoot "/var/www/wikijs"

    ProxyPreserveHost On
    ProxyPass / http://127.0.0.1:3000/
    ProxyPassReverse / http://127.0.0.1:3000/

    <Location />
        Require all granted
    </Location>

    SSLEngine On

    SSLCertificateFile /etc/ssl/wiki/wikijs.crt
    SSLCertificateKeyFile /etc/ssl/wiki/wikijs.key

    # Soucre : IT connect
    SSLProtocol -ALL +TLSv1 +TLSv1.1 +TLSv1.2
    SSLHonorCipherOrder On
    SSLCipherSuite ECDHE-RSA-AES128-SHA256:AES128-GCM-SHA256:HIGH:!MD5:!aNULL:!EDH:!RC4
    SSLCompression off

    ErrorLog ${APACHE_LOG_DIR}/wiki_error.log
    CustomLog ${APACHE_LOG_DIR}/wiki_access.log combined
</VirtualHost>
```

5. Générer notre clé et certificat SSL sur le serveur `web-datacenter`

   En premier il faut générez une clé identifiant le serveur avec :

```
openssl genrsa -out /etc/ssl/wiki/wikijs.key 4096
```

   Ensuite il faut générez une CSR en utilisant la commande :

```
openssl req -new -key /etc/ssl/wiki/wikijs.key -out wikijs.req
```

Ensuite on auto certifie la requête et on génère le certificat avec :

```
openssl x509 -req -days 30 -in wikijs.req -signkey /etc/ssl/wiki/wikijs.key -out /etc/ssl/wiki/wikijs.crt
```

Enfin on active le site et on redémarre le serveur avec :

