Atelier Professionnel:

Sommaire:

Contexte:

Dans le cadre de votre formation BTS, vous avez trouvé un stage de 6 semaines chez un prestataire informatique local.

Ce prestataire emploie actuellement 1 technicien (niveau Bac Professionnel) qui intervient essentiellement sur des opérations informatiques courantes:

- Nettoyage et formatage d'ordinateurs
- Diagnostic et dépannage matériel
- Remplacement de pièces (sur ordinateur fixes et portables)
- Installation / dépannage d'imprimantes
- Installation et paramètrage de petits réseaux locaux simples

La clientèle de l'entreprise est composée de 80% de clients particuliers et 20% de clients professionnels (petits artisans, TPE et quelques petites mairies et écoles communales).

Le gérant conscient de l'évolution du marché et des nombreuses demandes de clients professionnels, souhaiterait développer son offre auprès des PME en fournissant des architectures système et réseau évoluées avec un support technique adapté.

Afin d'évaluer la faisabilité du développement de l'activité et de prendre la décision de lancer un service dédié aux professionnels (PME, collectivités locales), le gérant profite de votre période de stage et des compétences que vous possédez pour vous charger de :

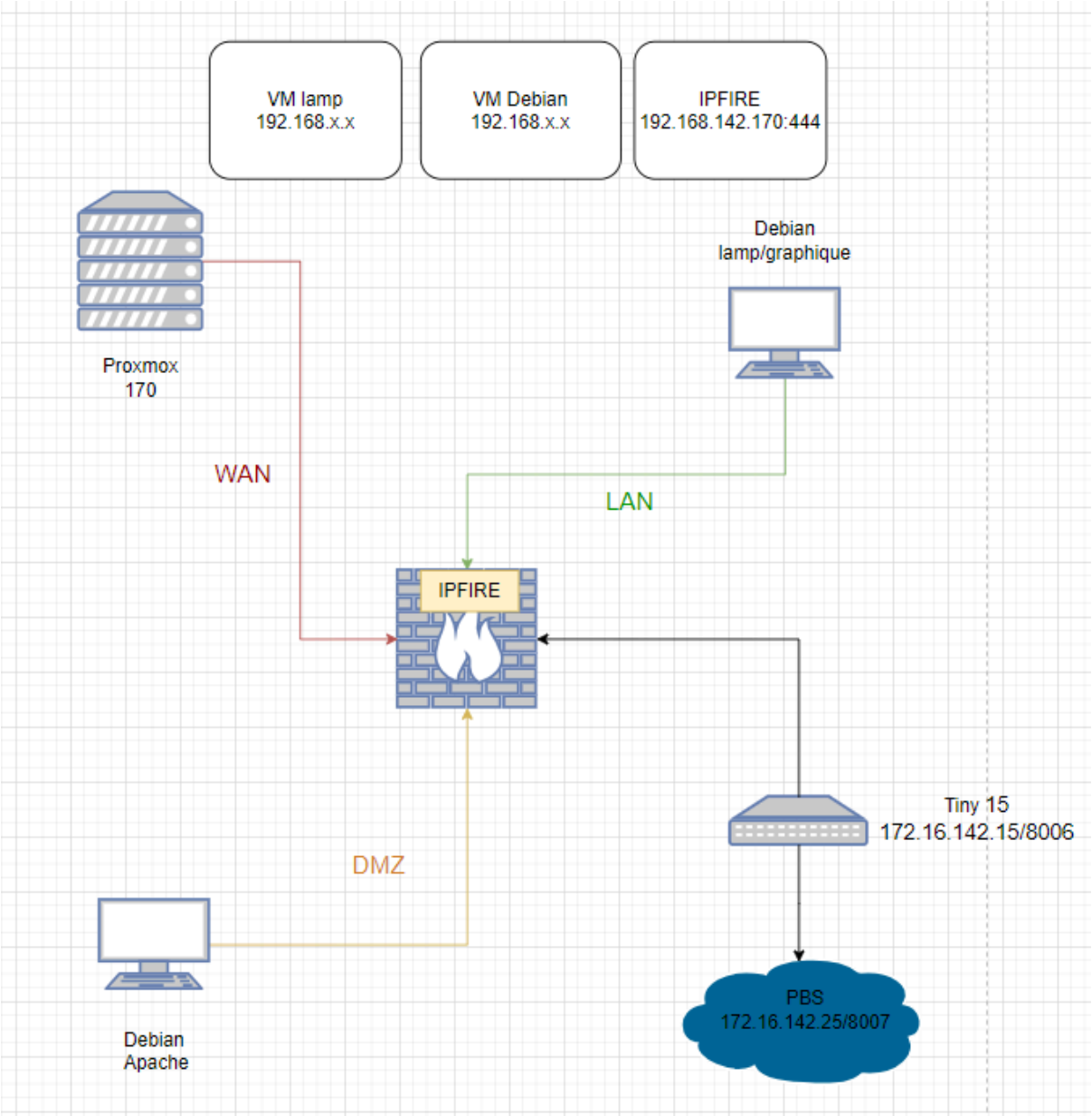
- Tester des infrastructures virtualisées
- De privilégier les solutions Open Source

Le gérant vous suggère d'utiliser l'hyperviseur Proxmox et le routeur/pare-feu open source IPFIRE dans un premier temps.

Pour réaliser vos tests, vous avez à votre disposition une machine 'serveur' (unité centrale ou portable) sur laquelle vous pourrez mettre en œuvre les différentes missions de tests demandées.

Les machines mises à disposition possèdent à minima, un disque système et un disque supplémentaire qui sera dédié aux futures 'vms'.

Architecture réseau:



Proxmox:

Accès au Proxmox : 172.16.142.170:8006

Nom	Utilisateur PAM	Mdp	Utilisateur VE	Adresse ip
Gwendoline	root	adminsio1	Gwenne	172.16.142.171
Mattéo	root	adminsio1	Matt	172.16.142.172

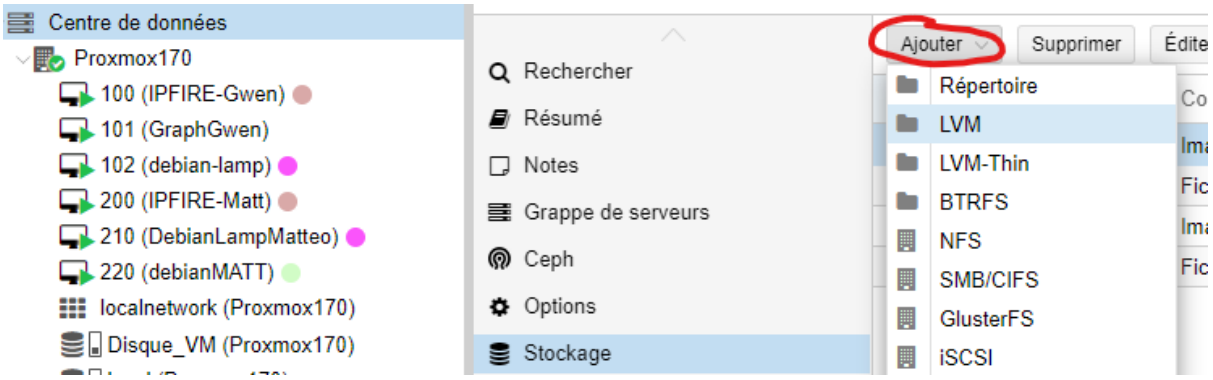
Notre objectif était d'installer un proxmox sur un serveur à partir d'une clé bootable. Pour cela nous avons commencé par aller chercher l'iso de Proxmox sur son site officiel, puis une fois l'iso telecharger nous avons utilisé le logiciel Baleina Etcher qui nous permet de rendre un fichier iso bootable à partir d'une clé USB.

