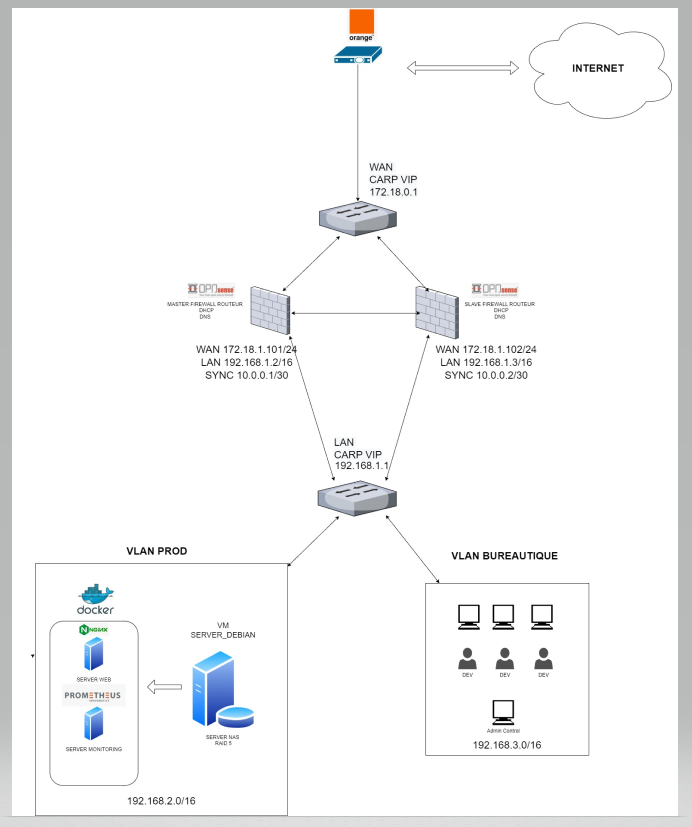
**PROJET ASR | SOFTEAM**

**Création d’une petite infrastructure**

Dans ce projet, nous allons mettre en place une petite infrastructure avec le minimum vital qu’on pourra par la suite proposer/déployer pour une petite société et tout cela à moindre prix :-) (Linux).

https://github.com/Yaou06/Projet-ASR-vini-hieu



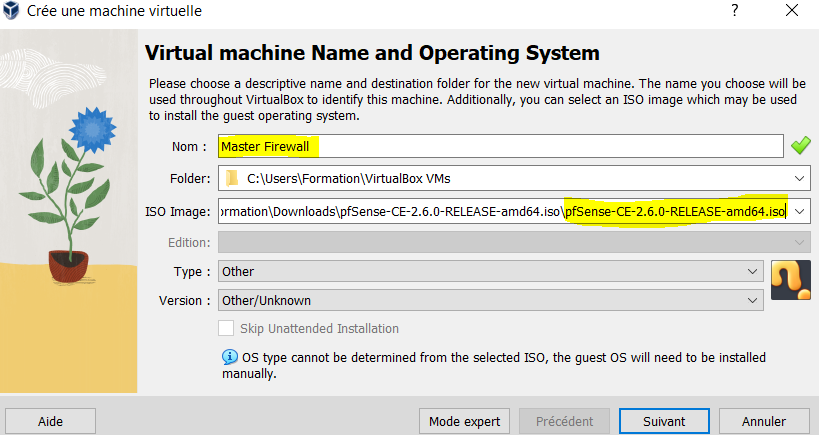
**Étape 1 : Mise en place d’un Firewall avec OPNsense**

a – Télécharger les sources https://opnsense.org/download/

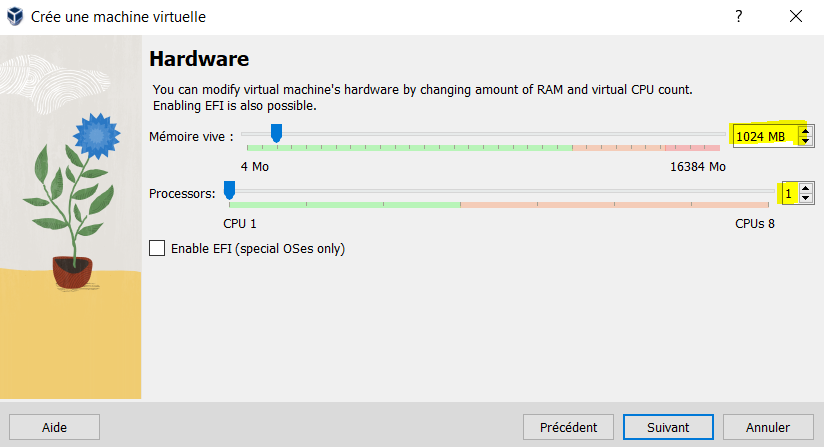
Dans notre exemple, nous avons télécharger la version 64 bits

b – Préparation de la VM avec VirtualBox en vu de l’installation OPNsense

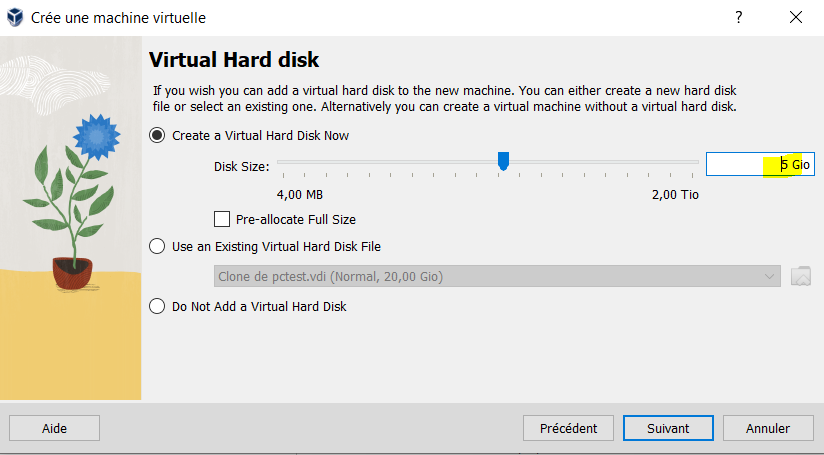
On spécifie un nom, ici « Master Firewall »