```
a2ensite site1
systemctl restart apache2
```

6. Mise en place de l'authentification pour les administrateurs sur le site de notre wikijs

   En premier, une fois sur le site suivre les étapes de création de compte administrateur et se connecter (Voir Figure 1) et pour les explication de la connexion via LDAP (voir section 4.5).

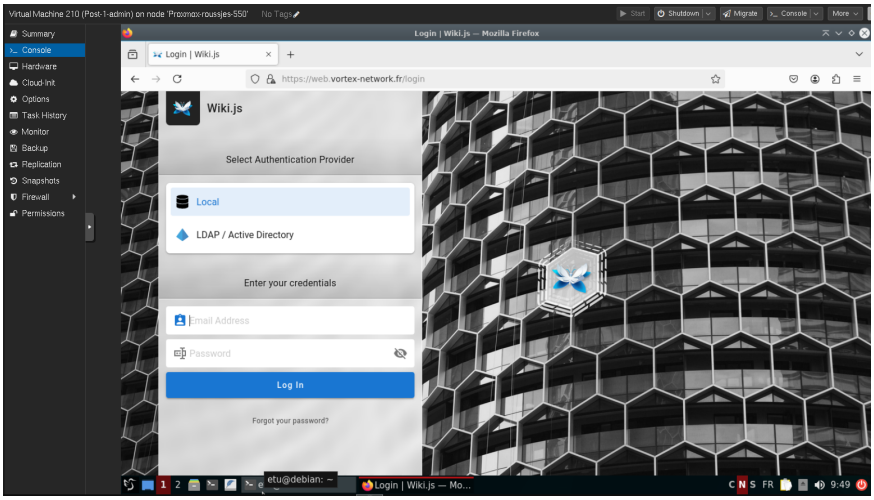


FIGURE 1 – Connexion wiki

Une fois connecter sur le compte administrateur voici l'interface (voir Figure 2) :
Voici une réécriture des paramètres de la configuration LDAP du wiki :
Nom d'affichage : LDAP / Active Directory

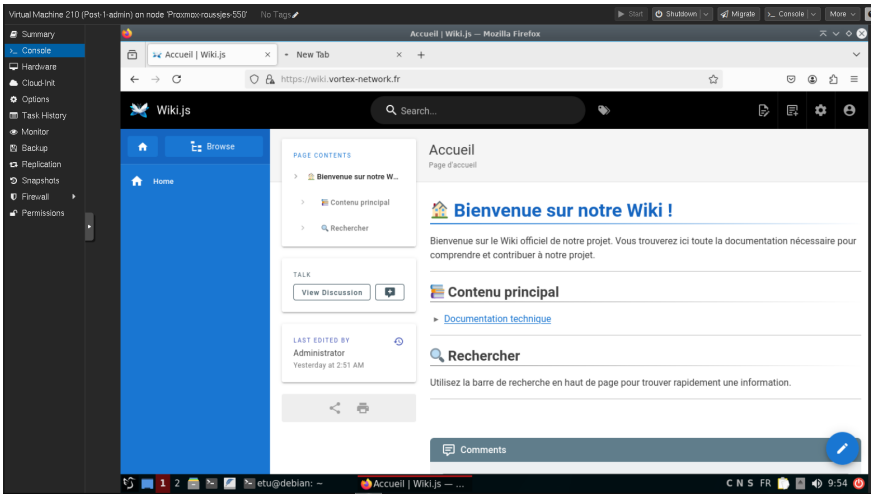


FIGURE 2 – Interface wiki

Configuration de la stratégie :
URL LDAP : `ldap://ldap-datacenter.vortex-network.fr:389`
Admin Bind DN : `cn=admin,dc=vortex-network,dc=fr`
Mot de passe Admin Bind : `root`
Base de recherche : `ou=users,dc=vortex-network,dc=fr`
Filtre de recherche : `(uid={{username}})`
Sécurité :
Utiliser TLS : Désactivé
Vérifier le certificat TLS : Désactivé
Chemin du certificat TLS : Non défini
Mappage des champs :
Identifiant unique : `uid`
Email : `mail`
Nom d'affichage : `cn`
Photo de profil : Non défini
Mappage des groupes LDAP :
Activer le mappage des groupes : Désactivé
Base de recherche des groupes : `ou=groups,dc=vortex-network,dc=fr`
Filtre de recherche des groupes : `(memberUid={{dn}})`
Portée de la recherche des groupes : `sub`
Propriété DN du groupe : `dn`
Champ de nom du groupe : `cn`
Enregistrement des utilisateurs :
Autoriser l'auto-inscription : Activé

Limiter à certains domaines email : Non défini

Groupes d'affectation automatique : Guests

