BTS Services informatiques aux organisations **SESSION 2026**

ANNEXE VII-1-A : Fiche descriptive de réalisation professionnelle (recto)

Épreuve E6 - Administration des systèmes et des réseaux (option SISR)

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE

N° réalisation : 1

Nom, prénom : Susperregui Patxi | **N° candidat :** 02311961969

Épreuve ponctuelle | **Contrôle en cours de formation** | **Date :** 26/ 05/2026

Organisation support de la réalisation professionnelle

Intitulé de la réalisation professionnelle : Mise en place d'une infrastructure réseau

Période de réalisation : Février à Mai 2026 Lieu : Lycée St John Perse
Modalité : Seul

Compétences travaillées

Concevoir une solution d'infrastructure réseau
Installer, tester et déployer une solution d'infrastructure réseau
Exploiter, dépanner et superviser une solution d'infrastructure réseau

Conditions de réalisation[1] (ressources fournies, résultats attendus)

Description des ressources documentaires, matérielles et logicielles utilisées[2]

Modalités d'accès aux productions[3] et à leur documentation[4]

Situation n°1

Entreprise Patilab

Le laboratoire Patilab est un environnement informatique dédié à l'expérimentation et à l'apprentissage des technologies d'administration systèmes et réseaux. Cette infrastructure permet de simuler un environnement d'entreprise réaliste afin de tester des solutions de déploiement, de sécurisation et de haute disponibilité des services.

L'infrastructure est organisée autour d'un pare-feu pfSense, qui assure les fonctions de routage, de filtrage réseau et de sécurisation des échanges entre les différentes zones. Celui-ci est connecté à Internet via une interface WAN et segmente le réseau en plusieurs sous-réseaux distincts : un réseau local (LAN) et une zone démilitarisée (DMZ).

Dans la DMZ, un serveur dédié au reverse proxy de haute disponibilité, basé sur HAProxy, permet de répartir le trafic entrant vers plusieurs serveurs web. Cette architecture garantit une meilleure disponibilité des services en cas de panne d'un serveur.

Les serveurs HAProxy sont munie de Keepalived pour se prémunir de potentiel disfonctionnement sur l'un des HAProxy, ce qui permet d'assurer la redondance des deux HAProxy.

Deux serveurs web, configurés avec Apache2, sont déployés dans cette zone afin d'héberger des sites web. Les répartiteurs HAProxy distribue les requêtes entre ces serveurs, permettant ainsi d'optimiser les performances et d'assurer une continuité de service.

Sur le réseau local, un serveur Windows Server joue le rôle de contrôleur de domaine Active Directory et de serveur de fichier. Il permet la gestion centralisée des utilisateurs, des postes clients et des stratégies de sécurité. Deux postes clients sous Windows 11 sont intégrés au domaine afin de simuler un environnement utilisateur en entreprise. La partie serveur et en lien direct avec l'Active Directory car des GPO sont là pour assurer le montage des différents lecteurs réseaux, qui sont soumis à de diverse contrôle tel que les quotas et la gestion de fichier.

