Installation et parametrage des serveurs web et base de donnÉes

WhatSup au service de la performance

Qwirk

GESTION DU DOCUMENT

Le tableau ci-après trace l’évolution du document depuis sa création.

| **Version** | **Date** | **Auteur** | **Nature des modifications** |
| --- | --- | --- | --- |
| 0.1 |  |  | Document initial - draft |
| 0.2 |  |  | Mise à jours et rajouts |
| 0.3 |  |  | Révision |
| 0.4 |  |  | Corrections |
| 0.6 |  |  | Changement stratégie de backup |
| 0.7 |  |  | Corrections |
| 0.8 |  |  |  |
| 0.9 |  |  |  |
| 0.95 |  |  |  |
| 0.96 |  |  |  |
| 0.97 |  |  |  |
| 0.98 |  |  |  |

Table des matières

[1. But du document 2](#_Toc484619157)

[2. Descriptif de l'installation 3](#_Toc484619158)

[2.1. Installation des serveurs de virtualisation 3](#_Toc484619159)

[2.1.1. ESXi (site 1) 3](#_Toc484619160)

[2.2. Configuration pfSense 5](#_Toc484619161)

[2.2.1. Réseau principal 5](#_Toc484619162)

[i. pf-sense 192.168.10.252 5](#_Toc484619163)

[ii. pf-sense 192.168.10.253 10](#_Toc484619164)

[II. Installation et configuration du cluster mysql 17](#_Toc484619165)

[1.1. Installation d’une distribution GNU/Linux 17](#_Toc484619166)

[Maintenant nous allons procédé à l'installation des machines virtuelles. 20](#_Toc484619167)

[1.2 Installation du cluster mysql //pierre 42](#_Toc484619168)

[1.2. Installation et configuration des serveurs WSGI & Nginx 45](#_Toc484619169)

[Reverse proxy Nginx : Load-balancing 50](#_Toc484619170)

[I. Présentation 50](#_Toc484619171)

[II. Reverse proxy, Load-Balancing 50](#_Toc484619172)

[III. Présentation de l'architecture Nginx 50](#_Toc484619173)

[IV. Installation & configuration d'Nginx en reverse Proxy 50](#_Toc484619174)

[V. Tests 50](#_Toc484619175)

[VI. Ajustement des paramètres 50](#_Toc484619176)

[6. Descriptif sauvegarde 50](#_Toc484619177)

[6.1. Stratégie 50](#_Toc484619178)

[6.2. Azure backup 51](#_Toc484619179)

[7. Descriptif sécurité 51](#_Toc484619180)

[7.1. Configuration du HTTPS 51](#_Toc484619181)

[Annexe 1 : comptes administratifs 51](#_Toc484619182)

# But du document

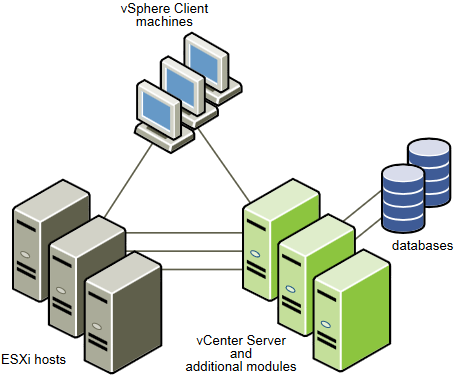
Ce document a pour but de présenter les étapes réalisées lors de l’installation des serveurs d’hébergement de votre domaine, web sur les datacenters de New York, Dallas, et Tokio.

# Descriptif de l'installation

*Cette section présente les étapes d’installation des différents logiciels*

## Installation des serveurs de virtualisation

*Ce chapitre présente la mise en place des serveurs de virtualisation.*



### ESXi (site 1)

**Réaliser une clé bootable hébergeant ESXi :**

* Téléchargez l’image .iso de l’ESXI sur le site de VMware. L’image pèse approximativement 350Mo.
* Ouvrir le fichier esxi.iso et ouvrir l’archive image.tgz.
* Dans  usr\lib\vmware\installer extraire le fichier VMware-VMvisor-Installer-201601001-3380124.x86\_64.bz2. Le résultat de l’extraction est un fichier au format .dd de 900 Mo.
* Utiliser l’utilitaire DD, qui se présente comme un fichier .exe (téléchargé sous la forme d’une archive à extraire). Ouvrir un invite de commandes, se rendre à l’emplacement où se trouve l’extraction de l’archive de l’utilitaire DD.
* Brancher la clé USB
* Utiliser la commande suivante pour lister les périphériques : **dd – list**
* Entrer la commande :

**dd bs=1M if=”c:\Documents and Settings\Administrateur\Bureau\usr\lib\vmware\installer\VMware-VMvisor-big-208167-x86\_64.dd ” of=\\?\Device\Harddisk1\DP(1)0-0+6**

Explications :

dd : indique la commande

bs=1M : taille des blocs à écrire

if= « … » : chemin du fichier extrait

of= « … » : destination à écrire (entrer la destination issue de la précédente commande dd –list).

**Démarrage sur la clé :**

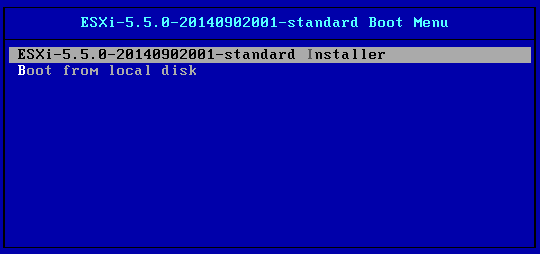
* Se rendre sur la machine à démarrer et y insérer la clé. Démarrer la machine et sélectionner un périphérique de démarrage temporaire.

NOTE : au démarrage, le choix d’un périphérique sur lequel booter s’affiche sur l’appui d’une touche, se reporter à la documentation du matériel.

* Sélectionner la clé USB comme périphérique de boot.
* ESXi démarre et charge ses différents modules.
* Retirer la clé USB.

**Configuration de l’ESXi :**

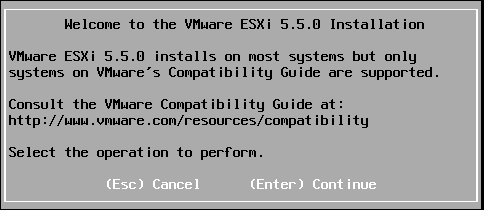
Une fois démaré le nous devons lancer l'installation dites 'standart' du systéme esxi pour ce faire appuyer simplement sur la touche 'entré'.



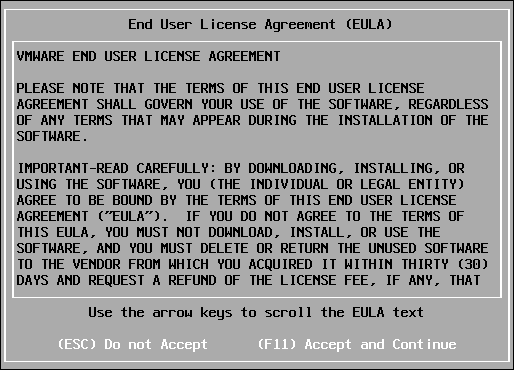
Les services installé sur la clé boutable se mettrons en marche afin d'effectuer l'installation du serveur.



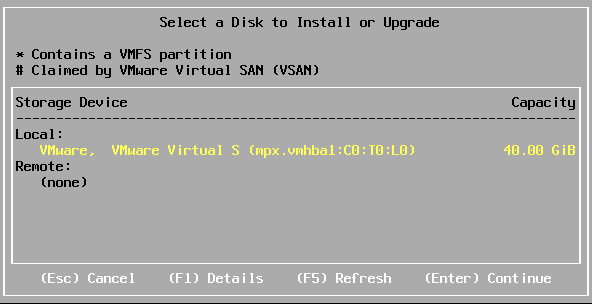
Une fois les services démarés nous allons rentrés dans le cœur du sujet c’est-à-dire l'installation du systéme.



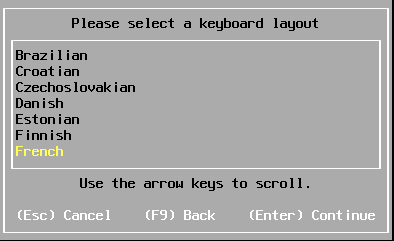
Appuyez sur votre toucher entré afin de confirmer l'installation puis F11 pour accepter la license.



Le service d'installation va détecté automatiquement le matériel pour afficher les différentes partitions pour le stockage de l'os dans notre cas nous avons une partition de 40 go ce qui est peut mais suffisant pour les besoins de notre projet.



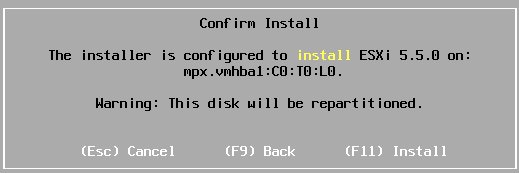
Une fois votre disque sélectionné appuyez sur la touche 'Entré' puis sélectionnez le type de clavier relié à votre machine dans notre cas nous utilisons une dispositions française.