Une fois la clé bootable prête, il suffit de l'insérer dans le port usb du pc qui servira de serveur et que la page de configuration du Proxmox se lance. Une fois lancé nous avons suivi les étapes d'installation, attribution des disques, création de l'utilisateur PAM (nom d'utilisateur/MDP) et attribution de l'adresse IP du serveur (172.16.142.170).

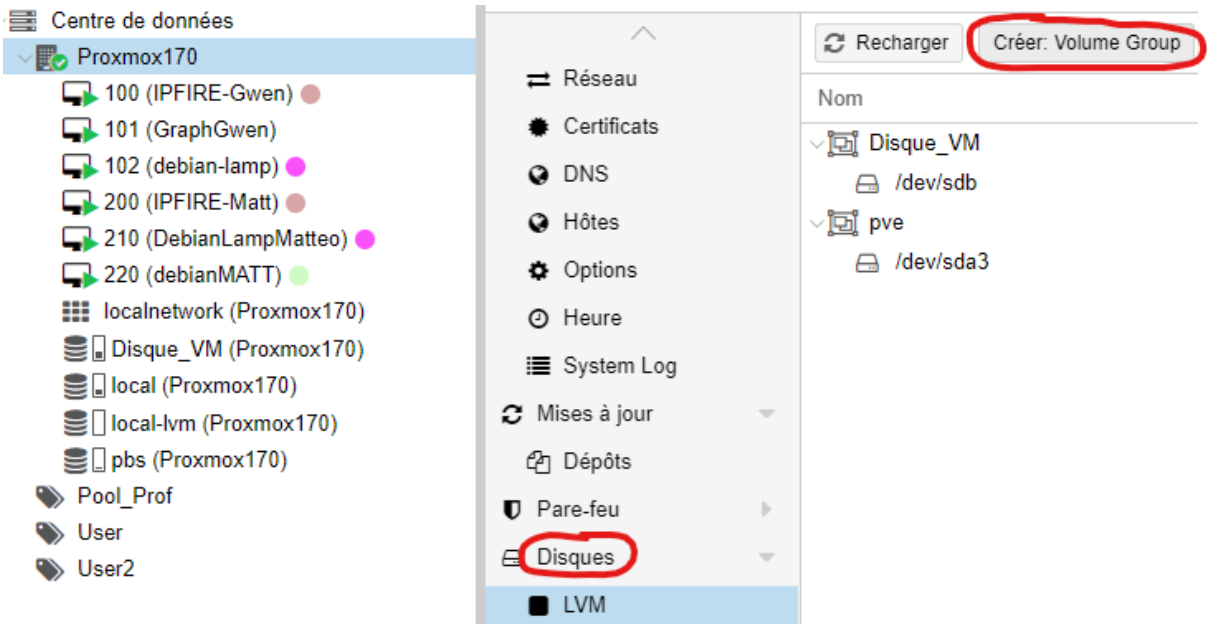
Une fois que le serveur proxmox a été mis en place, nous sommes entrés sur le serveur en tant que root, nous avons ajouté un disque "Disque VM" et un volume pour stocker nos VM et nous avons créé nos 2 premières VM, des vm

debian (lamp) en cli, non graphique, pour vérifier que tout fonctionnait bien sur notre serveur, de la création au fonctionnement des VM.

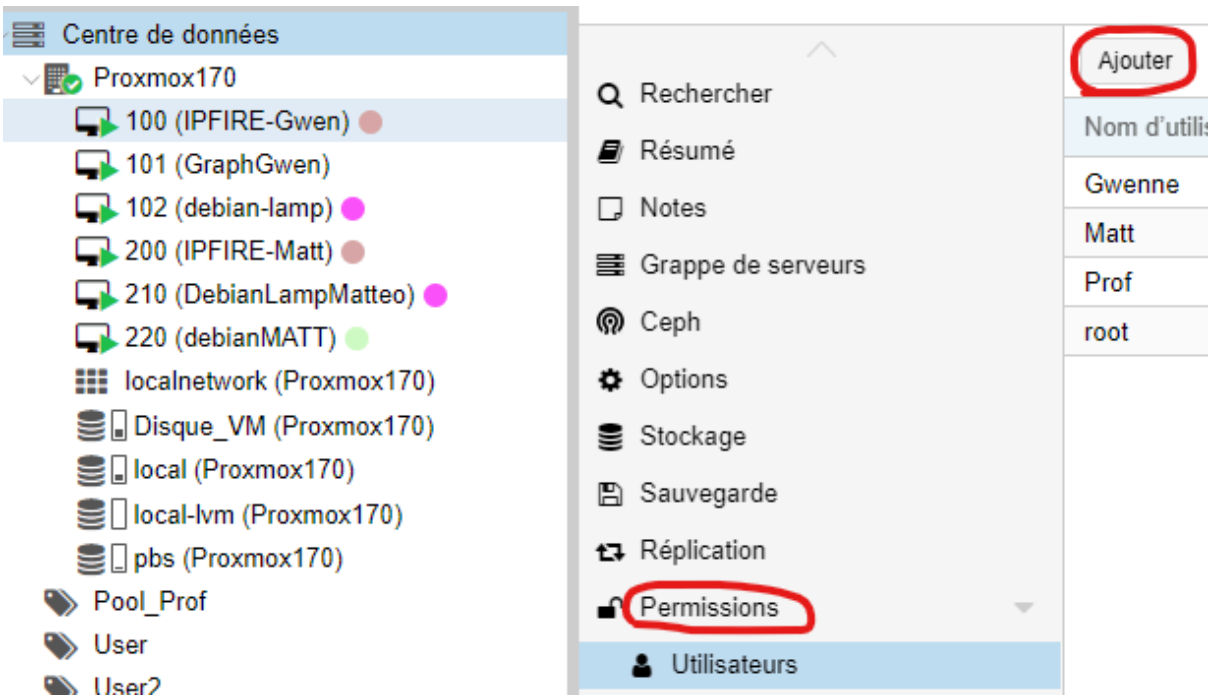
Pour ajouter un disque sur Proxmox, il suffit de se rendre sur son Proxmox sur l'onglet "Centre de données", puis de cliquer sur "stockage", "ajouter" et "LVM".



Après avoir ajouté le disque, il faut lui attribuer un volume. Pour cela, il suffit de se rendre sur l'onglet "Proxmox(numéro du serveur/170)" puis sur l'onglet "Disques", "LVM" et "Créer: Volume Group".