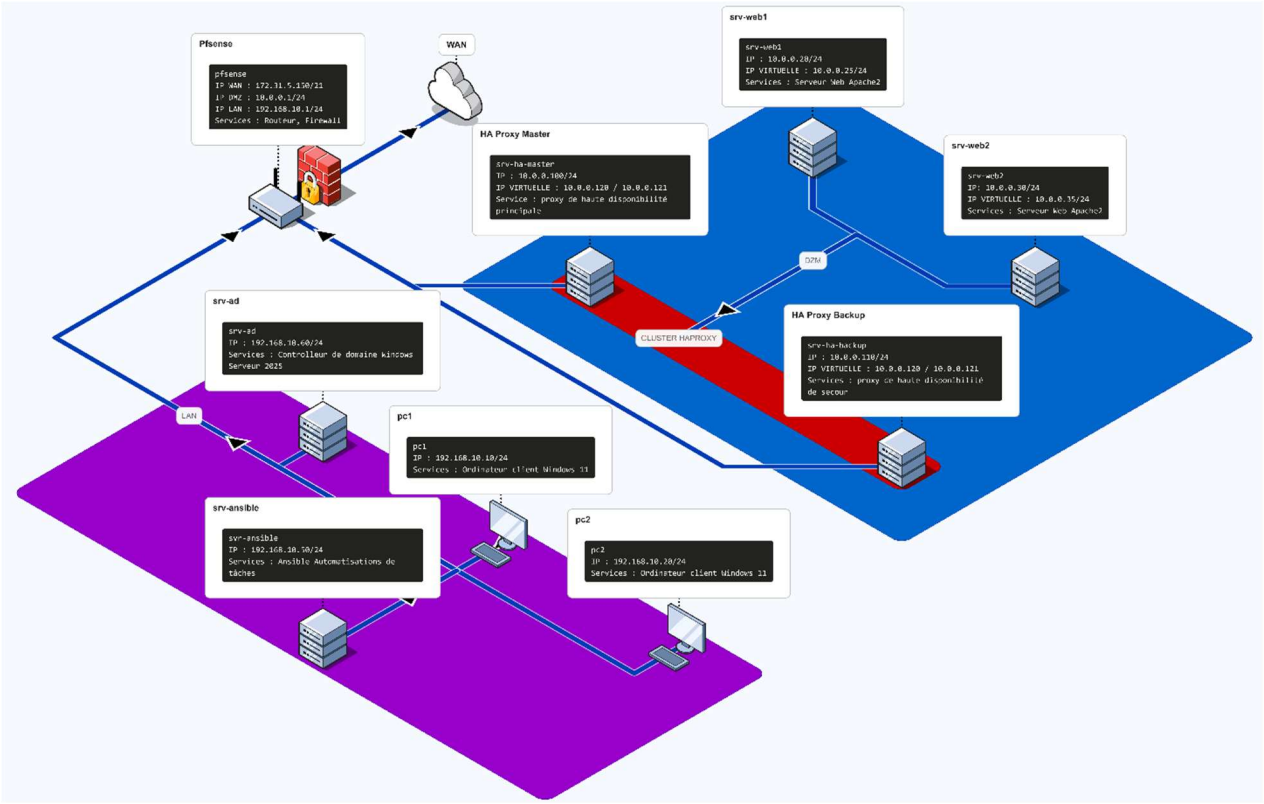
Un serveur dédié à l'automatisation, utilisant Ansible, est également présent sur le réseau local. Il permet de déployer, configurer et maintenir les différents serveurs de manière automatisée, facilitant ainsi l'administration et la reproductibilité des configurations.

Table des matières

Plan d'adressage IP :

Machines	Rôle	Adresse IP
pFsense	Routeur, Pare-feu	WAN 172.31.5.150/21 DMZ 10.0.0.1/24 192.168.10.1/24
DMZ		
Cluster haproxy	Serveurs de haute diponibilité	10.0.0.120/24 10.0.0.121/24
srv-haproxy-master	Serveur de haute diponibilité	10.0.0.100/24
srv-haproxy-backup	Serveur de haute diponibilité	10.0.0.110/24
srv-web1	Serveur web	10.0.0.20/24 10.0.0.25/24
srv-web2	Serveur web	10.0.0.30/24 10.0.0.35/24
LAN		
pc-user1	Client	192.168.10.10/24
pc-user2	Client	192.168.10.20/24
srv-ad	Controlleur de domaine,	192.168.10.60/24
srv-ansible	Automatisation Tâche	192.168.10.50/25

Schéma Réseaux



Fonctionnalités :

L'ajout de fonctionnalités permet aux postes et serveurs du réseau informatique d'être personnalisés, sécurisés et modulés selon les besoins et règles de l'entreprise.