On alloue de la mémoire, soit 1 Go Ram



On déclare la taille du disque, soit 5 Go

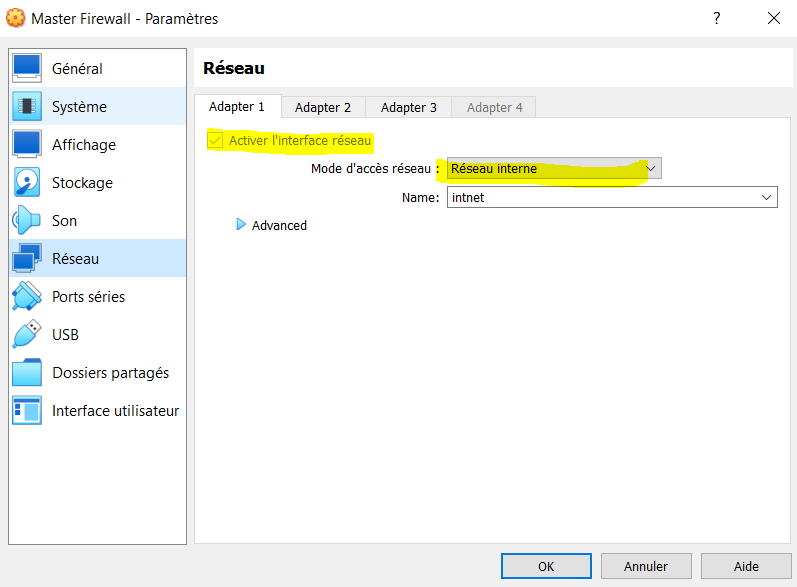


On déclare les trois carte réseaux :

- adapter 1, choisir méthode d accès réseau en « réseau interne » qui fera office de patte LAN

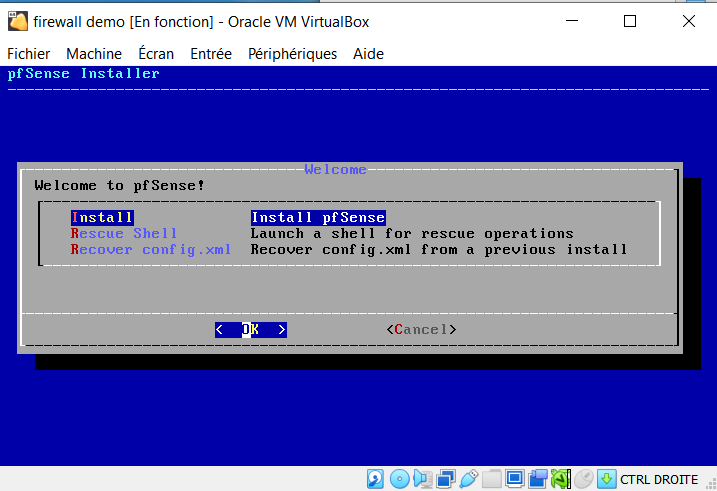
- adapter 2, choisir méthode d accès réseau en « réseau interne » qui fera office de patte de Synchro entre les 2 Firewall

- adapter 3, choisir méthode d accès réseau en « accès par pont » qui fera office de patte WAN

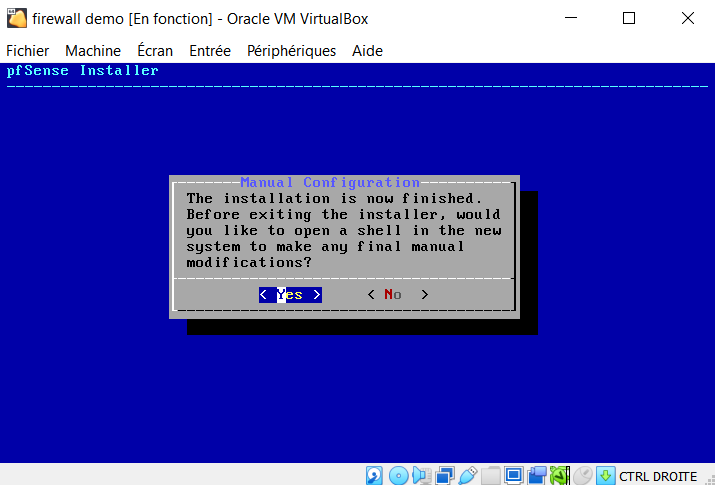


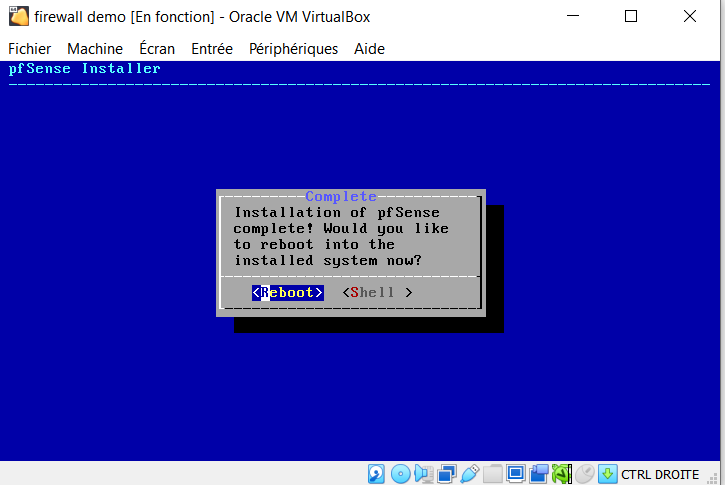
c – Installation OPNsense

Démarrer la VM, la machine devrait booter sur l’iso de l’installation comme ci-dessous