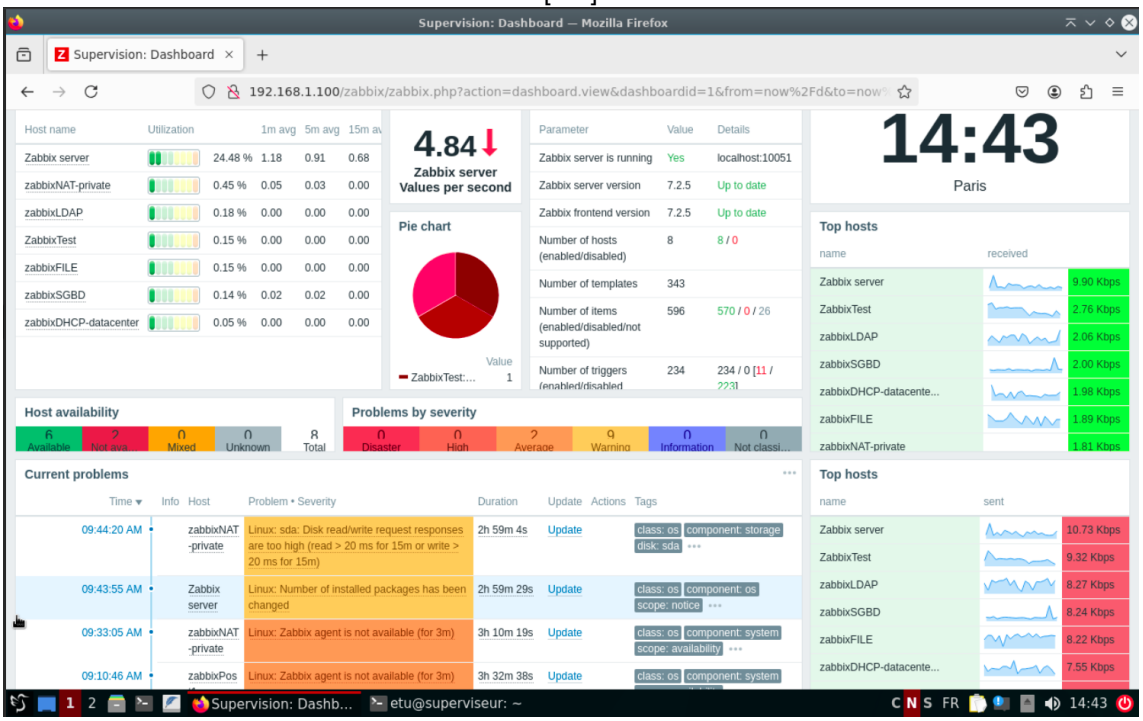
PS : Le site n'est accessible que pour les posts administrateur. Les posts employé et autres ne peuvent pas y accéder car la documentation technique est dessus. (voir détail partie RouteurNAT).

4.8 Supervision du réseau

Pour surveiller l'ensemble de l'infrastructure, nous avons choisi Zabbix. C'est un logiciel qui fait tourner un serveur sur une machine avec un tableau de bord regroupant toutes les informations. Un agent se déploie sur toutes les machines à surveiller pour pouvoir envoyer des données précises facilement. Le client se configure avec le fichier `/etc/zabbix/zabbix_agent2.conf`. Nous avons créé un script pour ajouter l'IP du serveur et le nom que l'agent aura.

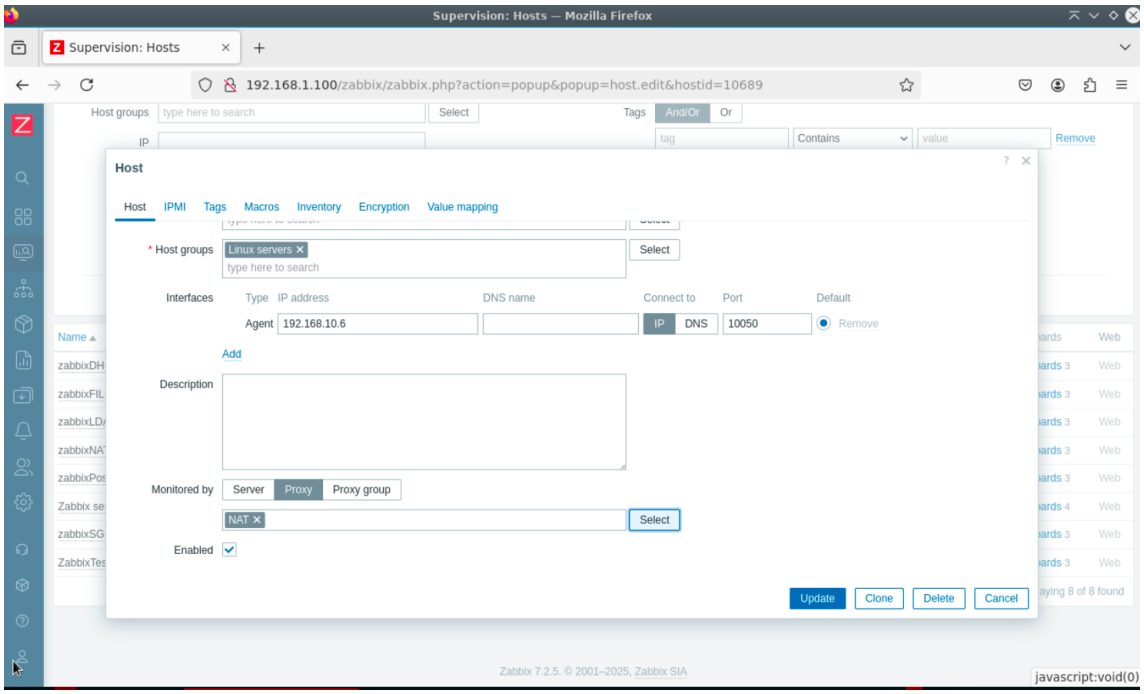
Zabbix nous a permis de centraliser la visualisation des événements et de nous séparer du serveur de logs. Nous avons également beaucoup de logs directement sur les routeurs pour pouvoir observer le trafic. Ce logiciel ne nous permet pas de le remplacer, mais il offre tout de même des données similaires à celles d'un SIEM, qui lui, nécessite beaucoup de ressources pour aller chercher les informations en profondeur.

[!ht]



Cette image nous montre le tableau de bord qui affiche les informations du réseau. Nous pouvons voir les ressources utilisées des machines (en haut à gauche), le nombre de machines démarrées et, par exemple, les entrées et sorties en Kb/s (en bas à droite).