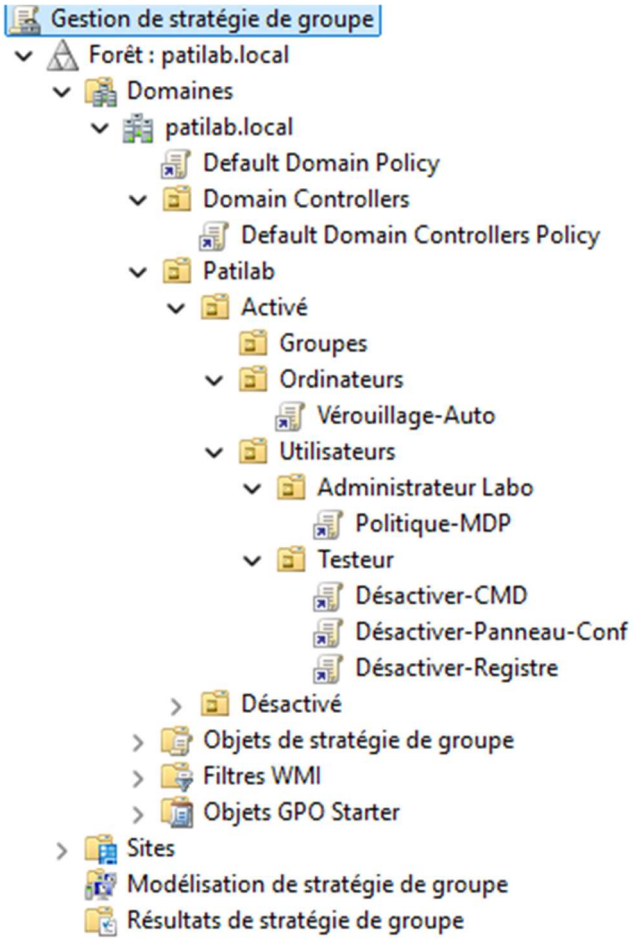
Service Active Directory :

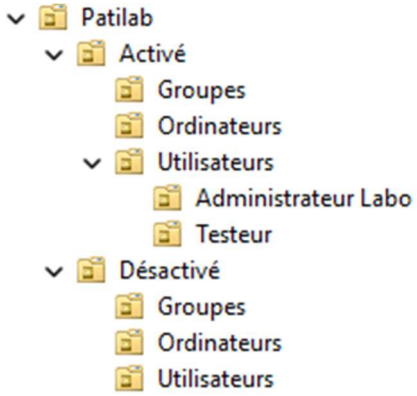
Forêt : patilab.local

Domaine : patilab.local

Gestion des stratégies :

- Sécurité des postes :
  - Verrouillage des postes au bout de 5 minutes d'inactivité
  - Politique de mot de passe accru pour les utilisateurs à risque, durée de vie maximal 90 et minimal 30 jours, exigences de complexité activée et longueurs minimale du mot de passe à 12 caractères
  - CMD, Registre et Panneau de configuration bloquer pour les utilisateurs classiques





Unité d'organisations :

- Domain Controllers :
  - SRV-AD
- Ordinateurs :
  - PC1
  - PC2
- Administrateur Labo :
  - Patxi (groupe Administrateur Labo)
  - Maïlys (groupe Administrateur Labo)
- Testeur :
  - Jean (groupe Testeur)
  - John (groupe Testeur)

Service Serveur de fichier :

Dossiers partagés, privilèges :

- Partage des testeurs
  - Lecture/Ecriture : groupe « Testeurs », « Administrateur »

- Partage Admins Labo
  - Lecture/Ecriture : groupe « Administrateurs Labo », « Administrateur »

- PARTAGE
  - Lecture/Ecriture : « tout le monde »

- Personnelle (« %username% »)
  - Lecture/Ecriture : utilisateur qui est connecté

Nom	Ordre	Action	Chemin d'accès	Reconnecter
P:	1	Mettre à jour	\\SRV-AD\partage	Oui
U:	2	Mettre à jour	\\SRV-AD\users-perso\%USERNAME%	Oui
A:	3	Remplacer	\\SRV-AD\administrateurs-labo	Non
B:	4	Remplacer	\\SRV-AD\testeurs	Non

Gestion de quotas :

Chemin d'accès du quota	% utilisé	Limite	Type de quota	Modèle source
Modèle source : Limite de 10 Go (1 élément)				
C:\Lecteurs Réseaux\partage	0%	10,0 Go	Inconditionnel	Limite de 10 Go
Modèle source : Limite de 2 Go (2 éléments)				
C:\Lecteurs Réseaux\administrateurs-labo	0%	2,00 Go	Inconditionnel	Limite de 2 Go
C:\Lecteurs Réseaux\testeurs	0%	2,00 Go	Inconditionnel	Limite de 2 Go
Modèle source : Limite de 5 Go (5 éléments)				
C:\Lecteurs Réseaux\users-perso\jdoe	0%	5,00 Go	Inconditionnel	Limite de 5 Go
C:\Lecteurs Réseaux\users-perso\jprade	0%	5,00 Go	Inconditionnel	Limite de 5 Go
C:\Lecteurs Réseaux\users-perso\mlalanne	0%	5,00 Go	Inconditionnel	Limite de 5 Go
C:\Lecteurs Réseaux\users-perso\psusperregui	0%	5,00 Go	Inconditionnel	Limite de 5 Go
C:\Lecteurs Réseaux\users-perso*	---	5,00 Go	Inconditionnel (Application automatique)	Limite de 5 Go