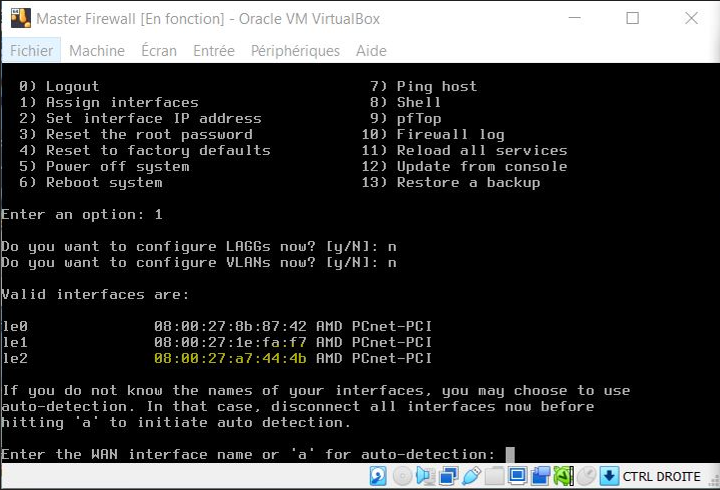
Suivre les instructions d’installation juste qu’a cette capture puis rebooter la vm



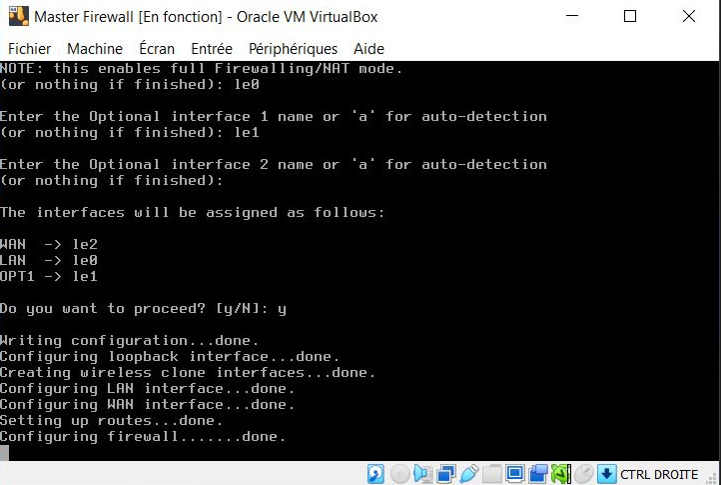


d – Paramétrage OPNsense

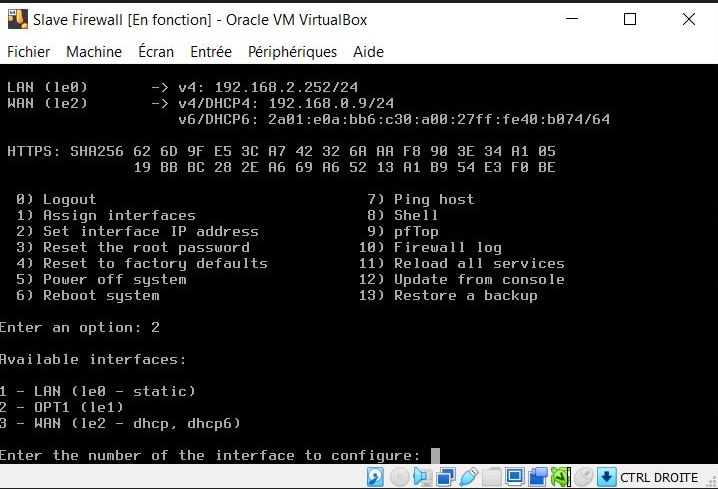
Au redémarrage, la première chose a faire et d assigner les interfaces réseaux. Se connecter en tant que « root » puis choisir l’option 1



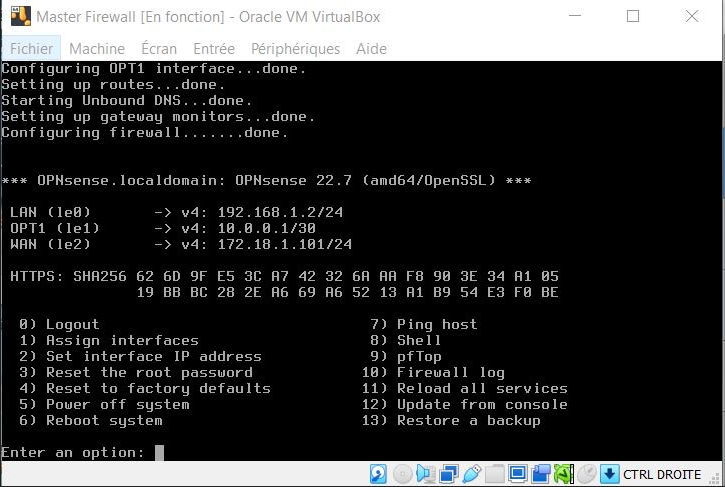
Ce qui donne



Puis on assigne les adresses IP a chaque interface



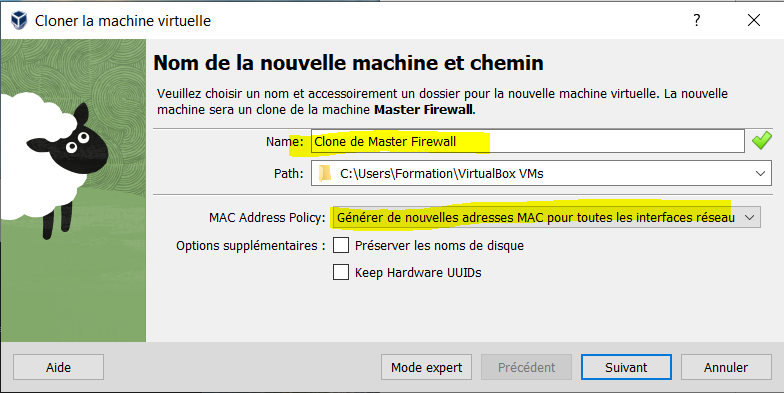
Ce qui donne



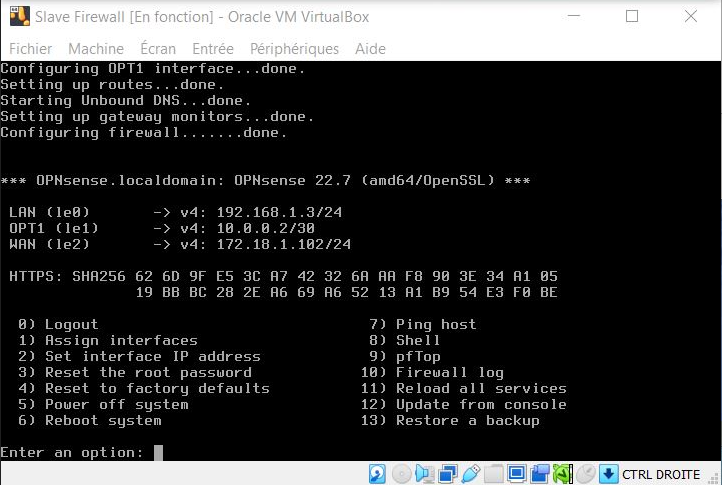
Bravo, vous avez fini le paramétrage du 1er Firewall, qui fera office de Master pour rappel.