Nous n'avons pas pu surveiller chaque machine du réseau par manque d'espace. Le serveur Zabbix étant sur un sous-réseau, il aurait fallu un proxy sur le routeur avec une base de données pour stocker les données de connexions. Même si, dans notre cas, il est compliqué de mettre la base de données en place, nous avons quand même regardé comment installer et configurer le proxy sur le serveur et dans l'interface graphique.



On peut voir sur l'image ci-dessus qu'il est possible d'ajouter un paramètre proxy en créant un nouvel hôte. Il faut également ajouter l'agent zabbix-proxy sur le serveur qui va servir de proxy.

4.9 Résolution de nom

Pour réaliser la résolution de nom a l'intérieur de notre réseau nous avons mis en place deux serveur DNS. Un est externe dans la DMZ, l'autre interne dans le sous-réseau datacenter.

4.9.1 Présentation BIND9

BIND9 permet de traduire les noms de domaine en adresses IP. Il prend en charge des fonctionnalités avancées comme la gestion des zones DNS, le DNSSEC (sécurisation du DNS).

4.9.2 configuration BIND9

1. `named.conf.options`
   Le fichier `/etc/bind/named.conf.options` est utilisé pour configurer les options globales du serveur DNS BIND9. On y définit des paramètres essentiels comme les serveurs

récursifs, les politiques de sécurité et les interfaces réseau écoutées. Parmi les éléments courants à configurer, on trouve :

— `forwarders` : liste des serveurs DNS vers lesquels les requêtes non résolues sont envoyées.
— `listen-on` : spécifie les adresses IP sur lesquelles BIND9 doit écouter.
— `allow-query` : contrôle les clients autorisés à interroger le serveur.
— `dnssec-validation` : active la validation DNSSEC pour renforcer la sécurité.

fichier `named.conf.options` de `dns-datacenter`