Gestion du filtrage de fichiers :

Chemin d'accès du filtre de fichiers	Type de filtrage	Groupes de fichiers	Modèle source
Modèle source : Bloquer les fichiers audio et vidéo (4 éléments)			
C:\Lecteurs Réseaux\partage	Actif	Bloquer : Fichiers audio et vidéo, Fichiers exécutables et images ISO	Bloquer les fichiers audio et vidéo
C:\Lecteurs Réseaux\users-perso	Actif	Bloquer : Fichiers audio et vidéo	Bloquer les fichiers audio et vidéo
C:\Lecteurs Réseaux\administrateurs-labo	Actif	Bloquer : Fichiers audio et vidéo, Fichiers exécutables et images ISO	Bloquer les fichiers audio et vidéo
C:\Lecteurs Réseaux\testeurs	Actif	Bloquer : Fichiers audio et vidéo, Fichiers exécutables et images ISO	Bloquer les fichiers audio et vidéo

Ansible :

Le serveur Ansible a été déployé afin d'automatiser l'administration des serveurs Linux de l'infrastructure. Il centralise l'exécution de tâches récurrentes sur l'ensemble des machines administrées via le protocole SSH.

L'automatisation est réalisée à l'aide de playbooks YAML appliqués à des groupes de machines définis dans un inventaire centralisé.

Fonctionnalités mises en place :

- Centralisation des hôtes administrés via inventaire ;
- Déploiement automatisé de paquets sur plusieurs serveurs ;
- Déploiement de fichiers de configuration ;
- Exécution de tâches d'administration en masse ;
- Standardisation des configurations Linux de l'infrastructure.

Fichier de configuration Ansible :

inventory.ini :

```
[windows]
pc1 ansible_host=192.168.10.10 ansible_user=sio ansible_password=sio
pc2 ansible_host=192.168.10.20 ansible_user=sio ansible_password=sio
srv-ad ansible_host=192.168.10.60 ansible_user=Administrateur@patilab.local ansible_password=Adminsio64@

[windows:vars]
ansible_connection=winrm
ansible_port=5985
ansible_winrm_transport=basic
ansible_winrm_server_cert_validation=ignore

[domaine_controller]
srv-ad ansible_host=192.168.10.60

[domain_member]
pc1
pc2
```

ansible.cfg :

```
[defaults]
inventory =./inventory.ini
host_key_checking = False
remote_user = sio
```

Playbooks/install-vlc.yml :

```
- name: Installer VLC sur win11
  hosts:
      - pc1
      - pc2
  gather_facts: no

  tasks:
    - name: Créer le dossier temporaire
      ansible.windows.win_file:
        path: C:\Temp
        state: directory

    - name: Télécharger le MSI officiel de VLC
      ansible.windows.win_get_url:
        url: https://download.videolan.org/vlc/last/win64/vlc-3.0.23-win64.msi
        dest: C:\Temp\vlc-3.0.23-win64.msi

    - name: Installer VLC
      ansible.windows.win_package:
        path: C:\Temp\vlc-3.0.23-win64.msi
        state: present
```

Playbooks/uninstall-vlc.yml :

```
- name: Désinstaller VLC sur win11
  hosts:
      - pc1
      - pc2
  gather_facts: no

  tasks:
    - name: Désinstaller VLC
      ansible.windows.win_package:
        path: C:\Temp\vlc-3.0.23-win64.msi
        state: absent
```

Playbooks/join-domain.yml :

```
- name: Faire rejoindre les postes Windows au domaine patilab.local
  hosts: domain_member
  gather_facts: false

  vars:
    domain_name: patilab.local
    domain_admin_user: Administrateur@patilab.local
    domain_admin_password: "sio"

  tasks:
    - name: Rejoindre le domaine
      microsoft.ad.membership:
        dns_domain_name: "{{ domain_name }}"
        domain_admin_user: "{{ domain_admin_user }}"
        domain_admin_password: "{{ domain_admin_password }}"
        state: domain
        reboot: true
      register: domain_join_result

    - name: Afficher le résultat
      ansible.builtin.debug:
        var: domain_join_result
```

pfSense :

