

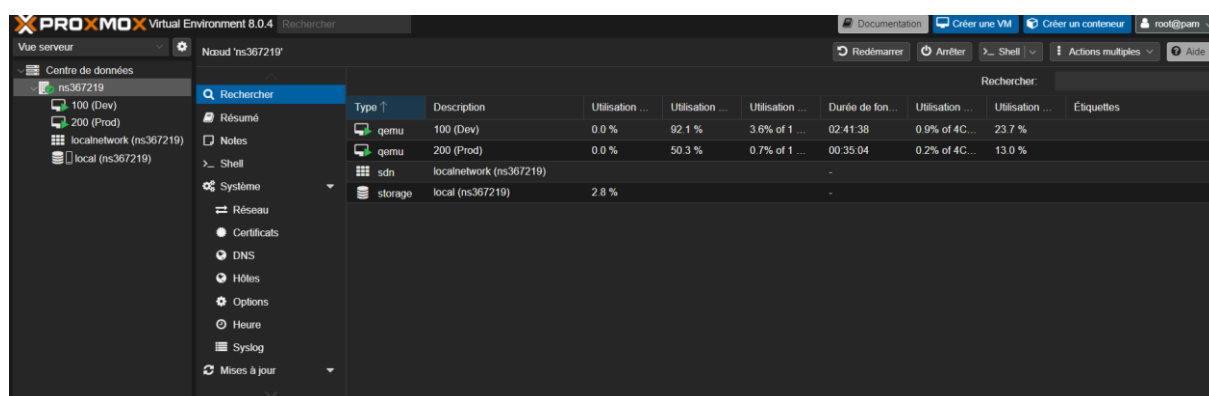
Guide de création des VPS depuis Proxmox



Il s'agit d'une documentation pour créer un seul VPS. A chaque nouveau VPS il faut reprendre la documentation. Elle se veut le plus simplifiée possible

Etape 1 : Configuration de Proxmox

Mettre Proxmox en francais dans les paramètres.



Dans proxmox, importer un iso Ubuntu dans le menu à gauche « **centre de données** » puis « **nom_machine** » puis dans « **local (nom_machine)** », sélectionner « **Images ISO** ».

Créer ensuite une carte réseau pour accueillir le réseau local, pour ce faire aller dans « **nom_machine** » dans le menu à gauche puis « **Système** » puis « **Réseau** », cliquer sur « **Créer** », mettre « **vmbr1** ».

IPv4/CIDR : 192.168.XX.XXX/24 → On peut mettre ce que l'on veut sauf :

Le troisième octet de l'adresse prend ce qu'il veut, exemple : 192.168.**24**.xxx.

Le quatrième octet :

- 192.168.xx.0 : L'adresse réseau (pas utilisée pour adresser des machines individuelles).
- 192.168.xx.255 : L'adresse de diffusion (broadcast) pour ce réseau particulier (pas utilisée pour adresser des machines individuelles).
- 192.168.xx.1 : Souvent utilisée comme l'adresse du routeur ou de la passerelle dans un réseau.

Pour le dernier champ de l'adresse IP, on peut presque mettre ce que l'on veut, au choix pour retenir aisément, mettre 100 donc **192.168.24.100**.

Coché la case « **Démarrage automatique** » puis « **OK** ».

Puis le bouton « **Appliquer la configuration** ».

Ouvrir le shell de Proxmox dans le menu à gauche « **centre de données** » puis « **nom_machine** » et entrer :

```
nano /etc/network/interfaces
```

Puis entrer :

```
auto vmbr1
iface vmbr1 inet static
    address 192.168.24.100 ← Mettre ses propres valeurs
    netmask 255.255.255.0
    bridge_ports none
    bridge_stp off
    bridge_fd 0
    post-up echo 1 > /proc/sys/net/ipv4/ip_forward
    post-up iptables -t nat -A POSTROUTING -s '192.168.24.0/24' -o vmbr0 -j MASQUERADE
    post-down iptables -t nat -D POSTROUTING -s '192.168.24.0/24' -o vmbr0 -j MASQUERADE
```

PS : Mettre ses propres valeurs

Entrer ensuite la commande : `sudo systemctl restart networking`

Etape 2 : Création de la VM sur Proxmox

Dans le menu à gauche « **centre de données** » puis « **nom_machine** », « **Créer une VM** ».

Dans **Général** → Donner un nom à la machine.