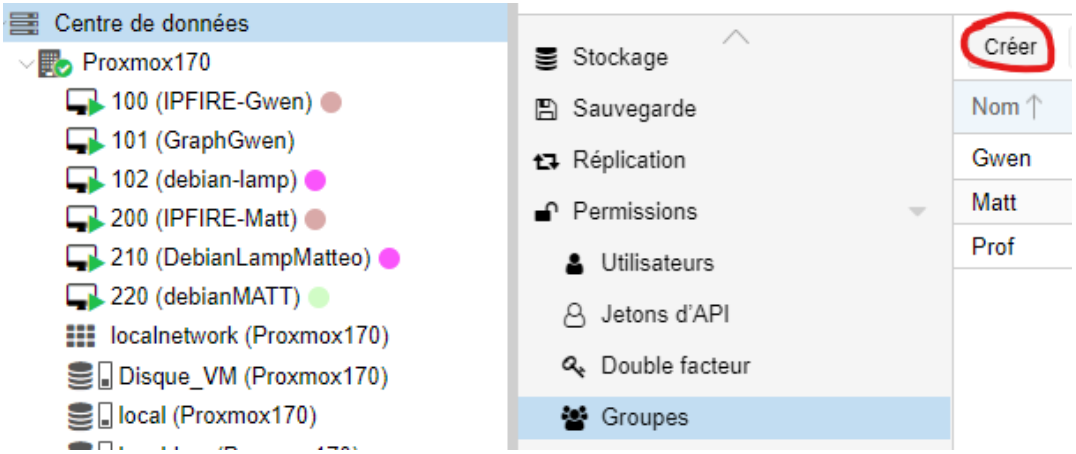
Une fois que nous avons vérifié que le Proxmox fonctionnait bien, que le disque stockait bien les vm et que les vm fonctionnait bien, nous avons créé des utilisateur VE qui sont des utilisateurs gérés par l'utilisateur PAM. Pour cela, il faut cliquer sur "Centre de données", "Permissions", "Utilisateur" et "Ajouter".



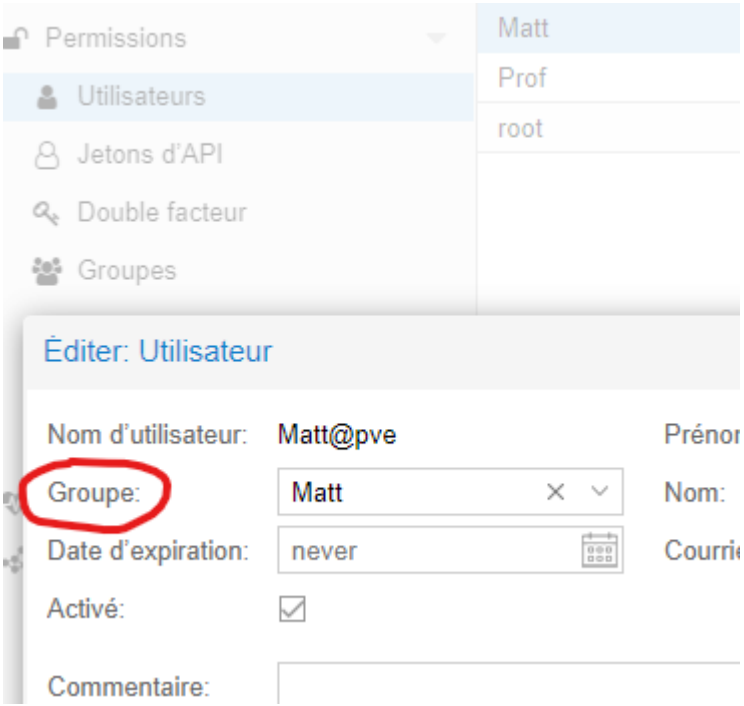
Ensuite notre but avec ces utilisateurs VE était qu'ils aient accès aux VM qu'ils ont créé sans qu'ils aient accès aux VM des autres utilisateurs VE. Pour cela nous avons donc dû leur attribuer des rôles/permissions.

Pour attribuer des permissions il est plus facile d'auparavant créer des groupes ou l'on assignera les utilisateurs pour que dans le futur nous ayons juste à ajouter les nouveaux utilisateurs à chaque groupe sans avoir à attribuer les permissions pour chaque utilisateur.

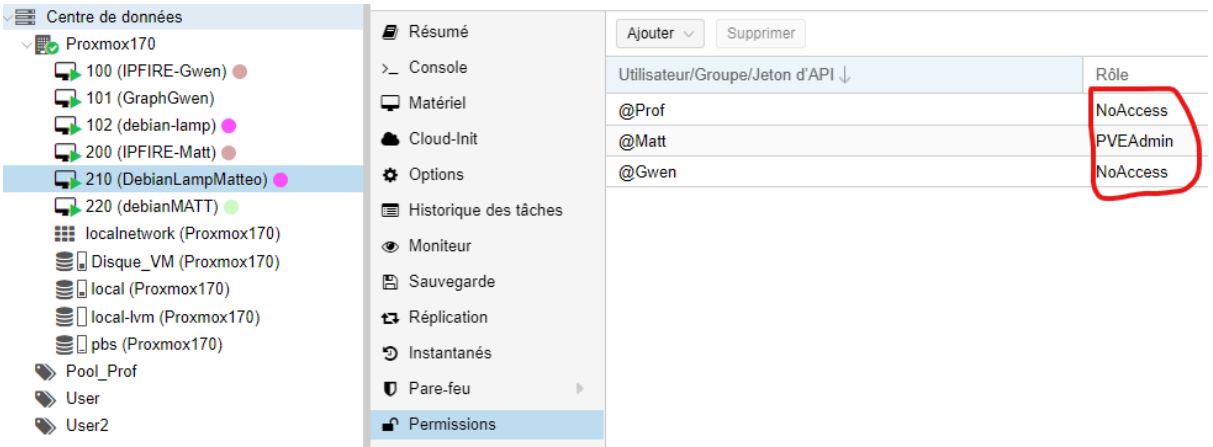
Pour ajouter un utilisateur à un groupe, il est d'abord essentiel de créer un groupe. Pour cela, il faut cliquer sur "Centre de données", "Groupes" et "Créer".



Après avoir créé le(s) groupe(s), il suffit de revenir sur l'onglet "Utilisateur" et d'éditer l'utilisateur que vous voulez mettre dans le groupe.



Au niveau des différentes permissions à ajouter, il faut donner l'accès au vm appartenant à chaque utilisateur, bloquer l'accès au VM aux utilisateurs qui n'en sont pas propriétaires et aussi d'accorder l'accès au disque qui doit contenir les VM et à celui qui contient les ISO aux différents utilisateurs. Pour cela, il suffit d'attribuer la permission "PVEadmin" pour autoriser une personne a voir et à se connecter à la VM, et "NoAccess" aux personnes dont on ne veut pas qu'il ait accès à certaines VM. Ces actions sont à faire en tant qu'administrateur root (PAM) pour chaque VM.



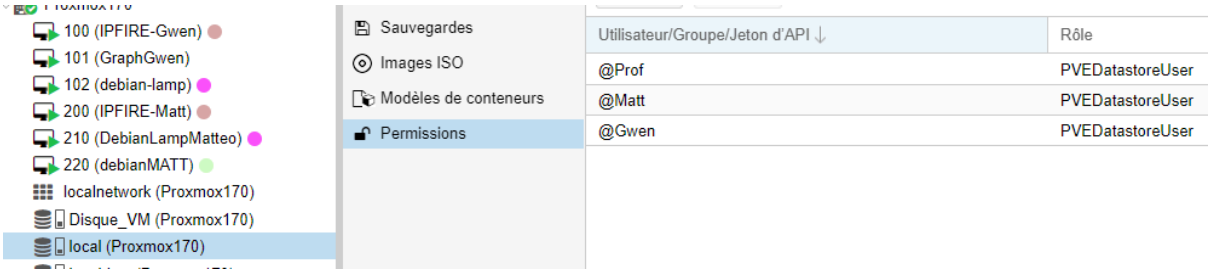
Par exemple dans le cas ci-dessus nous avons une Debian appartenant à l'utilisateur Mattéo. Le but est donc que lui puisse voir sa machine et s'y connecter sans que les autres utilisateurs VE puissent faire la même chose car ce n'est pas eux qui l'ont créé.