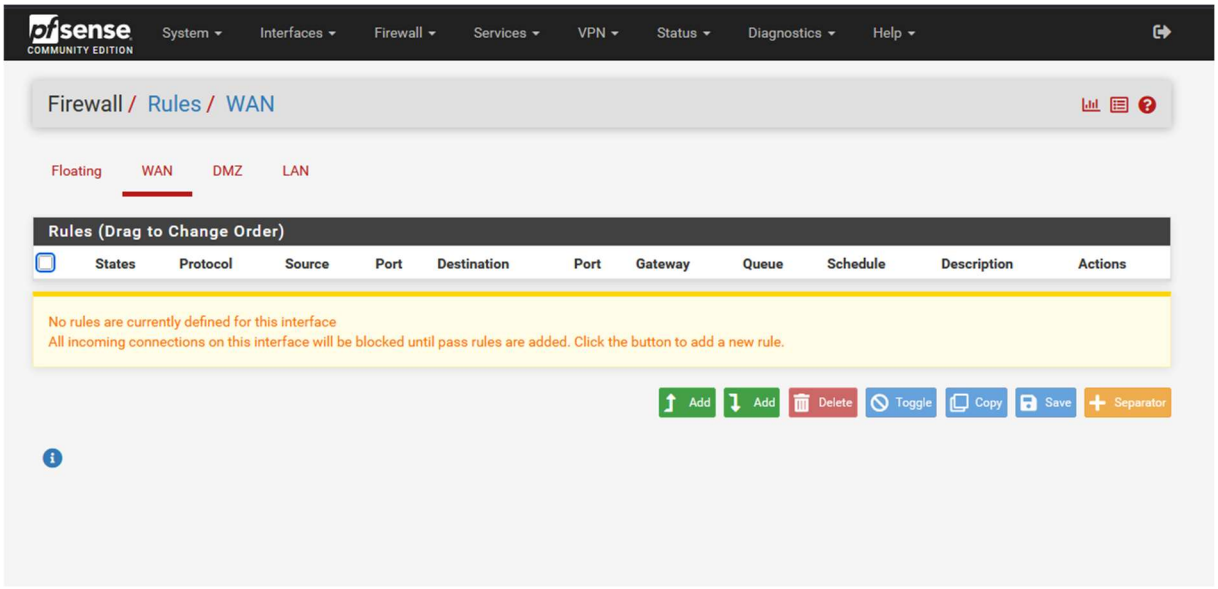
```
FreeBSD/amd64 (pfSense.home.arpa) (ttyv0)

VirtualBox Virtual Machine - Netgate Device ID: ac95e10ff619fcb853c6

*** Welcome to pfSense 2.7.2-RELEASE (amd64) on pfSense ***

 WAN (wan)       -> em0        -> v4: 172.31.5.150/21
 DMZ (lan)       -> em1        -> v4: 10.0.0.1/24
 LAN (opt1)      -> em2        -> v4: 192.168.10.1/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell

Enter an option:
```

Le pfSense dispose de trois interfaces réseaux :