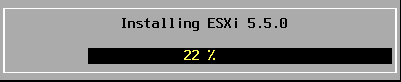
Puis validez votre choix en appuyant sur la touche 'entré', nous devons maintenant procédé à l'affectation du mot de passe pour votre serveur esxi, nous vous recommandons d'utiliser le code Supinf0 puis de valider avec la touche 'entré'.



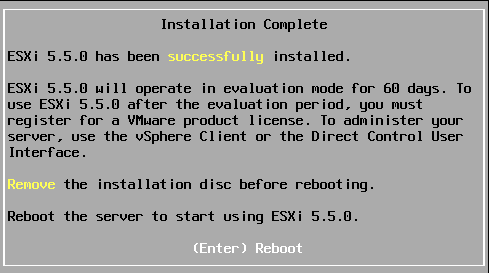
Une fois ceci nous devons confirmer notre installation en appuyant sur la touche F11.



Nous pouvons observer une barre de progression de l'état d'avancement du serveur.



Une fois ceci fait il suffit d'appuyer la toucher "entré" pour effectuer un redémarage de la machine.



Une fois la machine démarrer appuyez sur la touche F2 affin de pouvoir accéder à la configuration de la machine, vous serez inviter à vous identifiez à l'aide de vos identifiants saisis durant l'installation, puis re appuyer sur F2.

Sélectionner la ligne **Configure Management Network puis** la ligne **IPv4 Configuration**

* Taper l'adresse IP, le masque de réseau et la passerelle des tableaux ci-dessous pour tout les serveurs esxi. Valider avec la touche **[Entrée]**.
* Sélectionner la ligne **Restart Management Network**.

|  |  |  |  |
| --- | --- | --- | --- |
| Nom des serveurs | Ny-esxi | Tx-esxi | To-esxi |
| Adresse ip | 192.168.10.20 | 192.168.10.30 | 192.168.10.40 |
| Masque sous réseaux | 255.255.255.0 | 255.255.255.0 | 255.255.255.0 |
| Passerelle | 192.168.10.254 | 192.168.10.254 | 192.168.10.254 |

Les hyperviseurs VMware ESXi sont opérationnel.

## Configuration pfSense

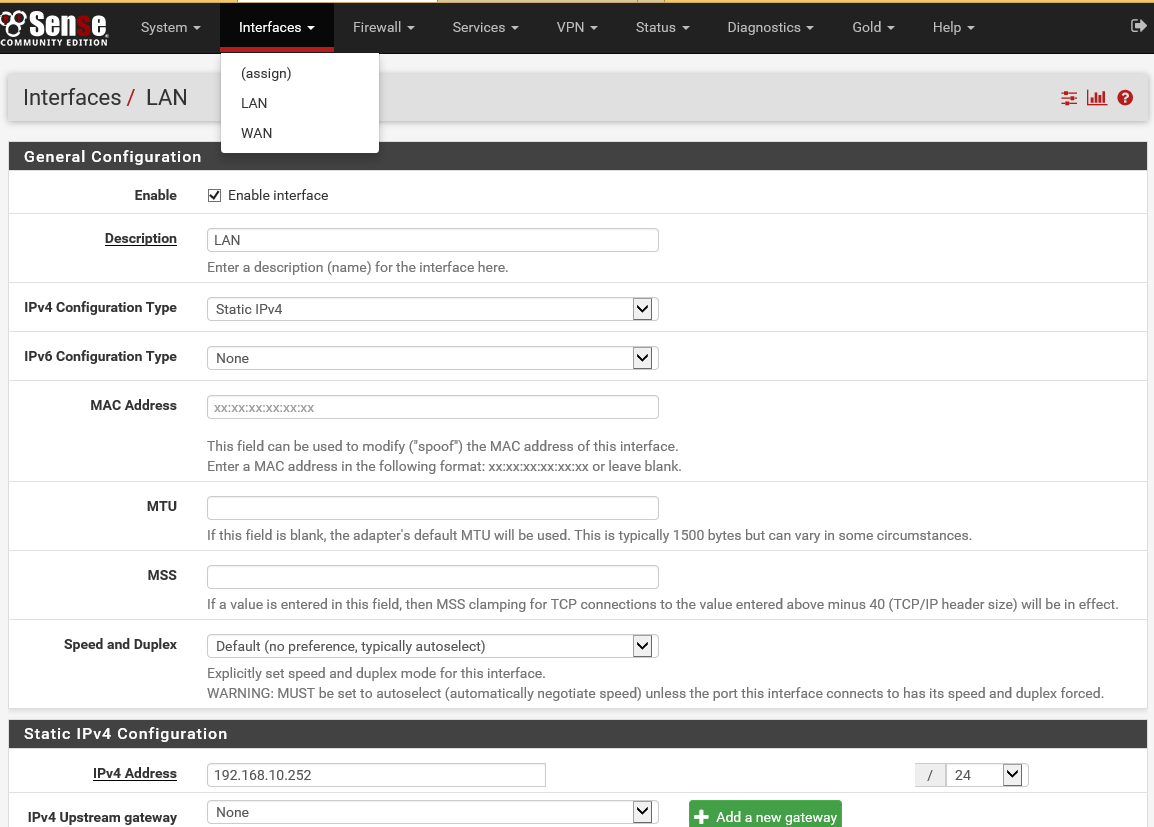
*Ce chapitre présente la configuration de pfSense.*