Ensuite, il faut attribuer des droits aux utilisateurs VE pour l'accès aux disques qui servent à stocker les VM et les fichiers ISO.

Pour cela, il suffit de se rendre sur le disque auquel on veut accorder des permissions, puis cliquer sur "permissions" et enfin, cliquer sur "Ajouter".

Pour le disque 'Disque VM' en donne le rôle "PVEADMIN" aux utilisateurs,

et pour le disque local contenant les fichier ISO en donne le rôle "PVE Datastore User".



Debian LAMP:

Nom	MDP root	utilisateur	MDP
Mattéo	adminsio1	utilisateur	adminsio1
Gwendoline	adminsio	user	adminsio

LAMP est un acronyme pour Linux, Apache, MySQL et PHP. C'est un groupe de logicielle comprenant:

- le système d'exploitation; (Linux)
- un serveur HTTP; (Apache)
- un système de gestion de bases de données; (MySQL)
- un langage de programmation interprété; (PHP)

L'ensemble de ces éléments permet de mettre en place un serveur web.

Matériel attribué à la Debian : 2Go de ram, 1 cœur, allocation d'un espace de stockage "Disque VM" de 20Go et attribution d'une carte réseau.

Une machine Debian LAMP est la base du Proxmox et elle nous sert à beaucoup de choses et notamment à créer des "stacks" pour Glpi, Next Cloud et Portainer qui par la suite seront accessibles via un navigateur web en tapant la Wan.

Pour faire en sorte que la Debian devienne une Debian LAMP il faut installer les paquets nécessaires grâce à des "apt-get install" suivie du nom du paquet comme par exemple le paquet "glpi".

Pour finir, il existe quelques commandes de bases qui sont importantes. Par exemple les commandes "apt update" et "apt upgrade" qui servent a mettre a jour l'ensemble des paquets de la Debian, "mkdir" pour créer un dossier, "docker ps -a" pour voir la liste et les statuts des conteneurs présent sur la Debian ou encore "ls"pour afficher les dossiers présent à l'endroit où l'on se situe.

IPFIRE:

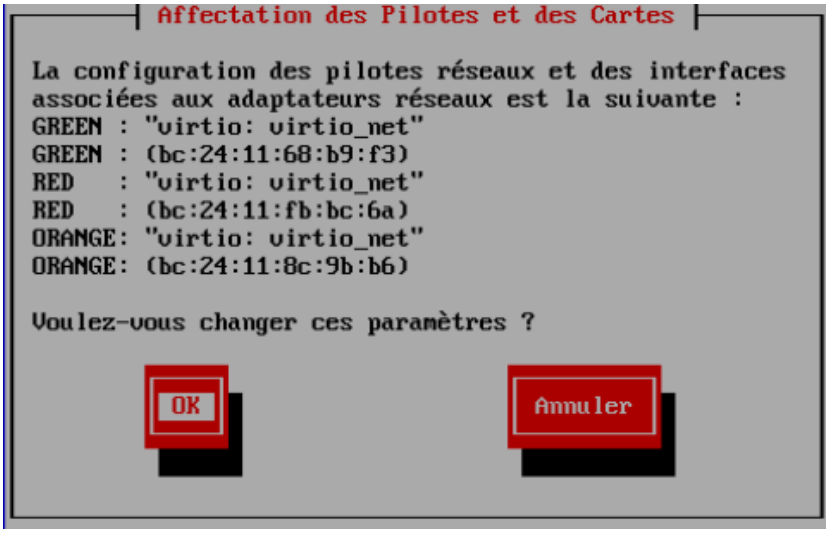
Nom	IPFIRE CLI	IPFIRE
Mattéo	root - adminsio1	admin - adminsio1
Gwendoline	root - adminsio1	admin - usersio

Adresse: 172.16.142.17X:444

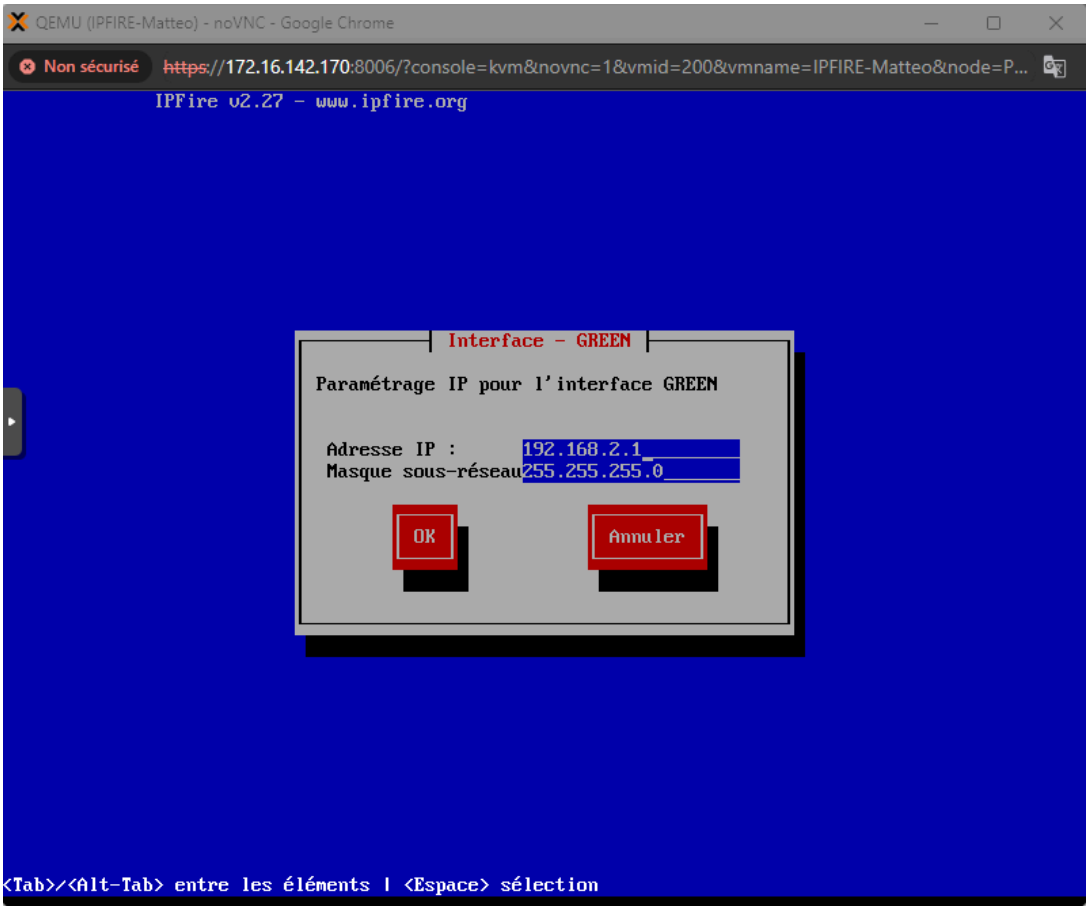
Pour notre IPFIRE, il va nous falloir 3 configurations réseaux, un qui nous servira de WAN (rouge) pour notre Proxmox, la LAN (vert) pour notre Debian Graphique et une DMZ (orange) qui nous servira d'hébergement pour notre serveur Apache (debian LAMP).