```
options {
        directory "/var/cache/bind";
        forwarders {
                192.168.1.1;
        };
        recursion yes;
        dnssec-validation auto;
        listen-on { any; };
        listen-on-v6 { any; };
        allow-query { lan; };
};
acl "lan" {
        192.168.1.0/24;
        localhost;
        localnets;
};
```

2. `named.conf.local`
Le fichier `/etc/bind/named.conf.local` est utilisé pour définir les zones DNS spécifiques à l'administrateur du serveur. Contrairement aux options globales, ce fichier permet de configurer des zones de résolution directe (`zone "vortex-network.fr"`), qui associent des noms de domaine à des adresses IP, ainsi que des zones de résolution inverse (`zone "1.168.192.in-addr.arpa"`), qui traduisent des adresses IP en noms de domaine.

fichier `named.conf.local` de `dns-datacenter`

```
zone "vortex-network.fr" {
    type master;
    file "/etc/bind/db.vortex-network.fr";
    allow-update { none; };
};
```

```
6
7   zone "1.168.192.in-addr.arpa" {
8       type master;
9       file "/etc/bind/db.reverse.vortex-network.fr";
10      allow-update { none; };
11  };
```

3. `/etc/bind/db.vortex-network.fr`
   Le fichier `/etc/bind/db.vortex-network.f` est un fichier de zone utilisé par le serveur DNS BIND9 pour définir les enregistrements DNS associés au domaine `vortex-network.f`. Ce fichier contient différentes informations essentielles comme :
   — Un enregistrement Start of Authority qui définit le serveur principal et les paramètres de gestion de la zone.
   — Des enregistrements Name Server indiquant les serveurs DNS responsables de la zone.
   — Des enregistrements `A` et `AAAA` associant des noms d'hôtes à des adresses IPv4 et IPv6.
   — Des enregistrements `CNAME` (alias) pour rediriger un sous domaine vers un autre nom de domaine.
   — Éventuellement des enregistrements `MX` pour la gestion des emails et `PTR` si la zone est utilisée pour une résolution inverse.

   fichier `db.vortex-network.fr` de `dns-datacenter`