**pfSense** est un routeur/pare-feu open source basé sur le système d'exploitation FreeBSD

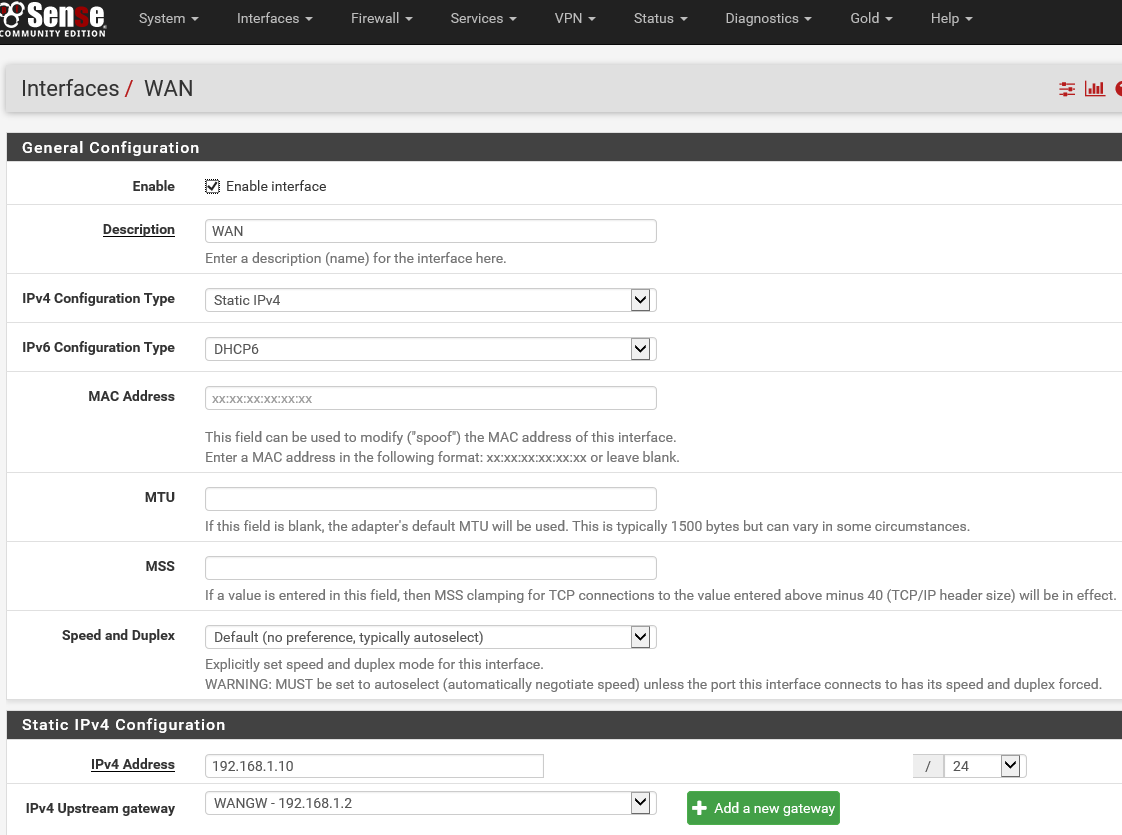
## Réseau principal

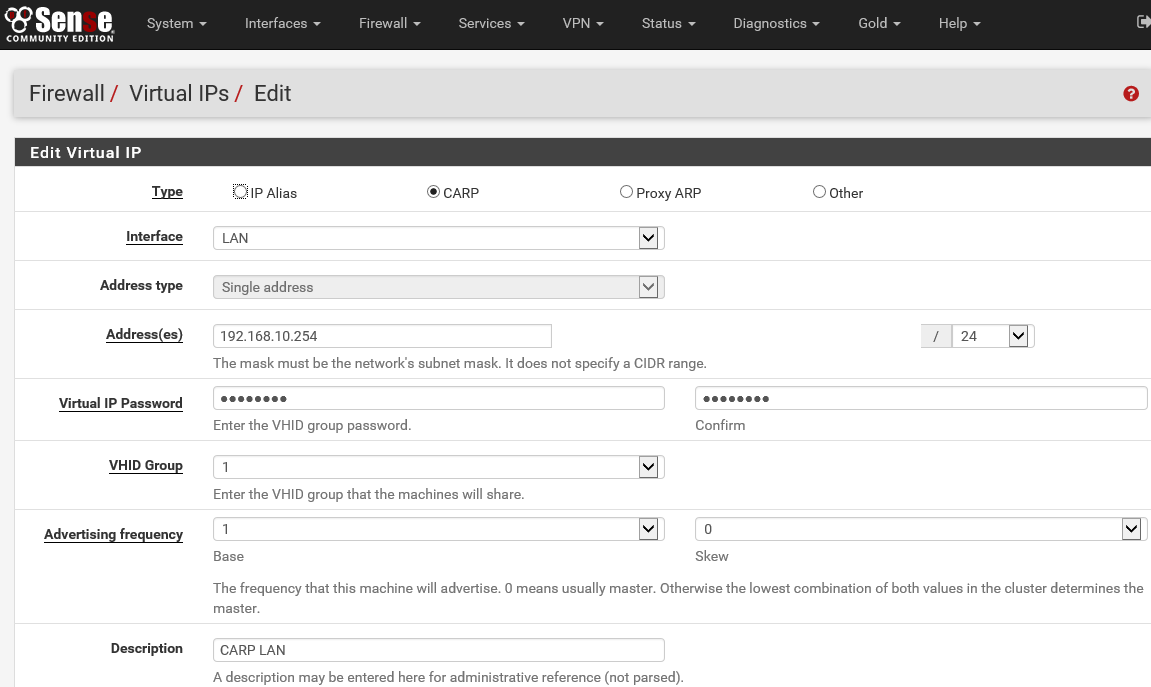
## pfSense 192.168.10.252

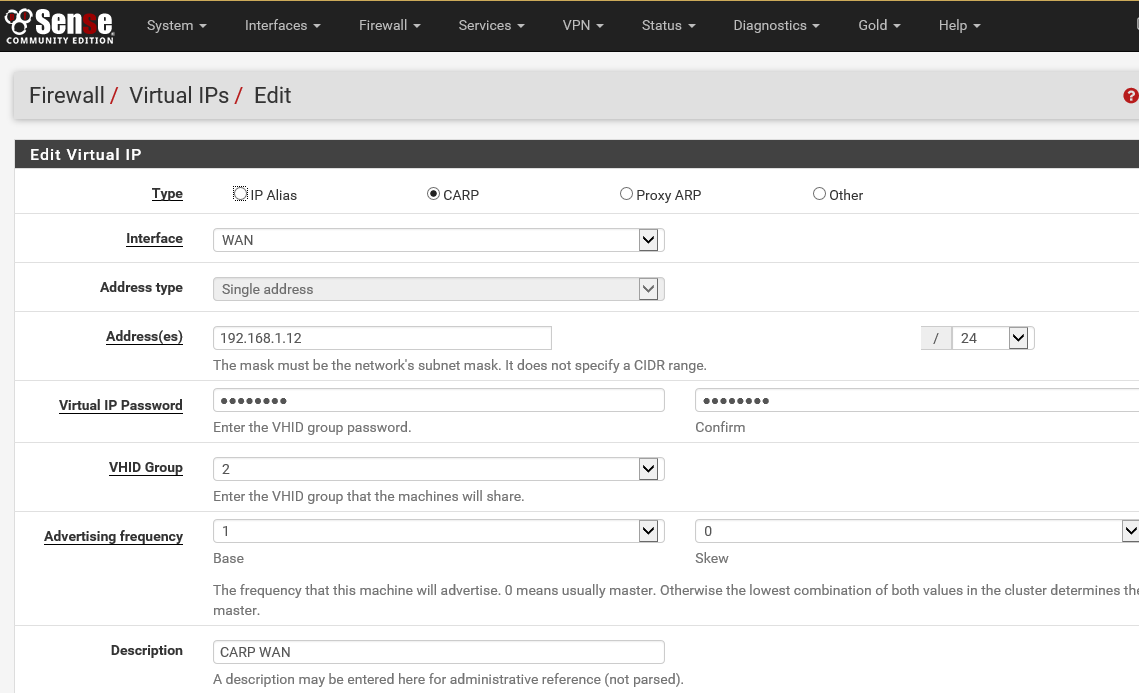
1) Configurez l'interface LAN au champ « IPv4 Address », sans oublier de l'activer en cliquant sur « Enable interface »



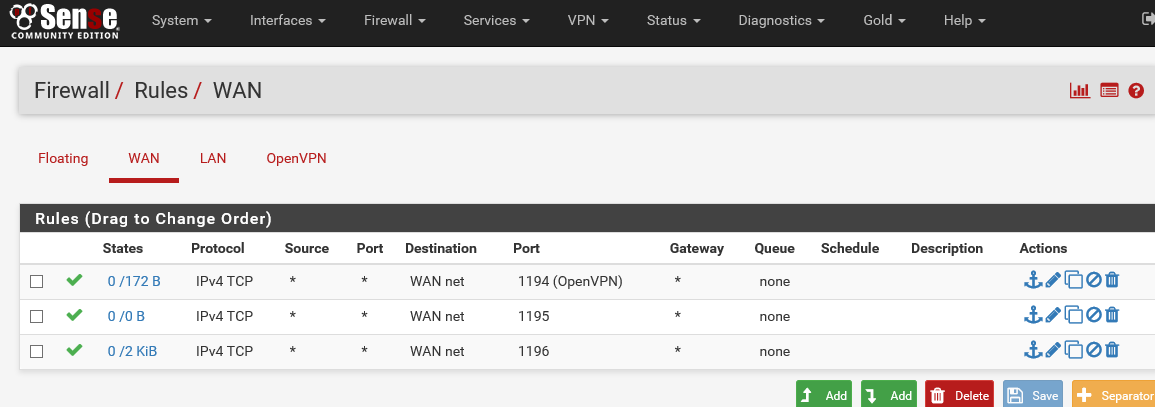
2) Configurez l'interface WAN au champ « IPv4 Address », sans oublier de l'activer en cliquant sur 'Enable interface'

3) Configurez une IP virtuelle avec le CARP, pour le LAN

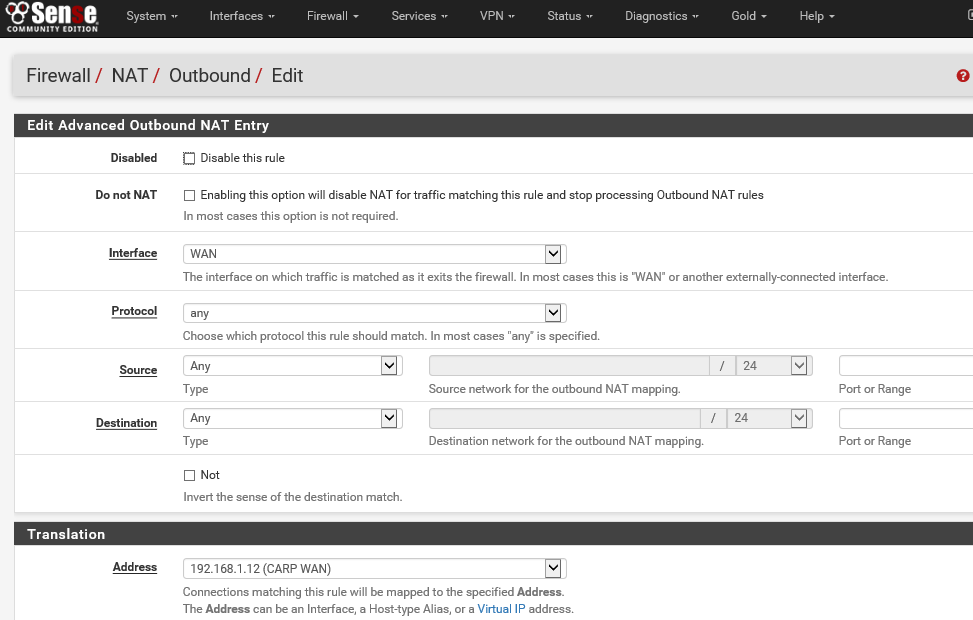
4) Configurez une IP virtuelle avec le CARP, pour le WAN