Il nous faudra aussi trois interfaces pour chaque réseau: le vmbr0 pour la WAN, le vmbr1 pour la LAN et le vmbr11 pour la DMZ donc il faut 3 vmbr pour chaque utilisateur. Lors de l'attribution des interfaces il faut prendre en compte l'adresse mac que chaque interfaces possède et ne pas se tromper lorsqu'on affecte des cartes réseaux aux types de réseaux.

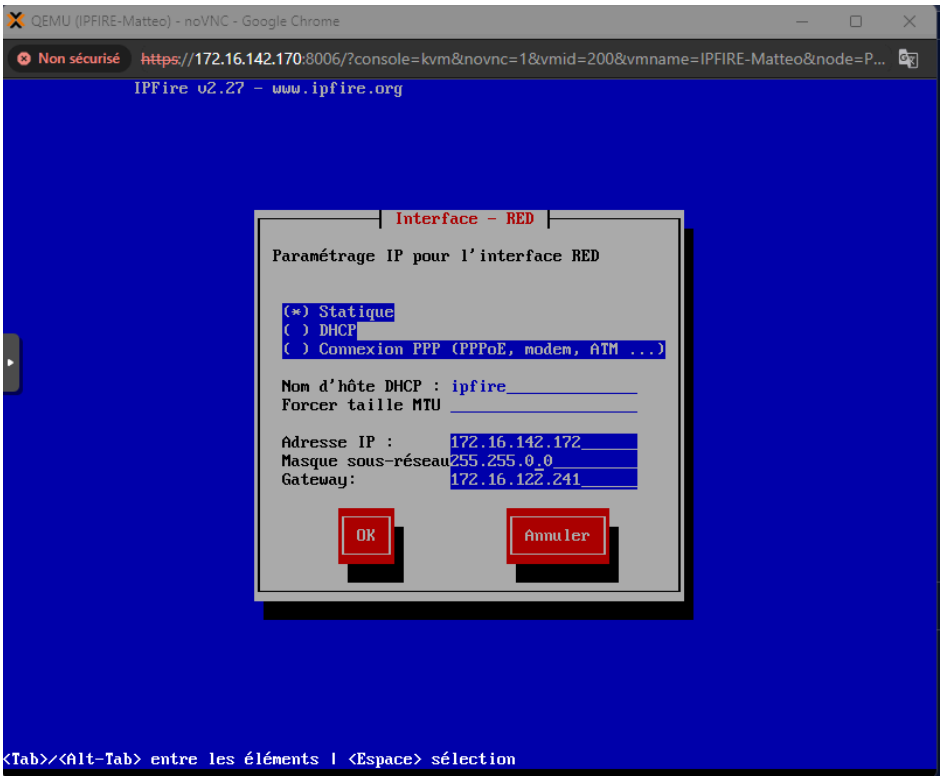
⇄ Carte réseau (net0)	virtio=BC:24:11:FB:BC:6A,bridge=vmbr0,firewall=1
⇄ Carte réseau (net1)	virtio=BC:24:11:68:B9:F3,bridge=vmbr1,firewall=1
⇄ Carte réseau (net2)	virtio=BC:24:11:8C:9B:B6,bridge=vmbr11,firewall=1



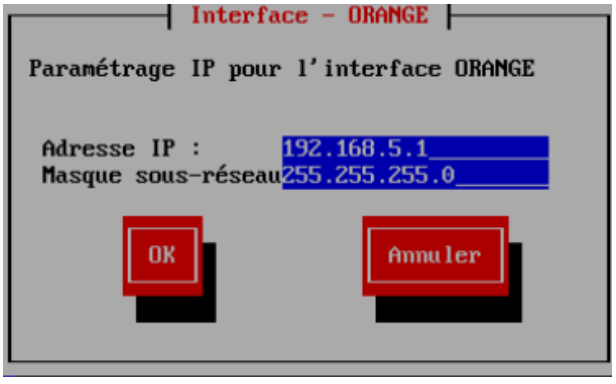
Après cela, on va attribuer une adresse réseau pour chaque interface, la LAN (192.168.x.x).



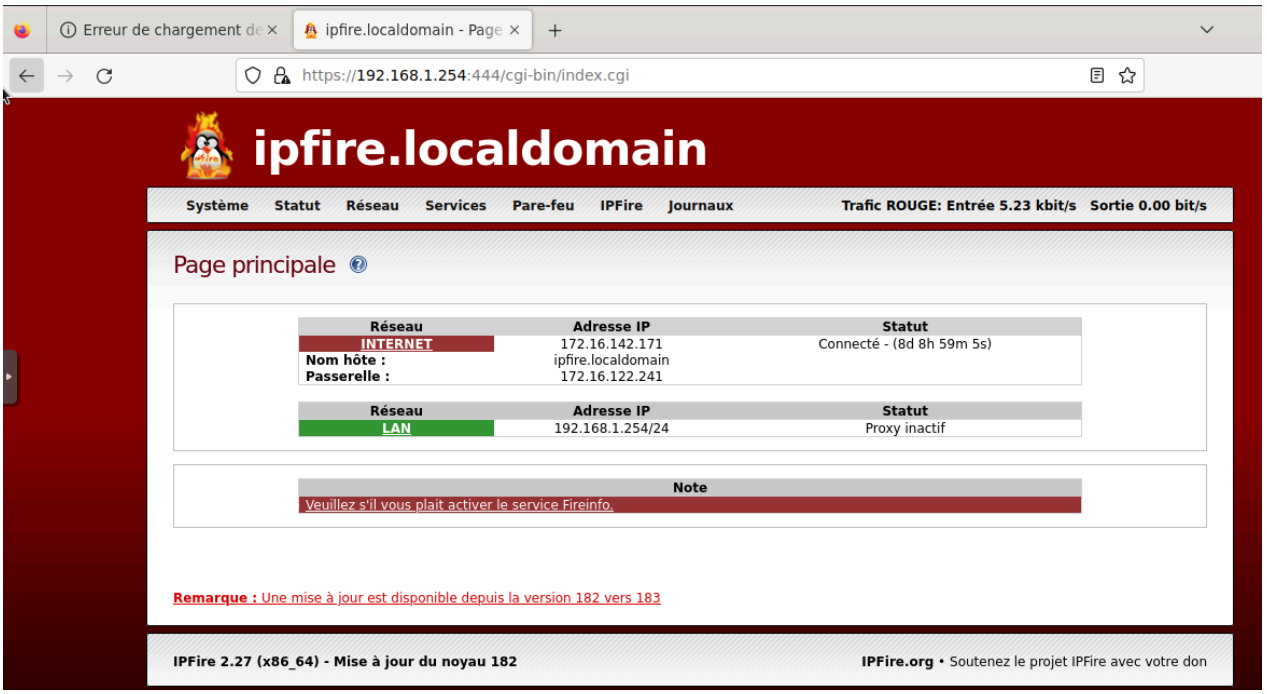
Pour la WAN (rouge) on lui attribue une adresse de 172.16.142.17X, ce qui va nous permettre d'accéder à la page d'IPFIRE sur notre réseau 172.16.136.x.