Pour le second Firewall qui fera office de Backup, nous allons cloner cette VM par gain de temps.

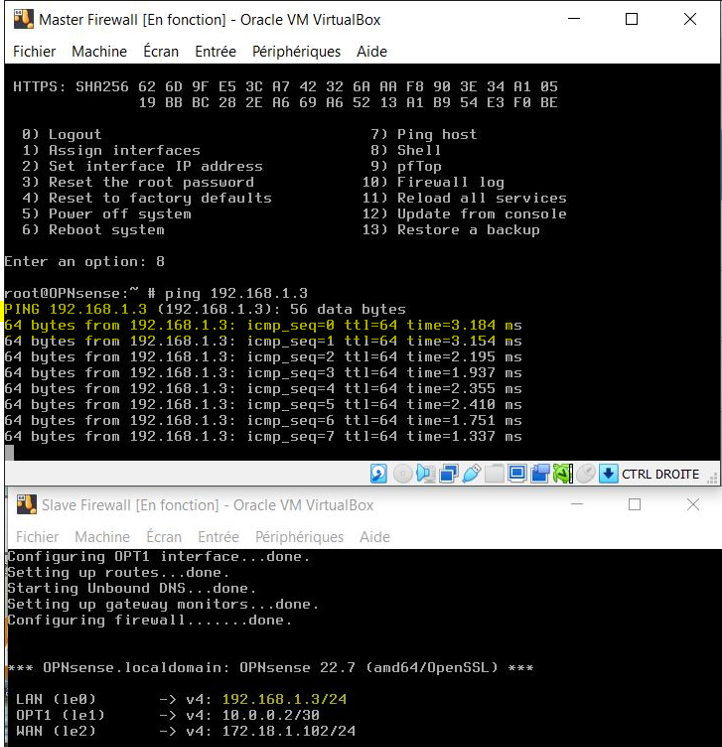
***Attention, lors du clone de Virtualbox, bien spécifié le nouveau nom de machine soit « Slave Firewall » et sélectionner générer des nouvelles adresses Mac.***



Une fois la machine cloner, il nous suffira juste de réassigner les nouvelles IP comme ci-dessous.