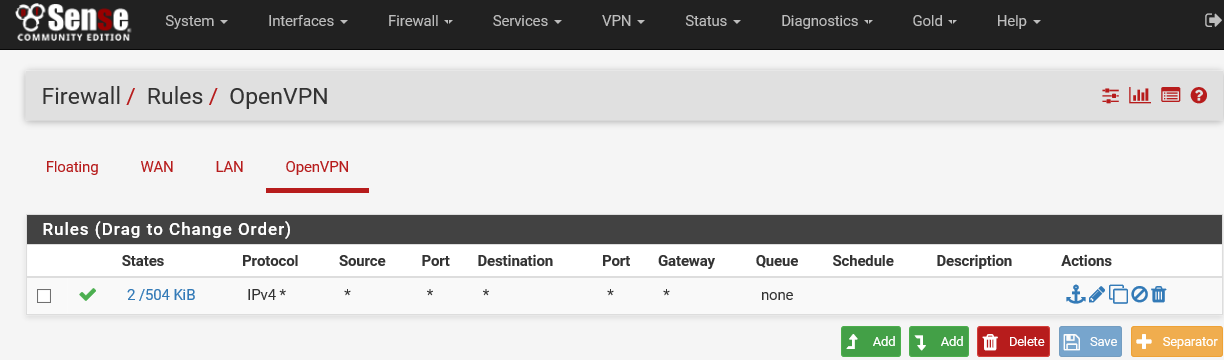
5) Configurez les règles concernant le WAN. Ici, 3 ports sont ouverts pour chaques sites.



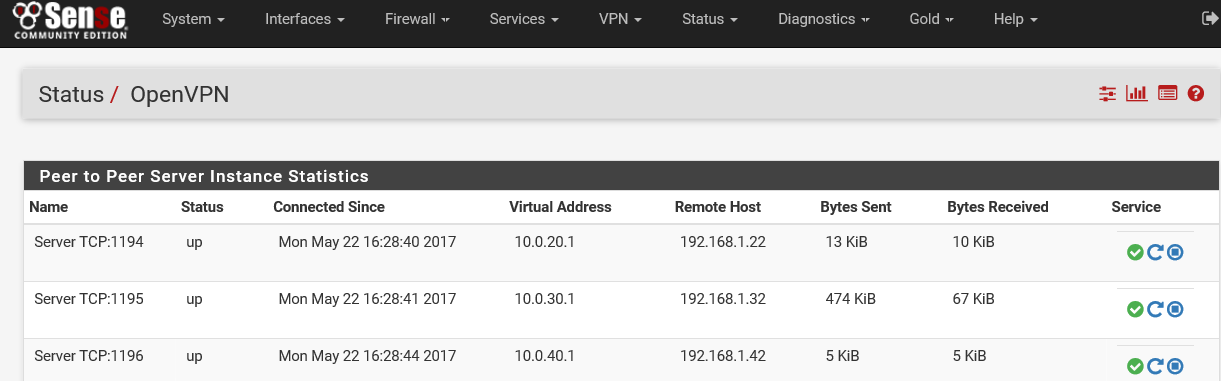
6) pfSense utilisera l'adresse IP virtuelle du cluster pour les communications externes.



7) On autorise tout type de protocole via OpenVPN.

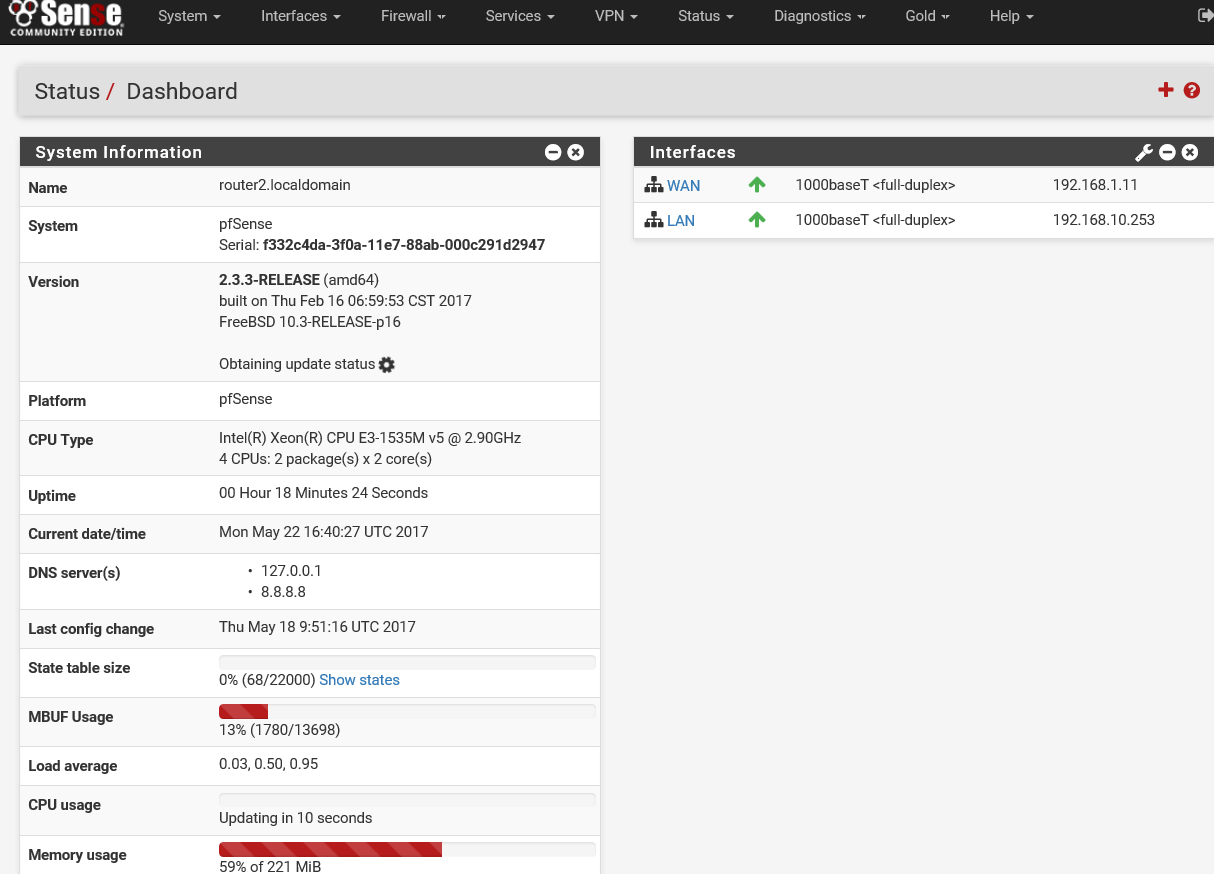


8) Nous pouvons remarquer que tout nos liens sont fonctionnels.