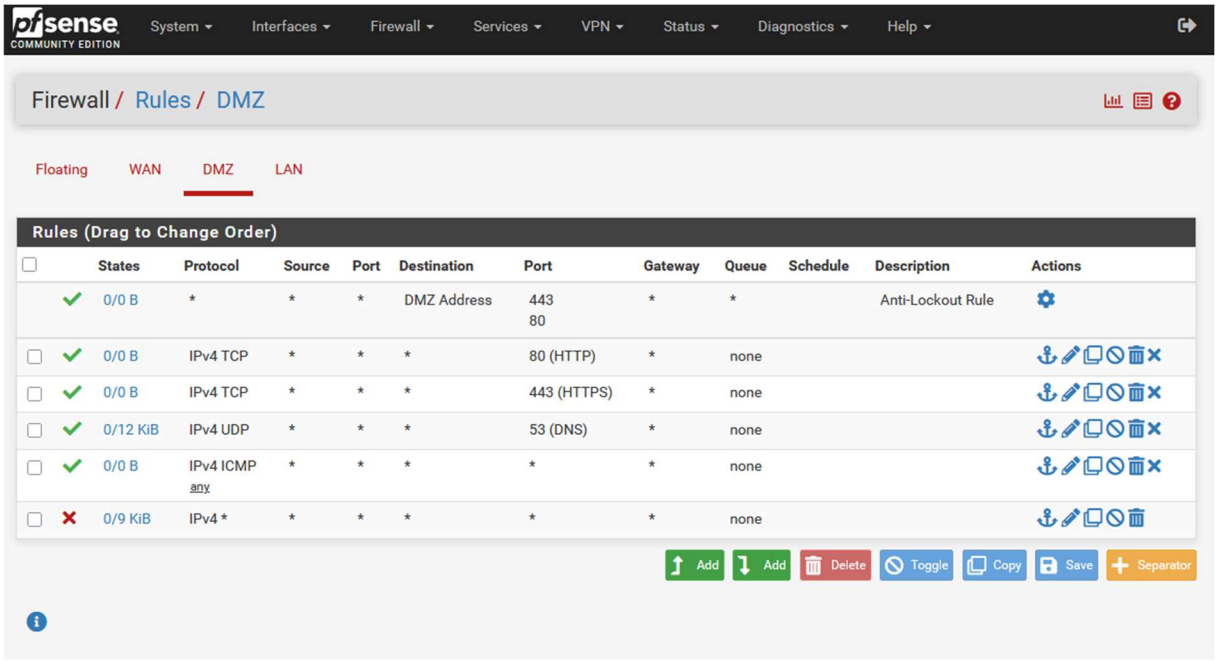
- WAN : 172.31.5.150/21
- DMZ : 10.0.0.1/24
- LAN : 192.168.10.1/24

Chaque interface possèdes ses propres règles de pare-feu

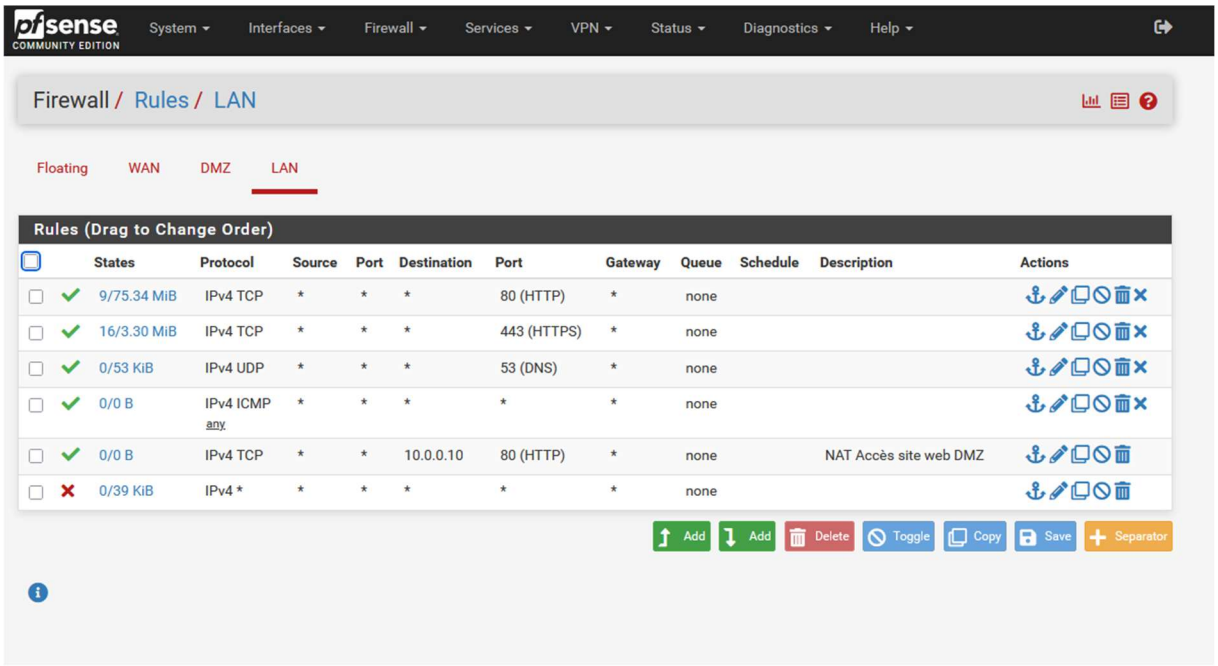
WAN :