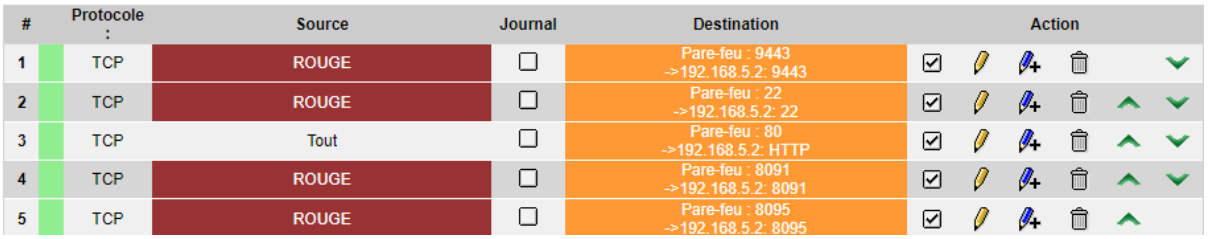
Pour la DMZ, on lui attribue une adresse de type classe C comme la LAN sauf qu'on évite de mettre 'la même adresse IP', pour le troisième bit on le remplace par un autre pour éviter les confusions.



Après l'installation terminée de l'IPFIRE, on pourra accéder à notre IPFIRE en tapant l'adresse 172.16.142.17x:444.



A partir de là, on pourra rajouter des règles de pare-feu dans l'IPFIRE comme le ssh, http ou encore portainer qui nous serviront plus tard. On peut aussi mettre en place un proxy web qui nous permettra de limiter les accès à des réseaux sociaux ou des sites malveillants.

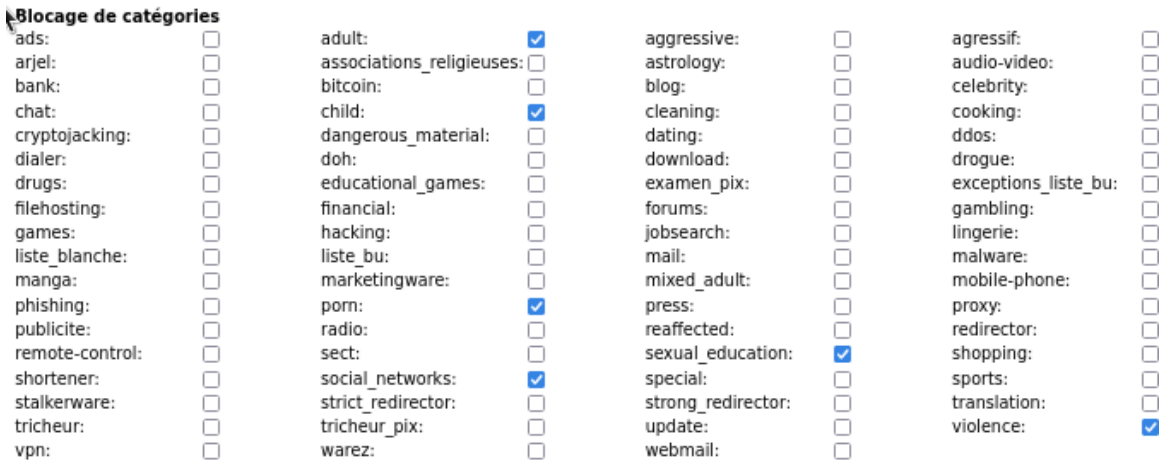


Debian Graphique:

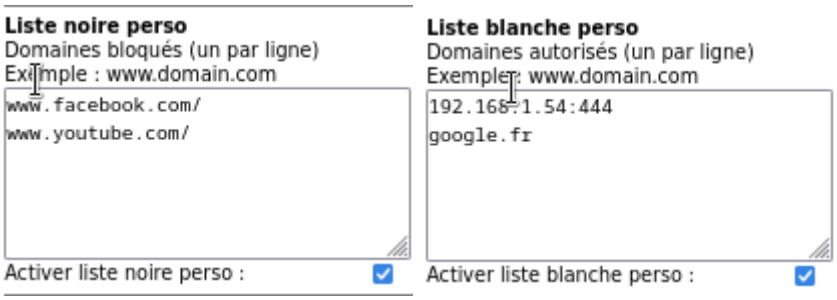
Nom	root	utilisateur
Mattéo	adminsio1	adminsio1
Gwendoline	adminsio1	adminsio1

Mission: Votre environnement de travail est maintenant fonctionnel avec une architecture réseau reposant sur une interface « WAN », une interface « LAN » et une interface dédiée à la « DMZ ». Votre serveur « LAMP » est fonctionnel et vous accédez à votre serveur web Apache depuis l'extérieur et depuis le réseau local. Vous avez veillé à ce que vos règles de pare-feu empêchent tout intervenant extérieur de pénétrer votre réseau interne (LAN) en ajoutant les règles nécessaires dans votre pare-feu. Une question est cependant soulevée par votre responsable au sujet de la navigation web au sein du réseau local : est-il envisageable de bloquer l'accès à certains sites et, plus particulièrement, l'accès aux réseaux sociaux (Facebook par exemple) ?

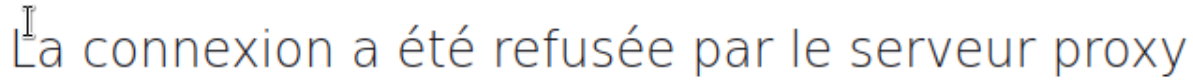
Notre VM Debian Graphique, nous sert plus de test pour mettre en place un proxy web ou un squid sur la debian LAMP. Le proxy web / squid va nous permettre de limiter les accès aux utilisateurs qui essaient d'aller sur des sites dangereux voire utiliser par des attaquants ou alors limiter certains accès pendant un certain moment dans la journée. On peut instaurer des règles dans l'IPFIRE pour bloquer un accès à des sites avec des phrases ou des mots clés qui sont cochés ci-dessous:



On peut même créer une liste pour savoir si on autorise les accès ou non. Ces listes s'appellent listes noires et blanches. Les listes noires bloquent l'accès aux sites comme les réseaux sociaux et pour les listes blanches ce sont les sites qui sont autorisés d'accès. Cette technique peut permettre d'éviter aux utilisateurs d'aller sur n'importe quel domaine.



Si on essaie d'accéder à facebook alors que ce dernier est bloqué, on obtient la fenêtre par laquelle il n'est pas autorisé à aller sur facebook.