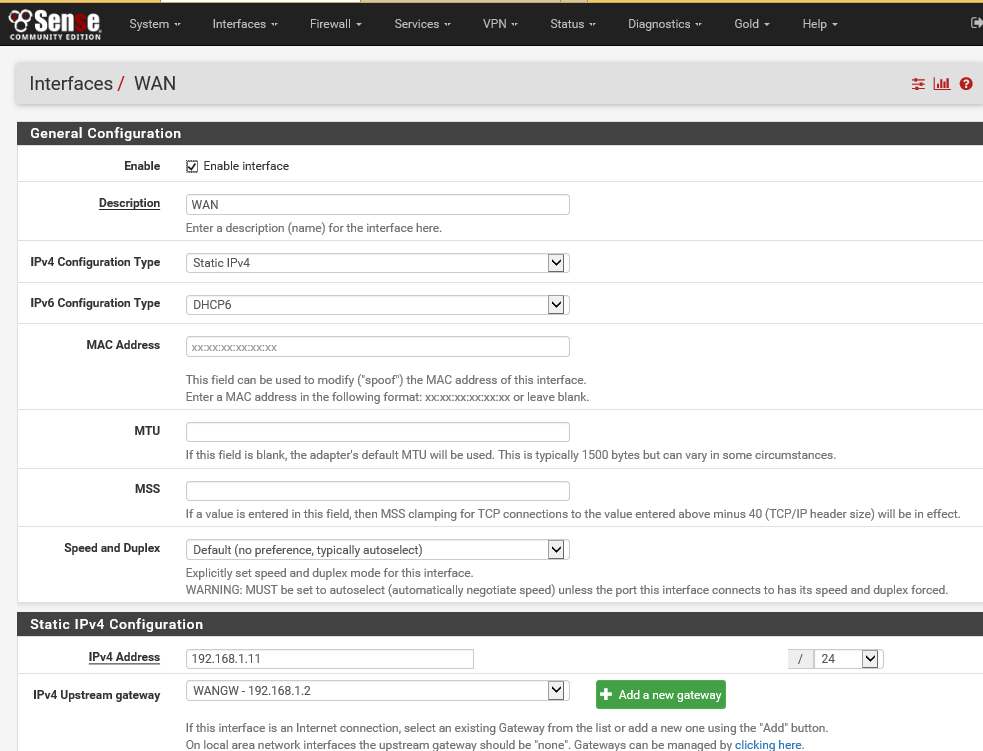


## pfSense 192.168.10.253

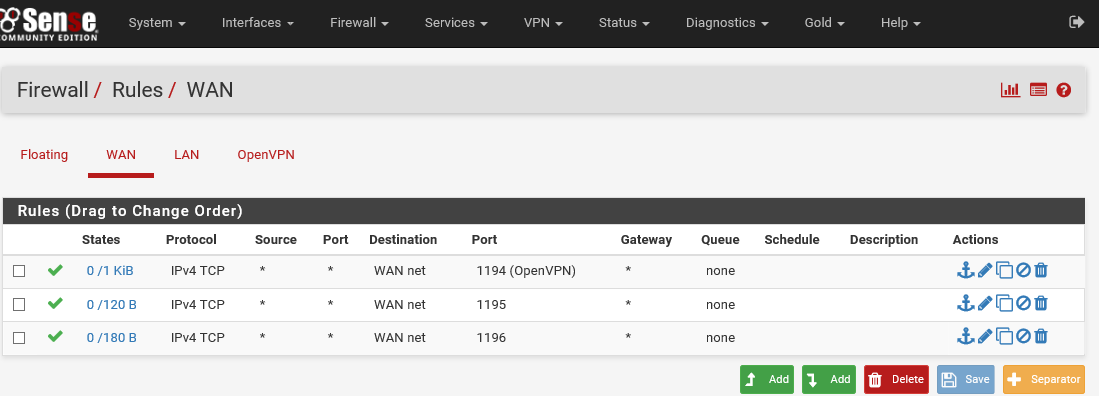
9) Tableau de bord de pfSense, nous avons ici un résumé de sa confiuration actuelle



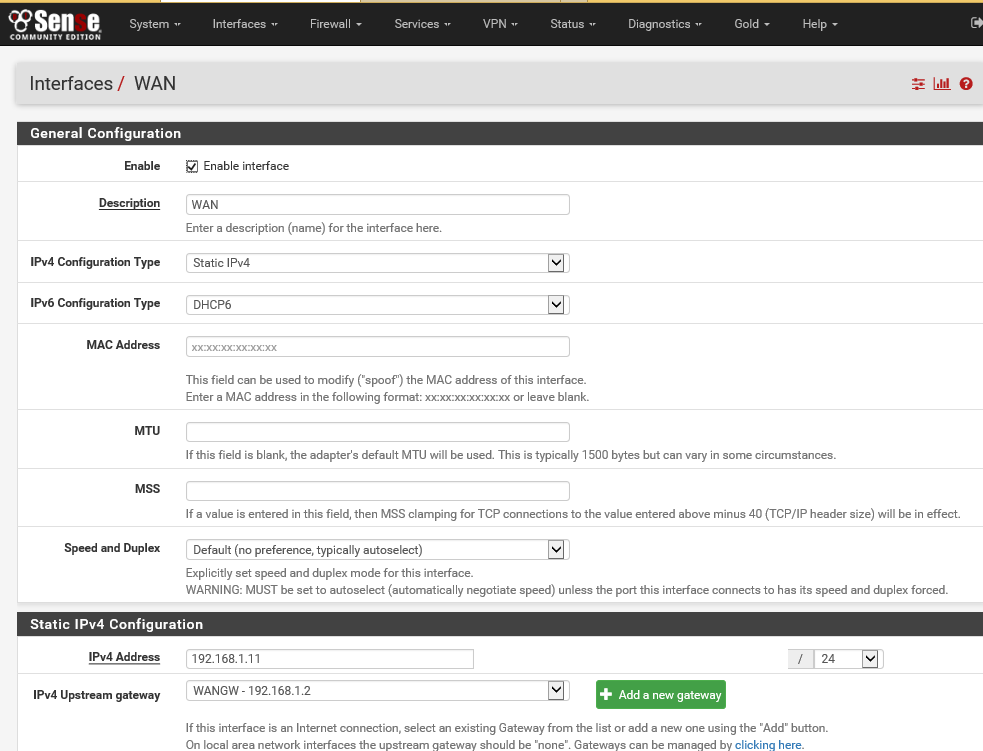
10) Configuration de l'interface WAN du pfSense.



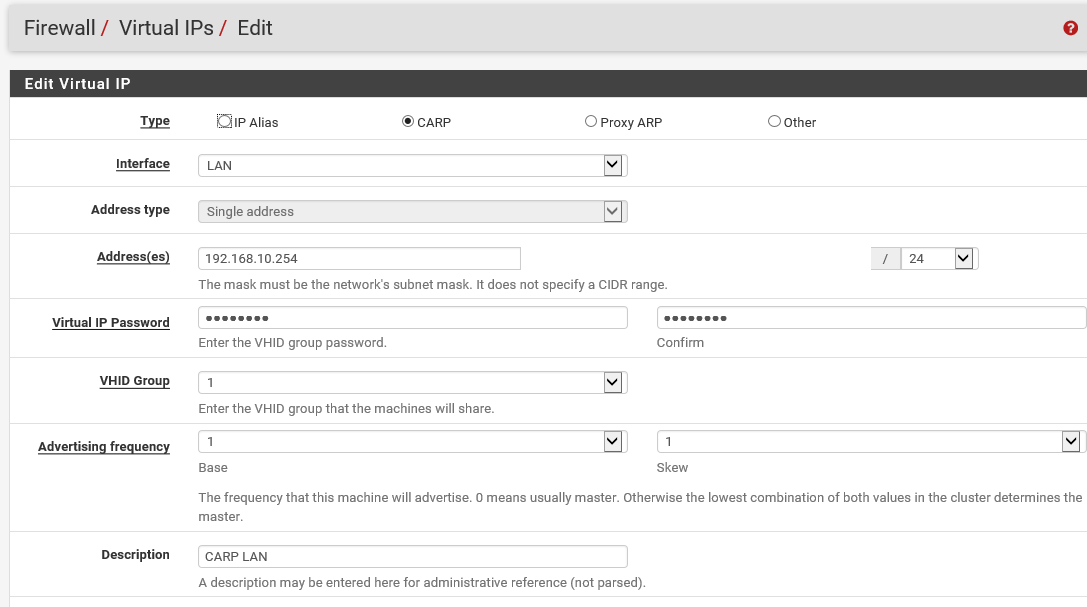
11) Nous créeons les serveurs OpenVPN pour les trois sous-sites



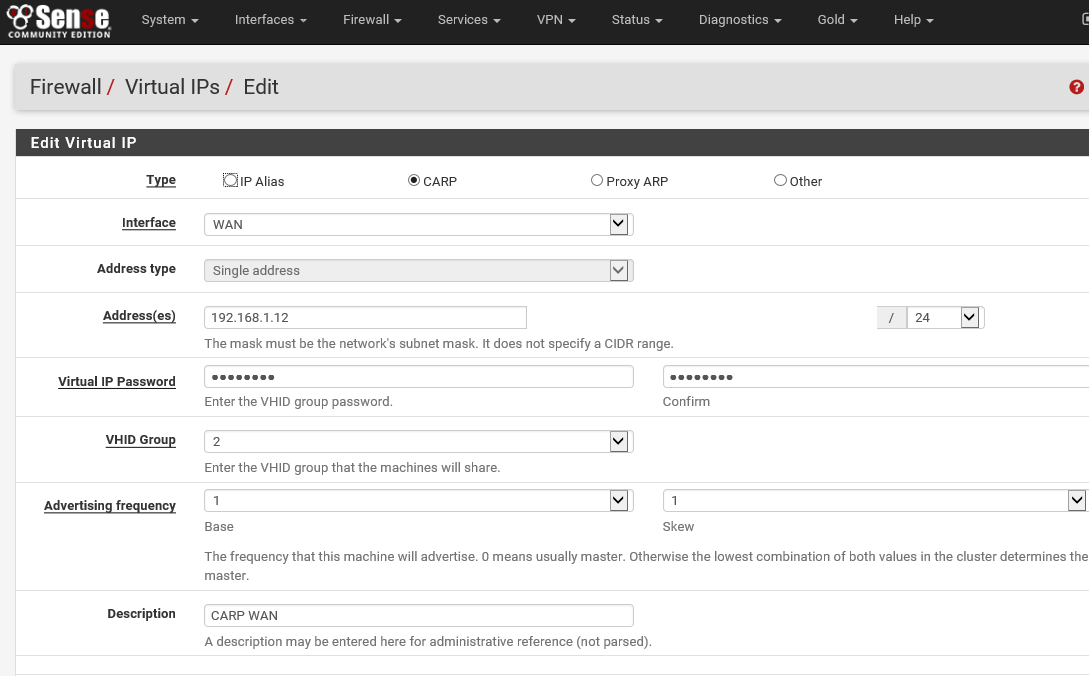
12) Configuration de l'interface LAN



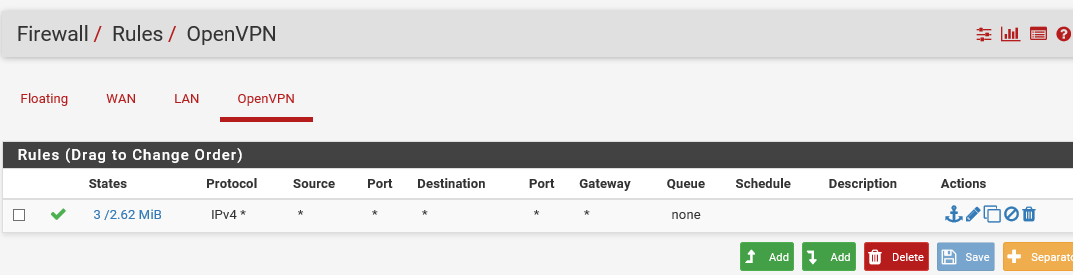
13) Configuration de la virtual IP CARP LAN