DMZ :

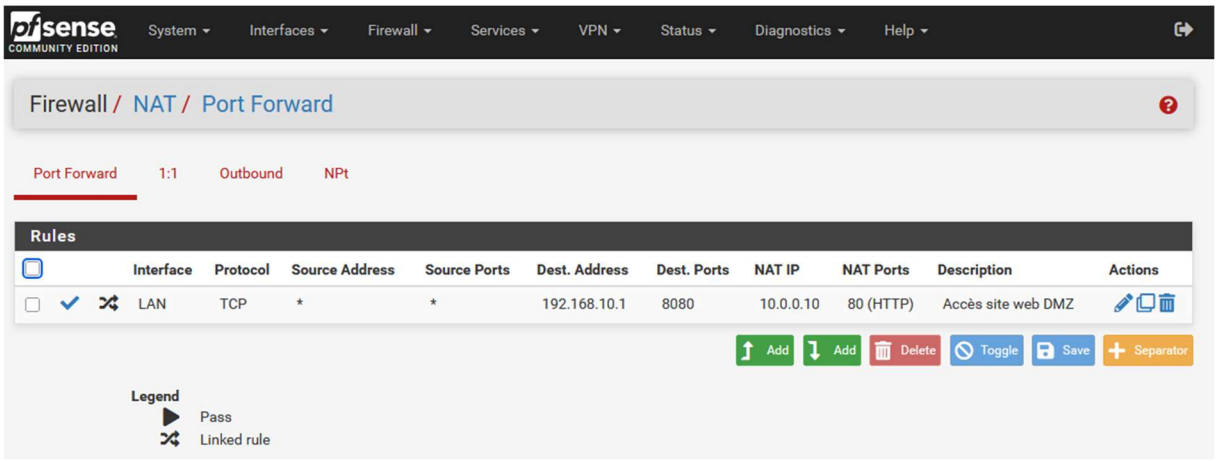


LAN :



Les règles autorisant les ports 443, 80, 53 et le ICMP permet une utilisation d'internet conviviale.

De plus l'interfaces LAN possède un règle NAT qui permet au réseau LAN d'avoir accès au site web de la DMZ.



Elle permet de rediriger le flux arrivant vers « 192.168.10.1:8080 » (adresse du routeur et un port ou il ne se trouve rien) vers le serveur haproxy qui se trouve en DMZ.

Service Haute disponibilité :

Un serveur de haute disponibilité est mis en place dans la DMZ pour assurer la disponibilité des serveurs WEB. Le service HAProxy a été installé et configurer pour répondre à cette tâche.

Ligne ajouter au fichier de configuration de HAProxy (/etc/haproxy/haproxy.cfg) :

Config Master

```
frontend site1_front
    bind 10.0.0.120:80
    mode http
    default_backend web_servers

frontend site2_front
    bind 10.0.0.121:80
    mode http
    default_backend site2_backend

backend web_servers
    mode http
    balance roundrobin
    option httpchk
    server srv-web1 10.0.0.20:80 check
    server srv-web2 10.0.0.30:80 check

backend site2_backend
    mode http
    balance roundrobin
    option httpchk
    server srv-web1-ipv 10.0.0.25:80 check
    server srv-web2-ipv 10.0.0.35:80 check

listen stats
    bind *:81
    mode http
    stats enable
    stats uri /haproxy?stats
    stats refresh 10s
    stats show-node
    stats admin if TRUE
```

Config Backup

```
frontend site1_front
    bind 10.0.0.120:80
    mode http
    default_backend web_servers

frontend site2_front
    bind 10.0.0.121:80
    mode http
    default_backend site2_backend

backend web_servers
    mode http
    balance roundrobin
    option httpchk
    server srv-web1 10.0.0.20:80 check
    server srv-web2 10.0.0.30:80 check

backend site2_backend
    mode http
    balance roundrobin
    option httpchk
    server srv-web1-ipv 10.0.0.25:80 check
    server srv-web2-ipv 10.0.0.35:80 check

listen stats
    bind *:81
    mode http
    stats enable
    stats uri /haproxy?stats
    stats refresh 10s
    stats show-node
    stats admin if TRUE
```

Keepalived :

Ajout de scripts Keepalived pour redémarrer HAProxy automatiquement

Dans le cadre de la haute disponibilité, les serveurs HAProxy utilisent Keepalived afin de partager deux adresses IP virtuelles : 10.0.0.120 10.0.0.121

Ces IP virtuelles permettent d'accéder aux services web même si le serveur HAProxy principal devient indisponible.