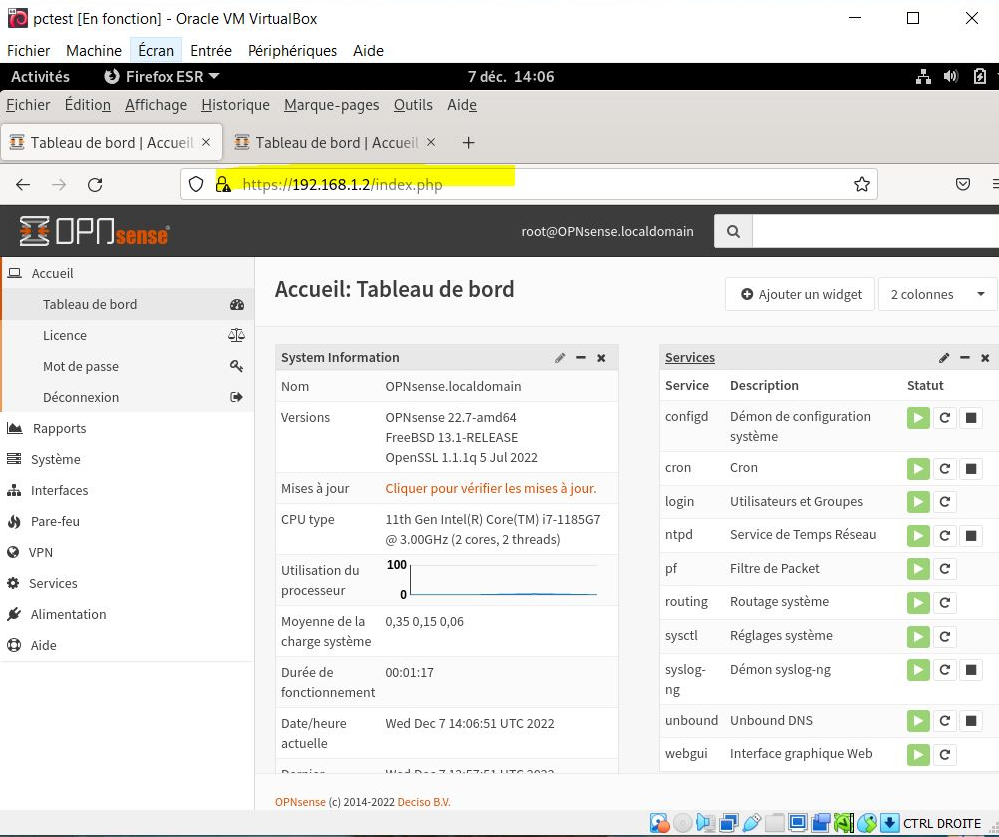


Vérifier que les 2 Firewall communique bien entre eux, via la commande ping



c – Paramétrage OPNsense en interface web

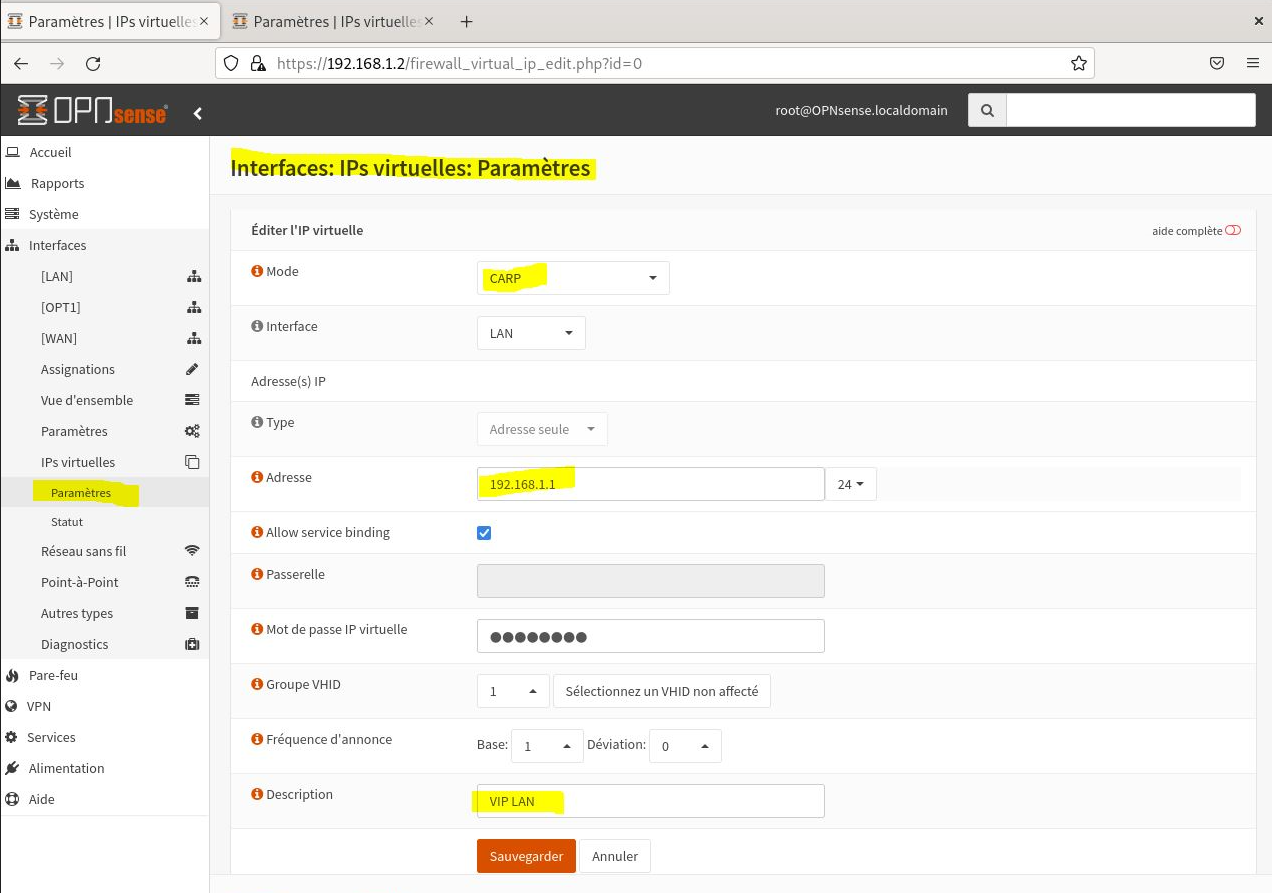
Aller a cette adresse 192.168.1.2 puis entrer l identifiant et le mdp



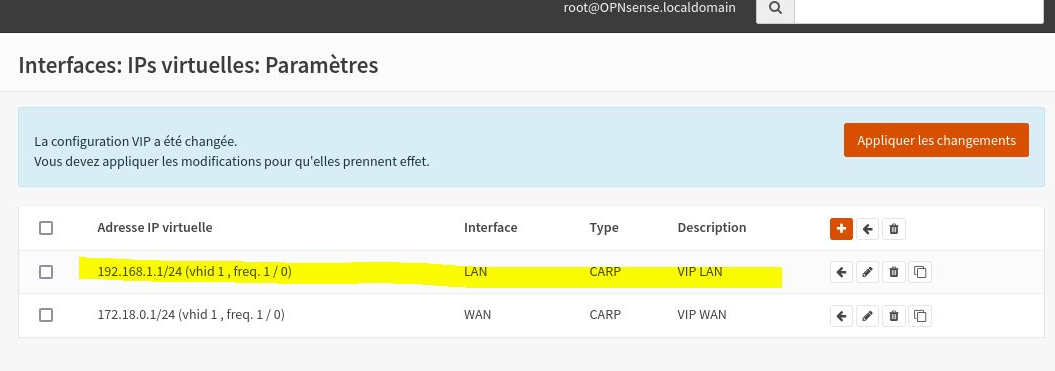
Dans un premier temps, nous allons créer les nœuds « LAN VIP CARP » et « WAN » :

- Aller dans « interfaces » puis « IPS Virtuelles » et enfin « Paramètres »

- Sélectionner le mode « CARP » puis saisir l’ip du LAN VIP CARP soit 192.168.1.1 et enfin donner une description et cliquer sur sauvegarder .

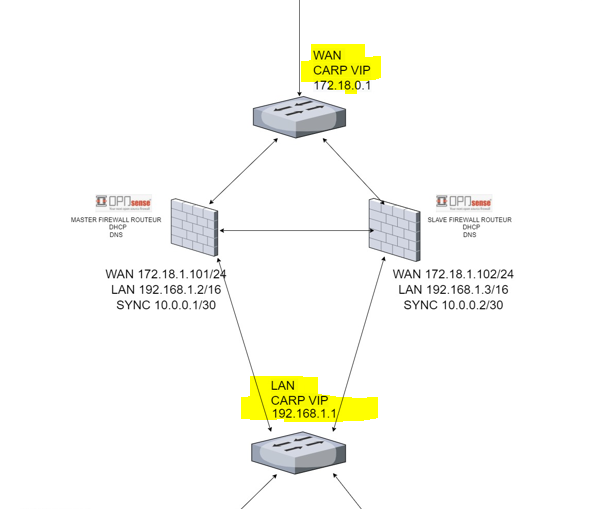


Ce qui donne



- Effectuer la même manip pour le WAN CARP VIP avec l’IP\_172.18.0.1

Pour Rappel :



Dans un second temps, nous allons activer la synchronisation entre les 2 Firewall pour la tolérance de panne

- Aller dans « Système » puis « Haute disponibilité » et enfin choisir « Paramètres »

- Cocher « Synchroniser les états »