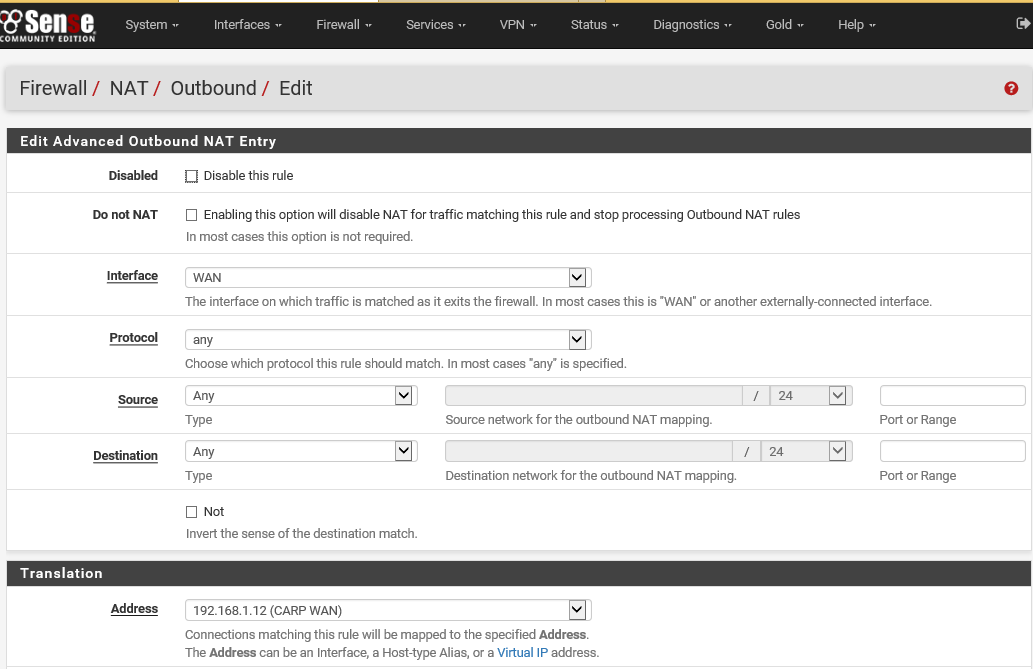
14) Configuration IP CARP WAN



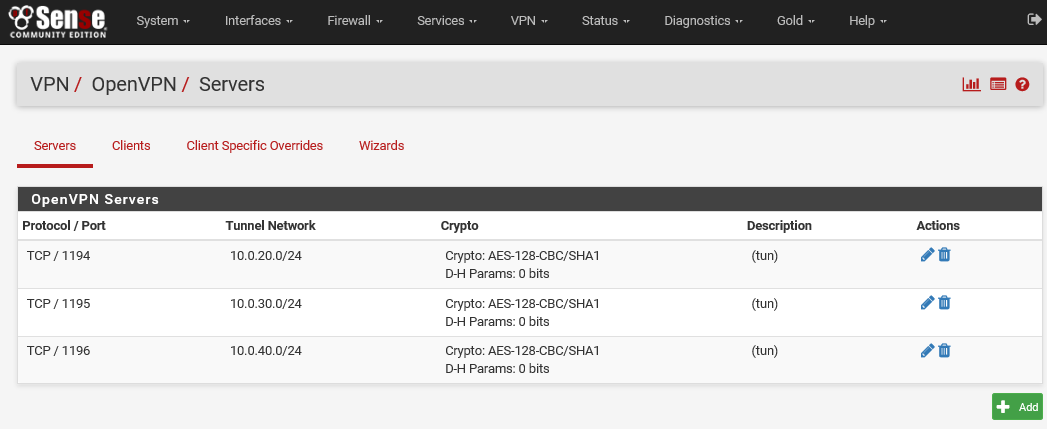
15) On autorise tout type de trafic sur la liaison OpenVPN



16) On force l'utilisation de l'ip virtuelle pour le trafic sortant



17) On créer trois serveurs VPN pour les 3 différents sous-sites



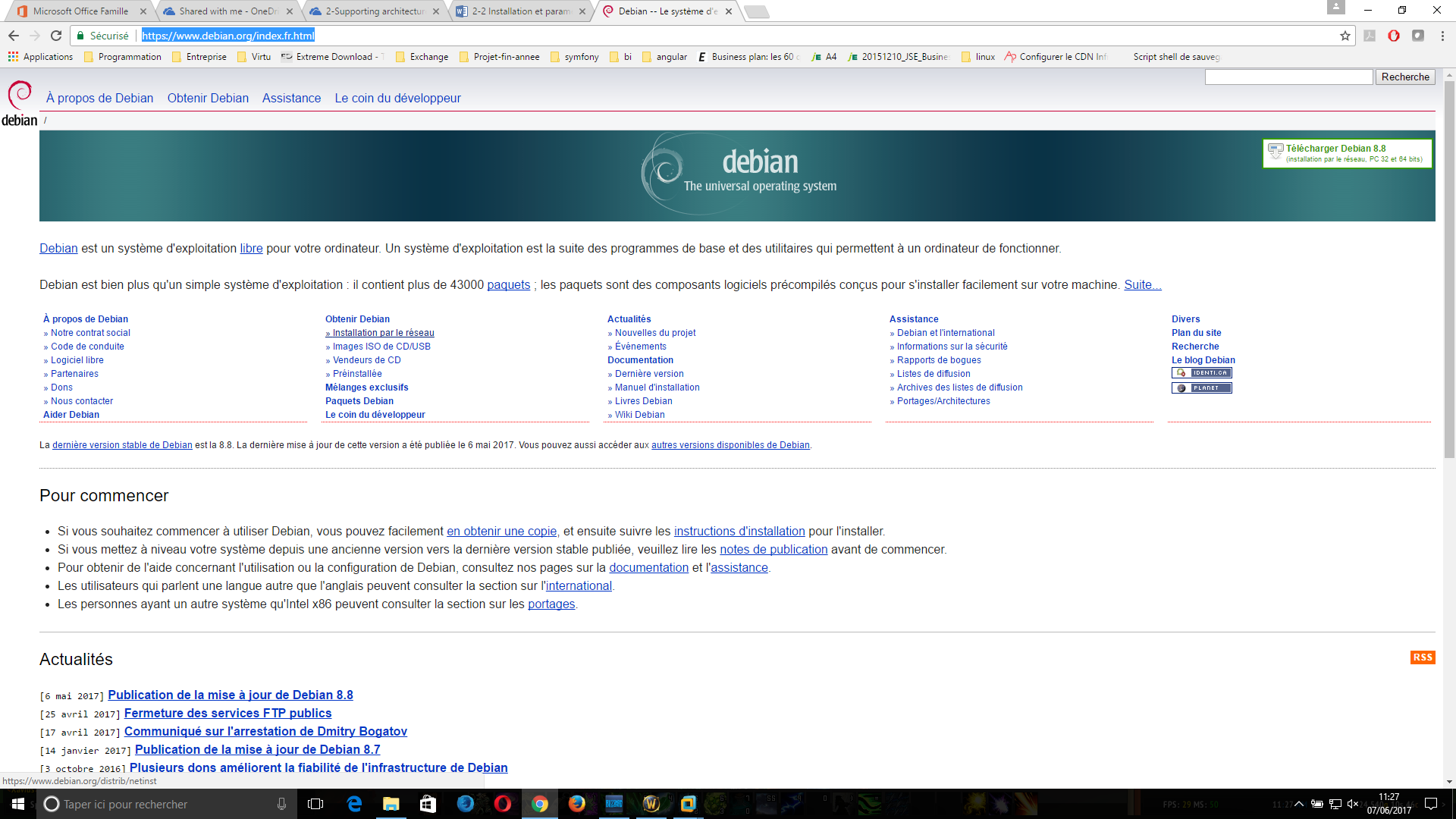
## II. Installation et configuration du cluster MySQL