GLPI:

	Gwendoline	Mattéo
Identifiant	glpi	glpi
Mot de passe	glpi	glpi

Adresse: 172.16.142.17X:8095

Mission: Votre responsable a validé ces premières missions et souhaiterait que vous testiez le déploiement de diverses applications open-source qui répondent à une demande des clients professionnels, notamment la solution de type « Helpdesk » de GLPI et l'application Nextcloud qui permet la mise en place d'un cloud privé. Dans l'absolu, il faudrait que le déploiement de ces applications soit le plus rapide possible et qu'il puisse être réalisé plusieurs fois (pour divers clients). Pour cette mission, on va mettre en place un open-source GLPI qui permettra aux clients de faire des tickets lorsqu'il y a des problèmes informatiques. Pour cela, il va nous falloir plusieurs outils notamment php (avec modules),mariadb(mysql) et apache cela nous permettra de mettre en place le GLPI pour nos clients. Il nous faudra aussi créer des fichier yml et env pour mettre en place GLPI, nos fichiers on va les créer dans le répertoire stack_glpi, puis on va créer un docker-compose.yml qui compose:

```yaml
---
services:
  mysql:
    image: mysql:latest
    container_name: mysql
    hostname: mysql
    volumes:
      - /var/lib/mysql:/var/lib/mysql
    restart: always
    environment:
      - MYSQL_ROOT_PASSWORD=diouxx
      - MYSQL_DATABASE=glpidb
      - MYSQL_USER=glpi_user
      - MYSQL_PASSWORD=glpi

  glpi:
    image: diouxx/glpi
    container_name: glpi
    hostname: glpi
    ports:
      - "8095:80"
    depends_on:
      - mysql
    volumes:
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
      - /var/www/html/glpi/:/var/www/html/glpi
    environment:
      - TIMEZONE=Europe/Paris
      - MYSQL_HOST=mysql
      - MYSQL_DATABASE=glpidb
      - MYSQL_USER=glpi_user
      - MYSQL_PASSWORD=glpi
    restart: always
```

Le fichier docker-compose se base sur l'image mysql lorsqu'il sera exécuté, on lui attribue un port pour permettre d'accéder à GLPI, il ne faut pas oublier de la mettre comme règle dans l'IPFIRE pour permettre l'accès.

Ensuite en créant un fichier mysql.env, on y met le nom de l'utilisateur avec son mot de passe, ensuite on lui indique le nom de sa base de données et le mot de passe du root.

```
GLPI_MYSQL_ROOT_PASSWORD=password
GLPI_MYSQL_DATABASE=glpidb
GLPI_MYSQL_USER=glpi_user
GLPI_MYSQL_PASSWORD=glpi
```

Ces deux fichiers nous permettront d'obtenir un conteneur et un volume grâce à docker lorsqu'il exécutera la commande docker compose up -d.

```
user@debian:~/stack_glpi$ docker compose up -d
[+] Running 2/2
 ✔Container mysql  Started
                   10.5s
 ✔Container glpi   Started
                   10.8s
```

Il ne reste plus qu'à installer glpi sur le navigateur et à mettre les informations importantes, ainsi les clients pourront accéder à GLPI lorsque l'utilisateur test glpi pourra se connecter.

NEXTCLOUD:

	Gwendoline	Mattéo
identifiant	nextcloud	nextcloud
mot de passe	nextcloud	nextcloud

Adresse: 172.16.142.17X:8091

Nextcloud est un logiciel libre de site d'hébergement de fichiers et une plateforme de collaboration. Le but est le même que GLPI et les manipulations sont exactement pareils, il y a quelques modifications à apporter pour l'exécution des fichiers. Pour le port d'écoute, on lui mettra le port 8091.

```
version: '3'
services:
  nextcloud_db:
    image: mariadb:latest
    container_name: nextcloud_db
    restart: always
    volumes:
      - ./db:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=$NEXTCLOUD_MYSQL_ROOT_PASSWORD
      - MYSQL_DATABASE=$NEXTCLOUD_MYSQL_DATABASE
      - MYSQL_USER=$NEXTCLOUD_MYSQL_USER
      - MYSQL_PASSWORD=$NEXTCLOUD_MYSQL_PASSWORD

  nextcloud_app:
    image: nextcloud:latest
    restart: always
    ports:
      - "8091:80"
    links:
      - nextcloud_db
    volumes:
      - ./data:/var/www/html
    environment:
      - MYSQL_HOST=nextcloud_db
      - MYSQL_DATABASE=$NEXTCLOUD_MYSQL_DATABASE
      - MYSQL_USER=$NEXTCLOUD_MYSQL_USER
      - MYSQL_PASSWORD=$NEXTCLOUD_MYSQL_PASSWORD

volumes:
  db_data: {}
  data_nextcloud: {}
```

```
NEXTCLOUD_MYSQL_DATABASE=nextcloud
NEXTCLOUD_MYSQL_USER=nextcloud
NEXTCLOUD_MYSQL_ROOT_PASSWORD=root_mysql_password
NEXTCLOUD_MYSQL_PASSWORD=nextcloud_password
```

PBS :

Accès à l'environnement Proxmox du PBS : 172.16.142.15:8006

Utilisateur	MDP
root	Adminsio1

Accès au PBS : 172.16.142.25:8007

Utilisateur	MDP
root	adminsio1

Le PBS est entreposé sur un serveur Proxmox a part. Il sert à gérer le stockage, à automatiser les sauvegardes des vm mais aussi à automatiser les purges, il est extrêmement important car ses sauvegardes permettent de prévenir et de résoudre les problèmes de perte de VM et de données.

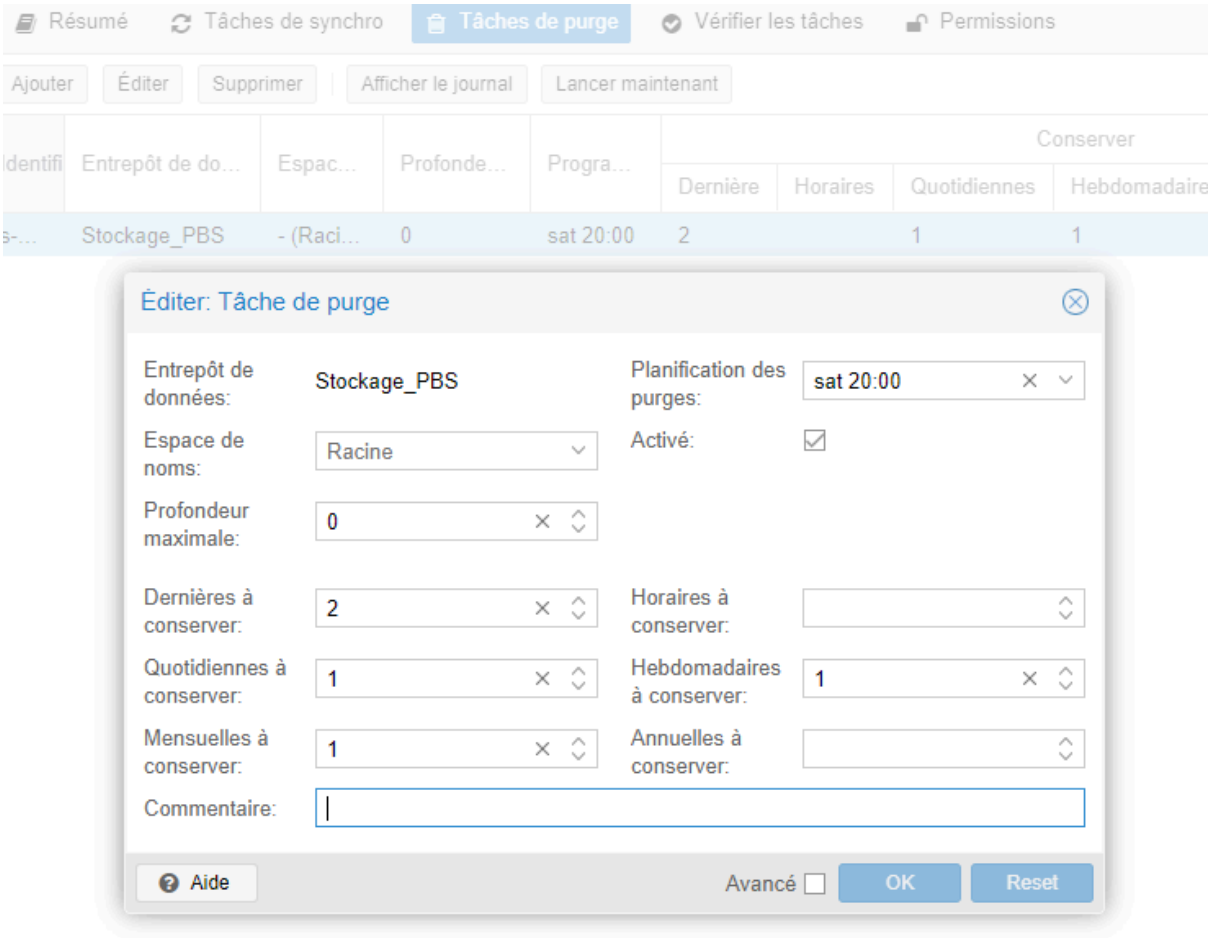
Dans notre cas, nous avons comme matériel une tour (PC) qui nous a servi de serveur pour notre PBS, ainsi qu'un disque ssd de 400 Go qui a servi à entreposer les sauvegardes de nos VM.