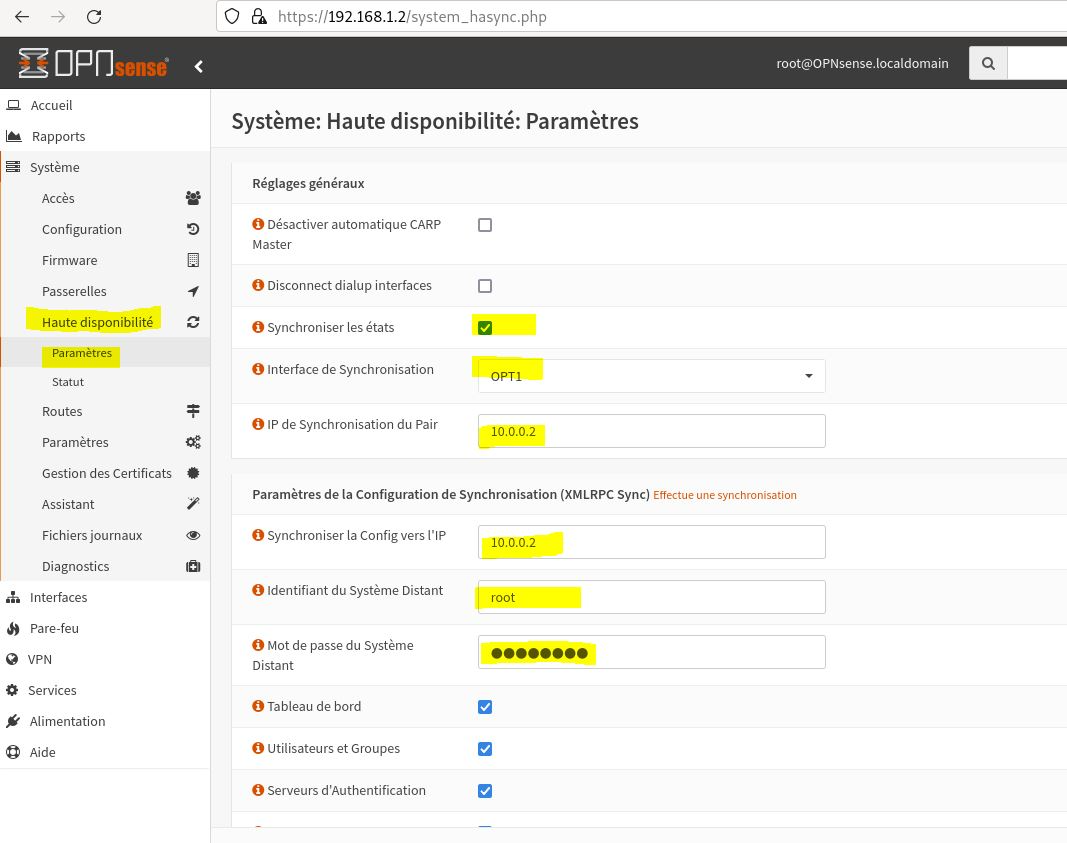
- Sélectionner l’interface OPT1 (patte de synchronisation pour rappel)

- Saisir l’IP de la patte de synchronisation du second Firewall soit 10.0.0.2

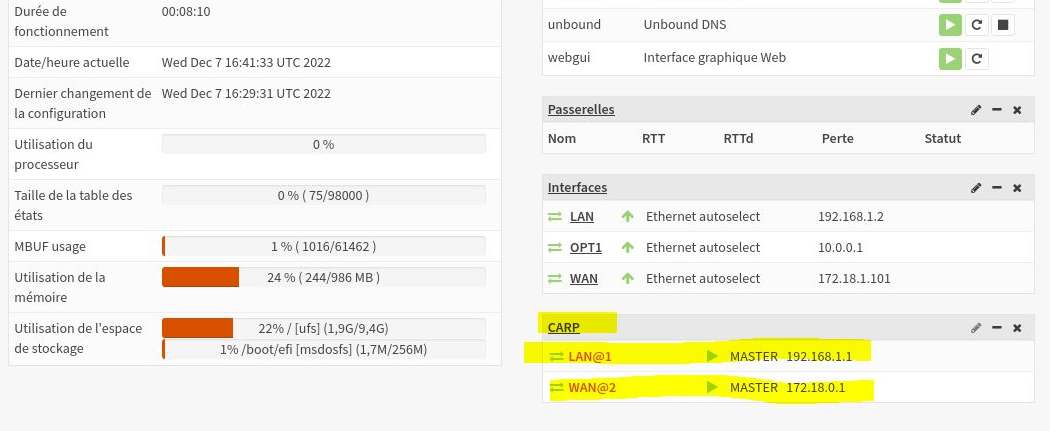
- Saisir l’identifiant et le mdp du compte « root » pour pouvoir accéder au second Firewall

- et enfin, cocher les différentes options que vous souhaitez synchroniser. Ici, nous avons coché toutes les options