Cependant, HAProxy peut rencontrer un problème au démarrage si les IP virtuelles ne sont pas encore présentes sur la machine. Pour corriger cela, Keepalived exécute automatiquement un script lorsqu'un serveur devient MASTER, BACKUP ou FAULT.

Configuration Keepalived du serveur MASTER

```
global_defs {
    router_id HA_MASTER
    enable_script_security
    script_user root
}

vrrp_instance VI_1 {
    state MASTER
    interface enp0s3
    virtual_router_id 51
    priority 200
    advert_int 1

    authentication {
        auth_type PASS
        auth_pass MonMotDePasseHa
    }

    virtual_ipaddress {
        10.0.0.120/24
        10.0.0.121/24
    }

    notify_master "/etc/keepalived/scripts/master.sh"
    notify_backup "/etc/keepalived/scripts/backup.sh"
    notify_fault "/etc/keepalived/scripts/fault.sh"
}
```

Configuration Keepalived du serveur BACKUP

```
global_defs {
    router_id HA_BACKUP
    enable_script_security
    script_user root
}

vrrp_instance VI_1 {
    state BACKUP
    interface enp0s3
    virtual_router_id 51
    priority 100
    advert_int 1

    authentication {
        auth_type PASS
        auth_pass MonMotDePasseHa
    }

    virtual_ipaddress {
        10.0.0.120/24
        10.0.0.121/24
    }

    notify_master "/etc/keepalived/scripts/master.sh"
    notify_backup "/etc/keepalived/scripts/backup.sh"
    notify_fault "/etc/keepalived/scripts/fault.sh"
}
```

Rôle des options ajoutées :

« enable_script_security »

Cette option autorise Keepalived à exécuter des scripts de notification de manière sécurisée.

Sans cette ligne, Keepalived affiche l'erreur suivante :

SECURITY VIOLATION - scripts are being executed but script_security not enabled.

« script_user root »

Cette option indique que les scripts seront exécutés avec l'utilisateur root.

Cela est nécessaire ici, car le script doit pouvoir redémarrer le service HAProxy.

Création des scripts

Dossier des scripts :

mkdir -p /etc/keepalived/scripts

Script exécuté quand le serveur devient MASTER :

nano /etc/keepalived/scripts/master.sh

Script exécuté quand le serveur devient BACKUP :

nano /etc/keepalived/scripts/backup.sh

Script exécuté en cas d'état FAULT :

nano /etc/keepalived/scripts/fault.sh

Contenu de tous les script :

#!/bin/bash

systemctl restart haproxy

Sécurisation des scripts

Les scripts doivent appartenir à root et être exécutables :

chown -R root:root /etc/keepalived/scriptschmod 700 /etc/keepalived/scripts/*.sh

Test de la configuration

La configuration Keepalived est testée avec la commande :

keepalived -t -f /etc/keepalived/keepalived.conf

Si aucune erreur bloquante n'apparaît, le service peut être redémarré :

systemctl restart keepalived

Puis vérifié avec :

systemctl status keepalived

Service Web:

Les serveurs web possèdes des ip virtuelles pour accueillir plusieurs site web sur le même serveur.

Pour se faire le fichier de configuration de l'interfaces réseau a été modifié (etc/network/interfaces) :

```
allow-hotplug enp0s3
iface enp0s3 inet static
address 10.0.0.20/24
gateway 10.0.0.1

    up ip addr add 10.0.0.25/24 dev enp0s3
    down ip addr del 10.0.0.25/24 dev enp0s3
```

De plus un fichier de configuration apache a été créer pour indiquer quelle ip et quel port va accueillir ce deuxième site web, avec également la localisation des fichiers du site (etc/apache2/sites-available/site2.conf):

```
<VirtualHost 10.0.0.25:80>
    ServerName site2.local
    DocumentRoot /var/www/site2
</VirtualHost>
```