## Installation d’une distribution GNU/Linux

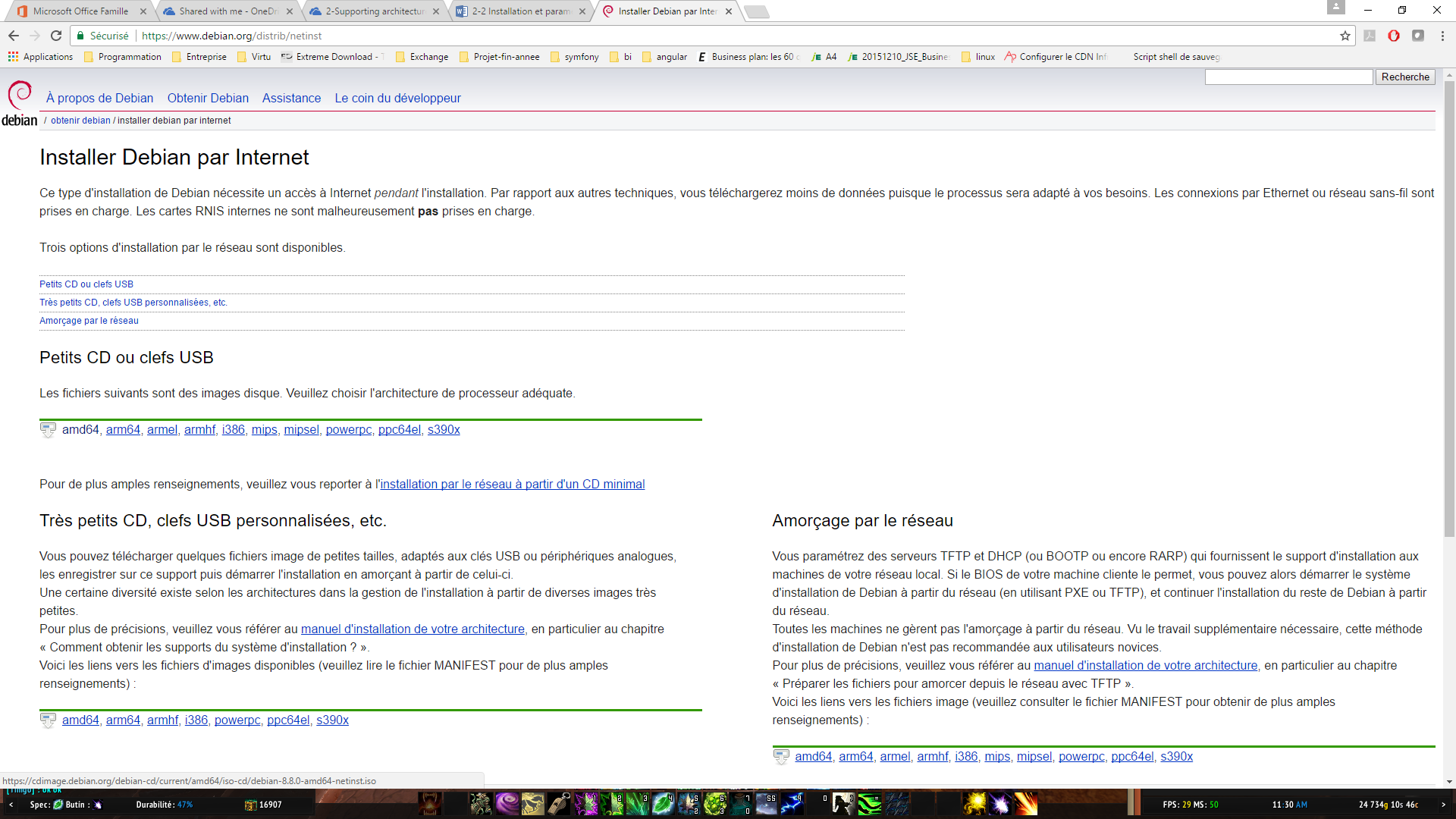
Pour répondre à la problématique de stabilité, nous avons décidé d'utiliser la distribution Debian.

Nous devons nous rentre sur le site officiel de Debian afin de récupérer une image NetInstall et la téléverser dans les datastore des ESX.

Pour se faire rendez vous sur https://www.debian.org/index.fr.html



Puis dans la partie « installation par le réseau »



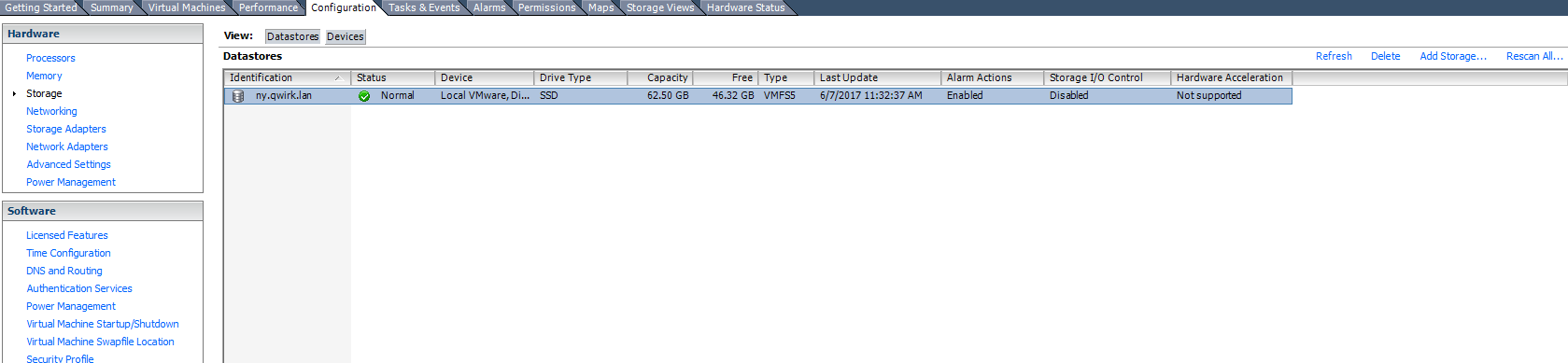
Puis selectionner le format adm64, le téléchargement se lancera automatiquement, il nous reste maintenant à le déposer sur les serveurs ESXI dans leurs datastores puis nous procéderont aux installations des machines virtuelles.

Pour se faire nous devons nous rendre sur notre serveur vCenter puis lancer vSphere Client et nous connecter avec les identifiants suivants :

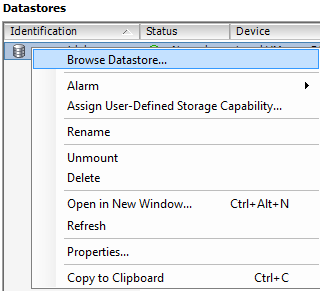
Hostname : Vcenter  
Username : [administrator@vsphere.local](mailto:administrator@vsphere.local)  
Password : Supinf0+



Il suffit de séléctionner un de nos serveurs ESX et se rendre dans l'onglet « configuration » puis, dans le menu de gauche, le lien « storage ».



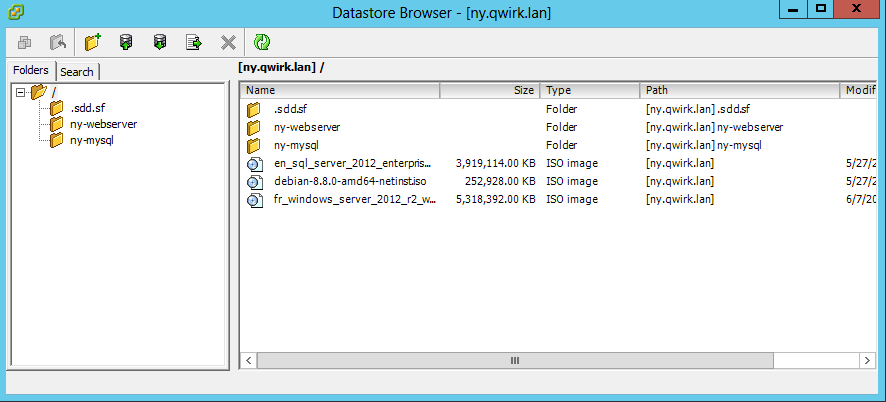
Nous remarquons la présence de notre datastore puis nous effectuons un clique droit dessus puis « browse datastore »



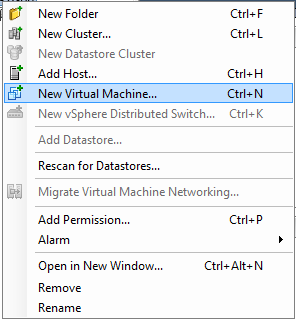
Une page devrait s'ouvrir et vous devez cliquez sur l'icone  puis sur « upload file », maintenant vous devez vous rendre dans le dossier dans lequel vous avez enregistré votre image Debian puis valider.

Nous pouvons constater une barre de progression pour nous indiquer l'état de l'envoie du fichier vers le serveur ESX.

Une fois celle-ci finie, nous devrions voir l'image ISO dans le datastore.