Dans **Système d'exploitation** → Mettre l'iso Ubuntu Desktop importer précédemment.

Dans **Système** → Ne rien toucher

Dans **Disques** → Mettre la taille du disque (GiB) selon son besoin.

Dans **Processeur** → Laisser 1 partout ou augmenter le nombre de cœurs si le processeur le permet.

Dans **Mémoire** → Mettre 4096 Mo (4 go de Ram)

Dans **Réseau** → Mettre la carte vmbr1

Puis validé, **lancer la VM et faire l'installation**.

Note : les cartes vmbr0 et vmbr1 sont des interfaces de réseau bridge (ou pont réseau). Elles servent à connecter plusieurs réseaux en les faisant fonctionner comme s'il s'agissait d'un seul réseau. Cette configuration est particulièrement utile dans des environnements virtualisés comme Proxmox, où elle permet aux machines virtuelles (VM) et aux conteneurs d'accéder au réseau externe.

Etape 3 : Mise en place du réseau internet sur la VM Ubuntu

Dans l'interface graphique, paramètre réseau, filaire, rentrer manuellement :

Adresse : 192.168.24.XX

Masque de sous-réseau : 255.255.255.0

Passerelle : 192.168.24.100

DNS : 8.8.8.8,8.8.4.4

Annuler

Filaire



Appliquer

Détails Identité **IPv4** IPv6 Sécurité

Méthode IPv4

☐ Automatique (DHCP)☐ Réseau local seulement
☒ Manuel☐ Désactiver
☐ Partagée avec d'autres ordinateurs

Adresses

Adresse	Masque de réseau	Passerelle	
192.168.25.160	255.255.255.0	192.168.25.100	
			

DNS

Automatique ☒

8.8.8.8, 8.8.4.4

Séparer les adresses IP avec des virgules

Sur Proxmox, clic droit sur la VM → Redémarrer.

La VM devrait être connecté à internet, ouvrir le terminal et taper « 8.8.8.8 » pour tester les DNS Google.

Etape 4: Redirection de port de Proxmox vers une VM Ubuntu pour la connexion SSH

Objectif : Rediriger le trafic entrant d'un port spécifique sur Proxmox vers un port correspondant sur une VM Ubuntu pour pouvoir se connecter en SSH.

Dans le shell Proxmox on va ajouter les règles de connexion entrantes vers un port spécifique, on modifie le port 22 ssh par défaut par soucis de sécurité. Dans l'exemple j'ai mis 22225 mais on peut choisir un autre sur les plages de ports disponibles.

1. Autoriser les connexions entrantes sur un port spécifique :

Configuration à entrer sur Proxmox :

```
sudo iptables -A INPUT -p tcp --dport 22225 -j ACCEPT
```

-A INPUT : ajoute une règle à la chaîne INPUT, qui traite les paquets qui entrent dans le système.

-p tcp : spécifie que cette règle s'applique aux paquets TCP.

--dport 22225 : précise que la règle s'applique aux paquets destinés au port 22225.

-j ACCEPT : indique que les paquets correspondants doivent être acceptés, c'est-à-dire autorisés à entrer dans le système.

2. Autoriser le transfert de paquets vers une adresse IP et un port spécifique :

```
sudo iptables -A FORWARD -p tcp -d 192.168.24.xxx --dport 22225 -j ACCEPT
```

-A FORWARD : ajoute une règle à la chaîne FORWARD, qui traite les paquets qui sont routés à travers le système.

-p tcp : spécifie que cette règle s'applique aux paquets TCP.

-d 192.168.24.xxx : précise que la règle s'applique aux paquets destinés à l'adresse IP 192.168.24.xxx.

--dport 22225 : précise que la règle s'applique aux paquets destinés au port 22225.

-j ACCEPT : indique que les paquets correspondants doivent être acceptés, c'est-à-dire autorisés à être transmis.