Ce qui donne

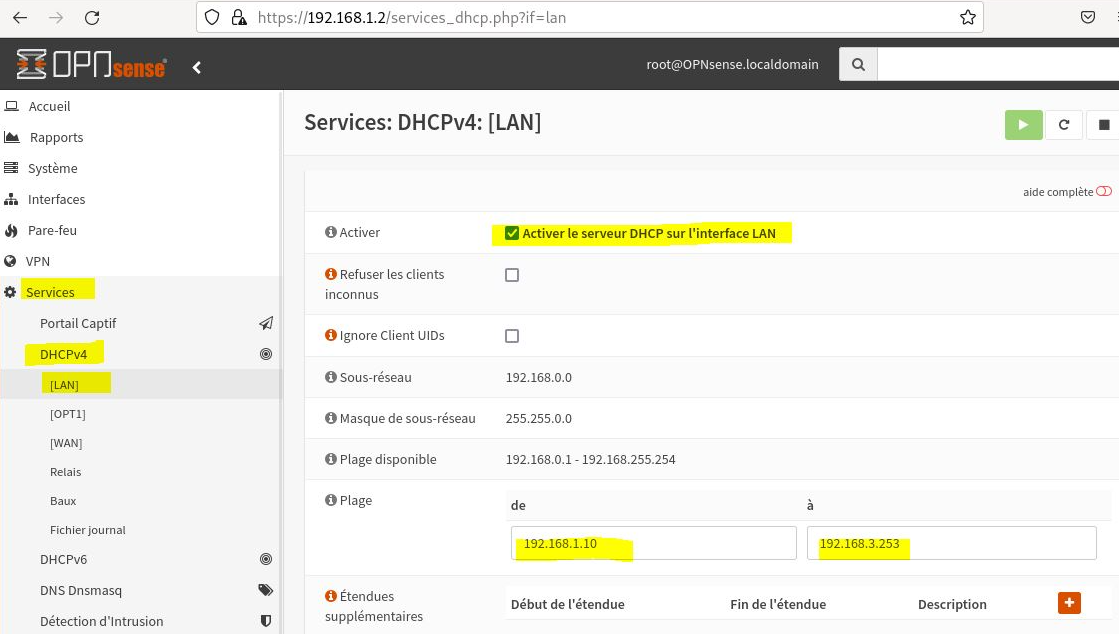


Le résultat, depuis le tableau de bord



***Attention : Toutes ces manips sont a reproduire sur le second Firewall***

Dans un troisième temps, activation du dhcp sur le LAN

- Aller dans « Services » puis « DHCPv4 » et enfin choisir « LAN »

- Cocher « Activer le serveur DHCP »

- Puis définir la plage IP, ici 192.168.1.10 jusqu’à 192.168.3.253

**BRAVO !!! Nous en avons fini avec la partie réseau :-)**

**Étape 2 : Mise en place d’un serveur Debian**

a – Télécharger les sources https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/

b – Préparation de la VM avec VirtualBox en vu de l’installation Debian 11

Se referer à la precedente installation OPNsense qui differe peu, ici je donne juste la configuration souhaitée:

- Nom: pctest

- Ram: 4 Go

- CPU: 2

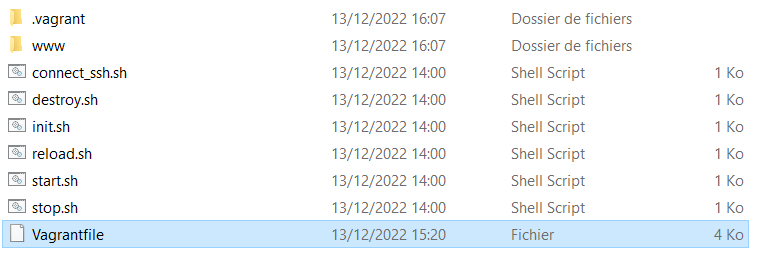
- Stockage: 20 Go

- Carte Reseau: 1 seule et choisir en mode d'acces "Reseau interne"

**Étape 3 : deploiement Ngnix avec vagrant depuis le serveur Debian**

a – Configuration du fichier Vagrantfile

C'est à partir de ce fichier que l'on declare ce que l'on souhaite deployer



Ouvrir le fichier à partir d'un bloc-notes et inserer ce texte :

Vagrant.configure("2") do |config|

config.vm.box = "debian/bullseye64"

config.vm.network "forwarded\_port", guest: 80, host: 8085

config.vm.network "private\_network", type: "dhcp", virtualbox\_\_intnet: true

config.vm.synced\_folder "./www", "/var/www/html"

config.vm.provision "shell", inline: <<-SHELL

apt update

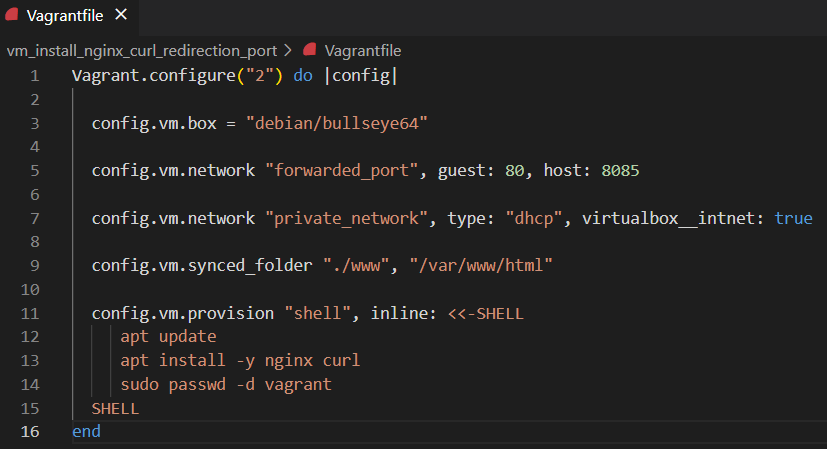
apt install -y nginx curl

sudo passwd -d vagrant

SHELL

end

Ce qui donne



En detail:

- config.vm.box = "debian/bullseye64"

>>> ici, il s'agit de la version 11 de debian à installer dans la VM

- config.vm.network "forwarded\_port", guest: 80, host: 8085

>>> ici, il s'agit de la redirection du port invité vers la machine hote

- config.vm.network "private\_network", type: "dhcp", virtualbox\_\_intnet: true

>>> ici, il s'agit de declarer une seconde carte reseau "en interne" dans virtualbox et de recuperer une IP via notre DHCP

- config.vm.synced\_folder "./www", "/var/www/html"

>>> ici, il s'agit de declarer les dossier correspondant au serveur web Ngnix

- config.vm.provision "shell", inline: <<-SHELL

apt update

apt install -y nginx curl

sudo passwd -d vagrant

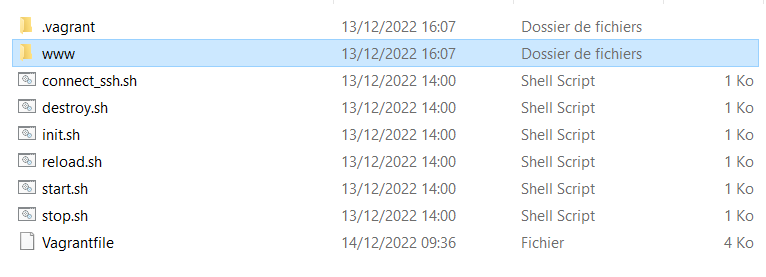
SHELL

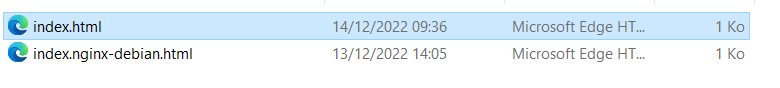
end

>>> ici, on souhaite installer nginx et curl. De plus, on souhaite retirer le mdp au compte: vagrant