Nous avons donc d'abord créé l'espace virtuel du proxmox afin de par la suite créer la VM qui nous servira à accéder à Proxmox Backup serveur (PBS) depuis notre navigateur web.

Par la suite, nous avons automatisé les sauvegardes de nos VM sur le serveur Proxmox principal où se trouvent toutes les VM par rapport à leur importance et aux changements effectués dessus tous les jours.Par exemple, pour les VM avec nos lamp, nous avons établie une sauvegarde tous les jours à 21h00 car les VM lamp sont souvent modifiées et changées, donc il est important de régulièrement faire des sauvegardes. Pour nos VM IPFIRE, nous avons décidé de faire une sauvegarde tous les dimanche à 01h00 car il n'y a quasiment aucune modifications effectuées sur cette VM et, pour finir, pour nos Debian graphique, nous avons décidé de faire des sauvegarde tous les mois car la Debian graphique ne change jamais, elle ne sert qu'au début pour pouvoir accorder l'accès à l'IPFIRE depuis notre navigateur web de notre PC.

✔	Proxmox170	21:00	2024-05-24 21:00:00	pbs	P...	102...
✔	Proxmox170	sun 01:00	2024-05-26 01:00:00	pbs	P...	100...
✔	Proxmox170	monthly	2024-06-01 00:00:00	pbs	P...	220

Nous avons donc automatisé la sauvegarde de toutes nos VM, mais il faut maintenant automatiser la purge pour éviter que les sauvegardes ne s'accumulent et que le disque de stockage soit plein. Cela créerait évidemment un problème car si le disque qui stocke les sauvegardes des VM est plein, il ne pourra alors plus stocker les nouvelles sauvegardes, c'est pour cela qu'il est important d'aussi automatiser les purges des sauvegardes des VM. Pour ce faire, nous avons ajouté une tâche de purge tous les samedi à 20h00, qui nous laisse les 2 dernières sauvegarde de chaque VM, qui nous laisse la dernière sauvegarde quotidienne, la dernière sauvegarde hebdomadaire et la dernière sauvegarde mensuelle.



Nous gardons des sauvegardes hebdomadaires et mensuelles au cas ou il y ai un probleme et que dans le cas d'une entreprise il y ai un probleme quelque qui nécessite une sauvegarde datant de plusieurs semaines ou d'un mois, par exemple en comptabilité où il est possible qu'on en est besoin.

Hardening:

Mission: Votre infrastructure de test est maintenant mise en place mais votre tuteur souhaite vous alerter sur les problèmes de cybersécurité rencontrés de nos jours. Vos machines et vos serveurs sont exposés au web et il est important de protéger vos configurations des attaques externes.

Qu'est ce que c'est:

Le Hardening est une solution qui nous permet de protéger nos données en réduisant les potentiels entrées utilisées par des attaquants.

Solution:

Squid:

Le squid est un serveur proxy, il nous permet de stocker les données consultées sur des pages. Il nous permet aussi de limiter des accès à des sites web comme facebook ou des sites malveillants. On peut le mettre en place sur une debian CLI, en installant le paquet. Dans les fichiers ci-dessous, on peut bloquer l'accès à Youtube, Facebook en mettant une heure où on veut que l'accès soit interdit à telle heure. On peut aussi bloquer des sites en mettant le lien de ce dernier comme école directe. Cela nous permettra de garantir une meilleure sécurité pour empêcher les clients/utilisateurs de cliquer sur n'importe quel lien ou tout simplement limiter les accès aux sites.

```
# ACL de blocage de sites
acl No_Youtube dstdomain .youtube.com
acl No_FacebookFR dstdomain .facebook.fr
acl FacebookFR_planning time M T W T F 12:00-14:00
acl No_FacebookCOM dstdomain .facebook.com

acl deny_domain url_regex -i "/etc/squid/block_domaines.txt"
http_access deny deny_domain

# Refuser l'accès aux sites interdits
http_access deny No_Youtube
http_access deny No_FacebookFR
http_access deny No_FacebookCOM
```

```
tutos-info.fr
ecoledirecte.com
ndlaprovidence.fr
```

Fail2ban:

Fail2ban est un pare-feu de prévention contre les intrusions, il nous permet de sécuriser notre debian LAMP contre les autres utilisateurs ou une personne root.

```
[sshd]
enabled = true
port    = ssh
filter = sshd
logpath = journal
backend = systemd
maxretry = 2
findtime = 300
banaction = iptables-allports
bandtime = 86400
ignoreip = 127.0.0.1
```

PBS:

PBS est un projet de solutions de sauvegarde open source, elle nous permet de sauvegarder nos VMS de notre proxmox principal. Elle nous permet aussi de récupérer d'anciennes sauvegardes vms lorsqu'on souhaite la récupérer pour récupérer un travail qu'on a effectué auparavant. Où alors si la machine a été piratée, il suffirait de supprimer la sauvegarde le jour ou la debian a été piratée et de reprendre la sauvegarde d'avant.

Lynis:

Lynis est un outil de sécurité, il permet d'analyser une machine debian en analysant les faiblesses qui se trouve dans cette dernière et en indiquant des conseils pour mieux sécuriser notre machine. En utilisant la commande ci-dessous, lynis analyse la machine et indique avec des croix rouges ou des * (jaunes) les changements à modifier ou à mettre en oeuvre. Par exemple, dans l'analyse effectuée, on nous indique qu'il faut changer le port ssh (port 22), ce port est très connu et il est utilisé par tout le monde, il vaut donc mieux le changer et prendre un port qui soit après 1024 (ex:1025) et le changer ensuite dans le pare-feu. Après avoir redémarrer ssh l'utilisateur root ne pourra plus se connecter en ssh.

```
root@debian:~/lynis# ./lynis audit system
```

```
Lynis security scan details:

Hardening index : 64 [############        ]
Tests performed : 286
Plugins enabled : 2
```

Avant changement

```
Hardening index : 72 [###############     ]
Tests performed : 264
Plugins enabled : 1
```

Après changement