3. Rediriger les paquets arrivant sur un port spécifique vers une autre adresse IP et port :

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 22225 -j DNAT --to-destination  
192.168.24.xxx:22225
```

-t nat : spécifie que nous modifions la table NAT (Network Address Translation) d'iptables.

-A PREROUTING : ajoute une règle à la chaîne PREROUTING, qui traite les paquets lorsqu'ils arrivent au système, avant qu'une route ne soit déterminée.

-p tcp : spécifie que cette règle s'applique aux paquets TCP.

--dport 22225 : précise que la règle s'applique aux paquets destinés au port 22225.

-j DNAT : spécifie que les paquets doivent être redirigés.

--to-destination 192.168.24.xxx:22225 : spécifie l'adresse IP et le port vers lesquels les paquets doivent être redirigés.

Ensuite on rend les règles persistantes en installant : `sudo apt-get install iptables-persistent`

Puis : `sudo netfilter-persistent save`

Pour finir : `sudo netfilter-persistent reload`

Configuration de la VM Ubuntu :

Sur la console de la VM depuis Proxmox, ouvrir un terminal et installer OpenSSH avec la commande : `sudo apt-get install openssh-server`

On vérifie que le service est actif avec la commande : `sudo systemctl status ssh`

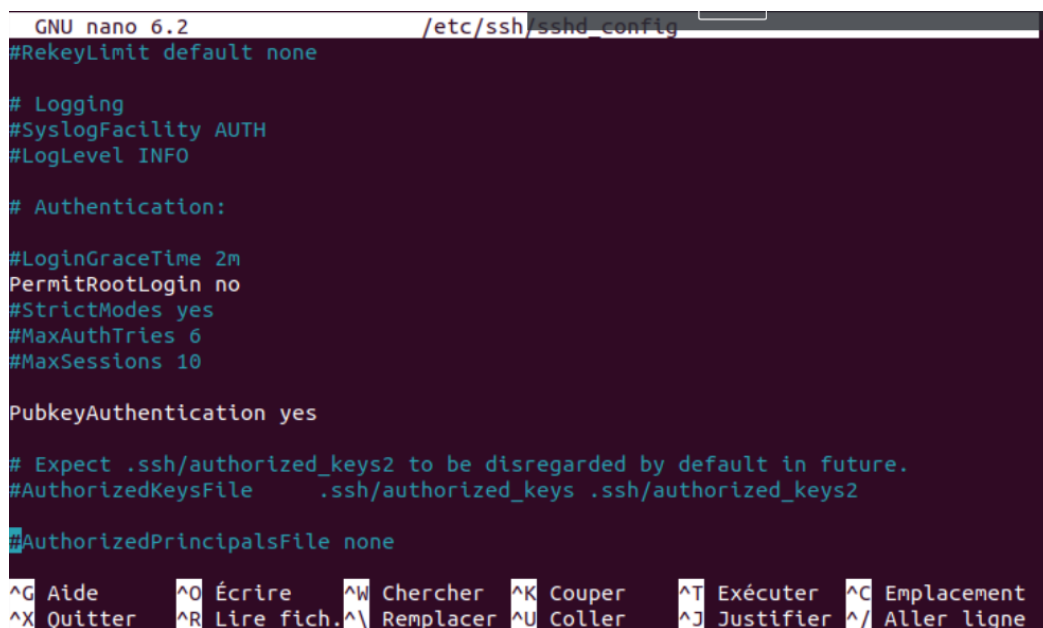
Ensuite on modifie le fichier de configuration SSH : `sudo nano /etc/ssh/sshd_config`

Et on vient décommenter « **Port** » et on modifie 22, on le remplace par 22225 à la place.

Pour finir on décommente « **PermitRootLogin** » et on rajoute « **no** » car on ne veut pas autoriser de connexion « root ».

On enregistre le fichier puis on vient redémarrer le service SSH avec : `sudo systemctl restart ssh`

On peut à présent se connecter sur notre VPS à l'aide de la commande : `ssh -p 22225 user@adresse_ip_serveur`



```
GNU nano 6.2 /etc/ssh/sshd_config
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

^G Aide    ^O Écrire  ^W Chercher ^K Couper   ^T Exécuter ^C Emplacement
^X Quitter ^R Lire fich.^_ Remplacer ^U Coller   ^J Justifier ^_ Aller ligne
```

A venir :

- **Mettre en place des sous domaines avec le nom de domaine park-it.fr pour les différents VPS**
- **Mettre en place un proxy type PFSense pour filtrer les connexions**
- **Mettre en place des connexions SSH uniquement par clé SSH privée**
- **Autorisé tout les futurs ports sur les VPS pour permettre à différents services de fonctionner**
- **Se renseigner quand à l'utilisation d'un Terraform (sûrement inutile pour des VPS en local avec une seul IP de la machine hôte)**