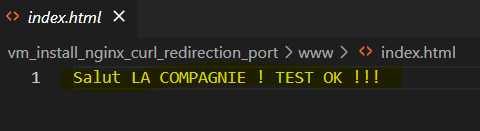
b – Configuration du fichier Index.html

C'est à partir de ce fichier que l'on declare ce que l'on souhaite voir apparaitre sur notre site web

Aller dans le repertoire "www" puis modifier le fichier Index.html

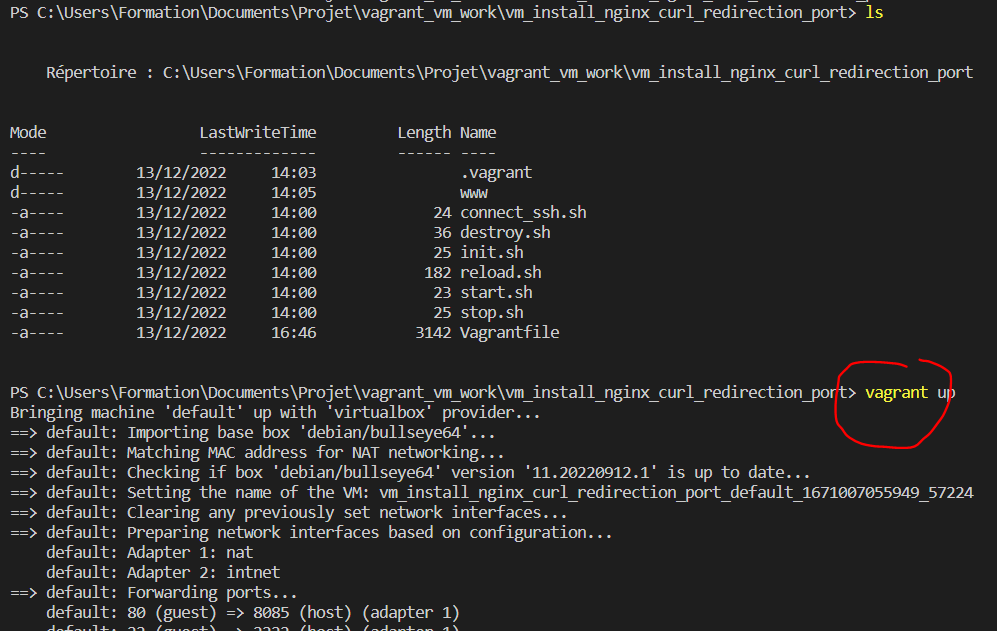


Dans notre cas, nous avons ajouté le texte suivant:

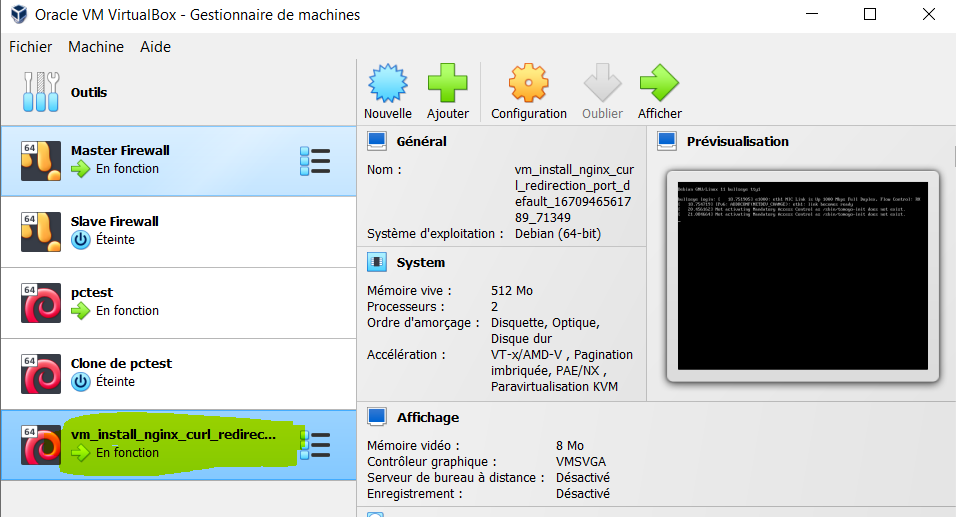


c – Deploiement de la VM via Vagrant

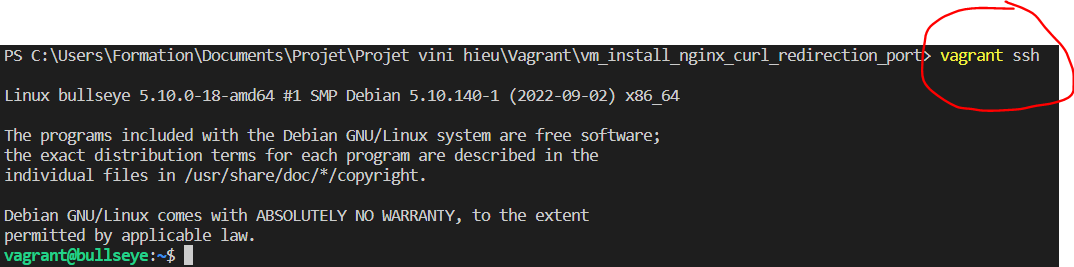
Depuis notre dossier contenant vagrant, lancer la commande "vagrant up" depuis un powershell



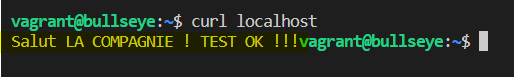
Si tout est OK, dans virtual box notre VM devrait apparaitre comme ci-dessous



Puis se connecter en ssh à notre vm via la commande "vagrant ssh"



Et enfin testons notre site web via la commande "curl localhost"



**BRAVO !!! Notre site Web est fonctionnel ;-)**