```
1       $TTL    604800
2   @       IN      SOA     srv-dns.vortex-network.fr. ↩
        admin.vortex-network.fr.  (
3                               2         ; Serial
4                          604800         ; Refresh
5                           86400         ; Retry
6                         2419200         ; Expire
7                          604800 )       ; Negative ↩
                              Cache TTL
8   ;
9   @               IN      NS      srv-dns.vortex-↩
        network.fr.
10  srv-dns         IN      A       192.168.1.10
11  dns             IN      CNAME   srv-dns
12  dhcp-datacenter IN      A       192.168.1.20
13  web-datacenter  IN      A       192.168.1.30
14  sgbd-datacenter IN      A       192.168.1.40
```

```
file-datacenter IN      A       192.168.1.50
log-datacenter  IN      A       192.168.1.60
cas-datacenter  IN      A       192.168.1.70
ipam-datacenter IN      A       192.168.1.80
ldap-datacenter IN      A       192.168.1.90
superviseur-vortex IN      A       192.168.1.100
post1-post      IN      A       192.168.10.5
dhcp-admin      IN      A       192.168.2.3
dhcp-post       IN      A       192.168.10.3
dhcp-d          IN      CNAME   dhcp-datacenter
dhcp-a          IN      CNAME   dhcp-admin
dhcp-p          IN      CNAME   dhcp-post
web             IN      CNAME   web-datacenter
sgbd            IN      CNAME   sgbd-datacenter
file            IN      CNAME   file-datacenter
log             IN      CNAME   log-datacenter
cas             IN      CNAME   cas-datacenter
ipam            IN      CNAME   ipam-datacenter
ldap            IN      CNAME   ldap-datacenter
superviseur     IN      CNAME   superviseur-vortex
post1           IN      CNAME   post1-post
```

4.9.3 Tableau de corespondance IP/Nom de domaine

Alias	Nom Domaine	IP
dns	srv-dns.vortex-network.fr	192.168.1.10
dhcp-d	dhcp-datacenter.vortex-network.fr	192.168.1.20
web	web-datacenter.vortex-network.fr	192.168.1.30
sgbd	sgbd-datacenter.vortex-network.fr	192.168.1.40
file	file-datacenter.vortex-network.fr	192.168.1.50
log	log-datacenter.vortex-network.fr	192.168.1.60
cas	cas-datacenter.vortex-network.fr	192.168.1.70
ipam	ipam-datacenter.vortex-network.fr	192.168.1.80
ldap	ldap-datacenter.vortex-network.fr	192.168.1.90
superviseur	superviseur-vortex.vortex-network.fr	192.168.1.100
dhcp-a	dhcp-admin.vortex-network.fr	192.168.2.3
dhcp-p	dhcp-post.vortex-network.fr	192.168.10.3
post1	post1-post.vortex-network.fr	192.168.10.5

TABLE 14 – Tableau des alias et IPs du réseau privé

Alias	Nom Domaine	IP
dns	srv-dns.vortex-network.fr	192.168.0.2
web	web-dmz.vortex-network.fr	192.168.0.4

TABLE 15 – Tableau des alias et IPs du réseau privé

5 Sécurité Logiciel

Tous les logiciels mentionnés dans le tableau ci-dessous sont installés dans leur version la plus récente de Debian 12 "Bookworm". Ils sont régulièrement mis à jour pour garantir la sécurité de l'infrastructure.

5.1 Failles des logiciels utilisés dans l'infrastructure

Logiciel	Lien vers les failles
Bind9	https://security-tracker.debian.org/tracker/source-package/bind9
ISC-Kea	https://security-tracker.debian.org/tracker/source-package/isc-kea
Zabbix	https://security-tracker.debian.org/tracker/source-package/zabbix
OpenLDAP	https://security-tracker.debian.org/tracker/source-package/openldap
Samba	https://security-tracker.debian.org/tracker/source-package/samba
PostgreSQL 15	https://security-tracker.debian.org/tracker/source-package/postgresql-15
Apache2	https://security-tracker.debian.org/tracker/source-package/apache2

TABLE 16 – Failles de sécurité des logiciels utilisés

Glossaire

BIND9 Berkeley Internet Name Domain version 9 est un logiciel de serveur open source largement utilisé pour la résolution de noms de domaine sur Internet et dans les réseaux privés. . 2, 29–31

DHCP Dynamic Host Configuration Protocol (DHCP, protocole de configuration dynamique des hôtes) est un protocole réseau dont le rôle est d'assurer la configuration automatique des paramètres IP d'une station ou d'une machine, notamment en lui attribuant automatiquement une adresse IP et un masque de sous-réseau. [4]. 1, 14, 15

dn Le **distinct name** est un attribut spécial du format LDAP Data Interchange Format (LDIF). C'est un attribut requit pour la création de chaque entrée, et sert à identifier l'entrée de manière unique. . 21

IPAM La gestion des adresses IP (IP address management, IPAM) est une méthode mise en œuvre dans les logiciels informatiques pour planifier et gérer l'attribution et l'utilisation des adresses IP et des ressources étroitement liées d'un réseau informatique. Il ne fournit généralement pas de services DNS (Domain Name System) et DHCP (Dynamic Host Configuration Protocol), mais gère les informations pour ces composants [5] . 5

NAT En réseau informatique, on dit qu'un routeur fait du network address translation (NAT, « traduction d'adresse réseau » ou parfois « translation d'adresse réseau ») lorsqu'il fait correspondre des adresses IP à d'autres adresses IP. [6] . 5

SDN Le software-defined networking (SDN) est un modèle d'architecture réseau qui permet aux administrateurs de réseaux de gérer les services de réseaux par abstraction de fonctionnalités. [1] . 5

Simple Le User Datagram Protocol (UDP, en français protocole de datagramme utilisateur) est un des principaux protocoles de télécommunication utilisés par Internet. Il fait partie de la couche transport du modèle OSI, quatrième couche de ce modèle, comme TCP. [2] . 5

VXLAN Le VXLAN (Virtual Extensible LAN) est une technologie de virtualisation réseau qui vise à résoudre des problèmes d'évolutivité associés au déploiement du cloud computing. Il utilise une technique d'encapsulation proche du VLAN et permet d'encapsuler des trames Ethernet de couche 2 OSI dans des datagrammes UDP de couche 4. Le numéro de port UDP de destination par défaut attribué pour le VXLAN est le 4789. [3] . 4, 5

Acronymes

CSR Certificate Signing Request. 25

DMZ Zone Démilitarisée. 1, 5, 13, 29

DNS Domain Name System. 5, 13, 29–31

IP Internet Protocol. 13, 14, 28–30

LDAP Lightweight Directory Access Protocol. 1, 17, 18, 21, 26

LDIF LDAP Data Interchange Format. 18

NAT network address translation. 3, 5, 10, 11

NS Name Server. 31

NSS Name Service Switch. 14

PAM Pluggable Authentication Modules. 14

SDN Software-Defined Network. 4

SGBD Système de Gestion de Base de Données. 13, 22

SIEM Security Information and Event Management. 28

SMB Server Message Block. 14, 21

SOA Start of Authority. 31

SSH Secure Shell. 13

SSL Secure Sockets Layer. 24, 25

URI Uniform Resource Identifier. 21

Références

[1] Proxmox. *SDN*. URL : https://pve.proxmox.com/pve-docs/chapter-pvesdn.html.

[2] Proxmox. *Simple*. URL : https://pve.proxmox.com/pve-docs/chapter-pvesdn.html#pvesdn_zone_plugin_simple.

[3] Proxmox. *VXLAN*. URL : https://pve.proxmox.com/pve-docs/chapter-pvesdn.html#pvesdn_zone_plugin_vxlan.

[4] Wikipédia. *DHCP*. URL : https://fr.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol.

[5] Wikipédia. *IPAM*. URL : https://fr.wikipedia.org/wiki/Gestion_des_adresses_IP.

[6] Wikipédia. *NAT*. URL : https://fr.wikipedia.org/wiki/Network_address_translation.