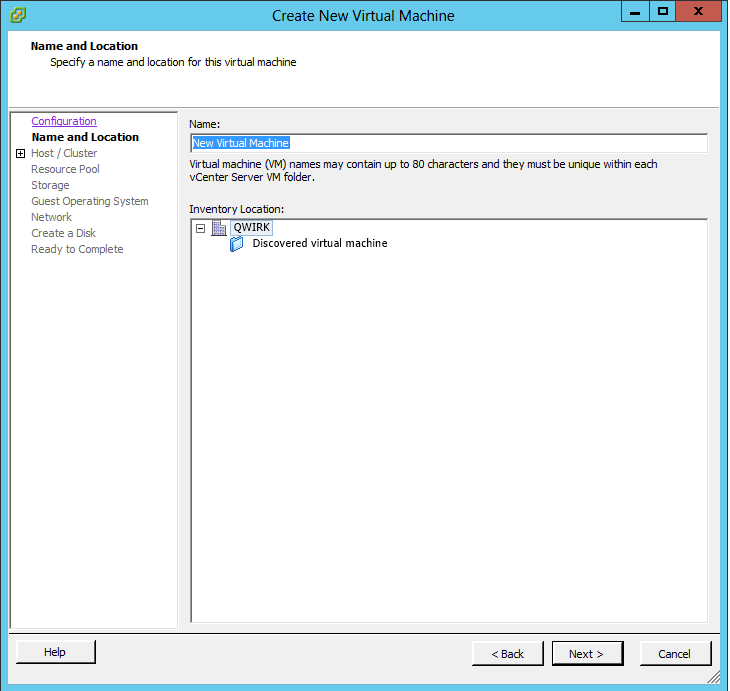


### Maintenant nous allons procéder à l'installation des machines virtuelles.

Effectuer un clique droit sur « Qwirk » puis « new virtual machine » comme indiqué sur l'écran 

Nous laissons les paramètres par défaut.





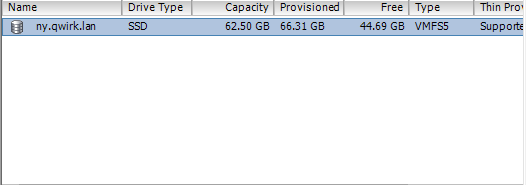
Nous renseignons le nom de notre machine, nommez la « **ny-mysqlserver** » puis cliquez sur le boutton « Next », nous devons le renseigner sur le cluster.



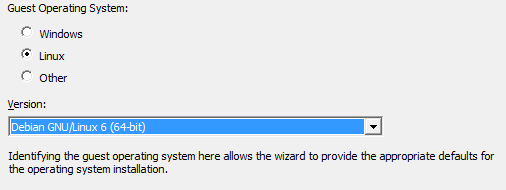
Pour notre hôte, nous allons selectionner l'adresse suivante :



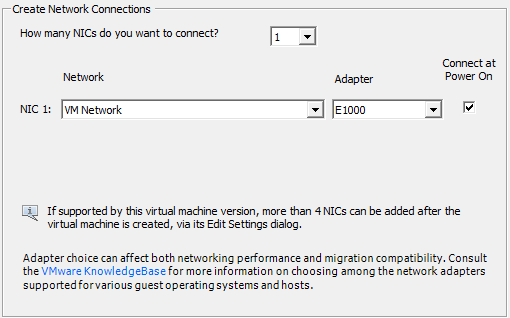
Il nous est demandé de sélectionner l'emplacement de stockage de la machine virtuelle, comme nous n’avons pas de serveurs dédié, nous allons le stocker sur l'espace de stockage du serveur ESX, donc nous laissons la valeur par défaut.



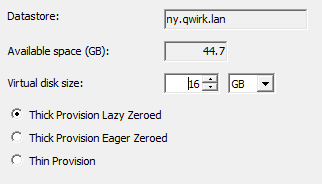
Cliquer sur le boutton « next », puis pour « optimiser » la machine virtuelle nous allons renseigner le futur système d'exploitation qui sera Linux, et la distribution qui sera Debian.



Pour les cartes réseaux nous allons laisser les paramètres par défaut car nous n’avons pas forcement besoin d'en utiliser plusieurs, étant donné que l'on dispose de deux sites physiques.

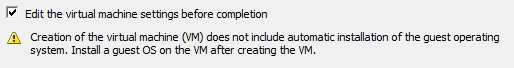


Au niveau du disque, plusieurs choix sont disponible nous allons donc allouer « dynamiquement » de la place pour notre machine, nous allons donc choisir la dernière option.

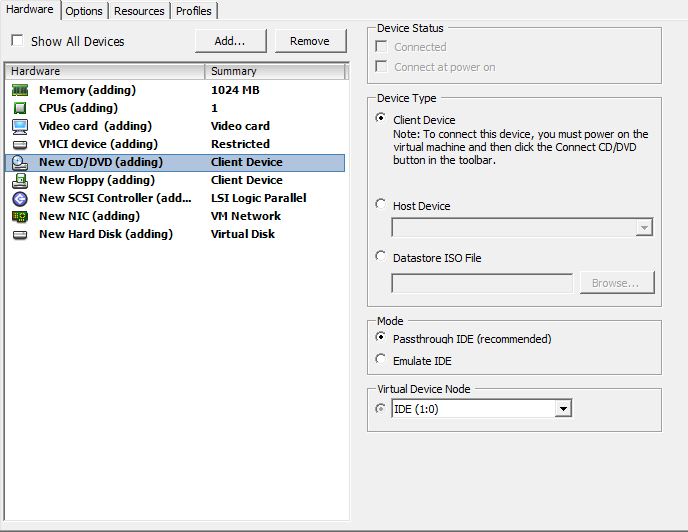


Nous avons un récapitulatif de notre machine virtuelle.

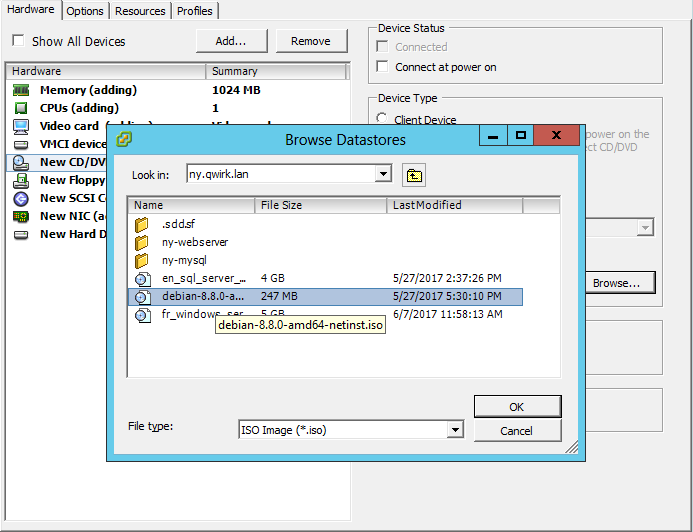
Pensez à cocher « Edit the virtual machine setting before completion » afin de pouvoir insérer le disque Debian sur la machine.



Une page s'ouvre, rendez-vous dans l'onglet CD/DVD puis cliquez sur « datastore ISO file »

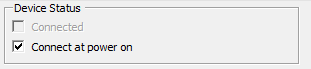


Puis avec "browse", selectionner l'emplacement du datastore de l'esx (dans notre cas ny.quirk.lan) et selectionner le fichier d'installation debian.



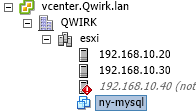
Valider en appuyant sur le boutton "OK".

Une fois ceci fait rendez vous dans l'onglet Device Status et vérifier que le lecteur CD s'allume bien lors du démarage de la machine

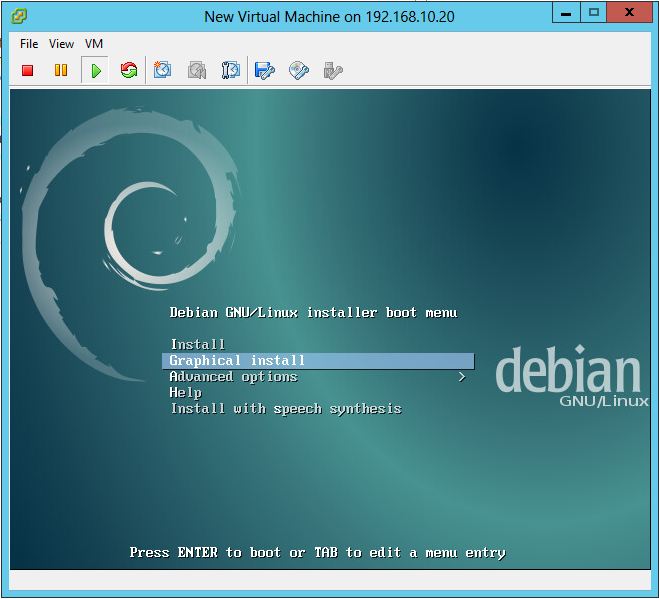


Puis confirmer par le boutton "finish".

Nous pouvons donc apercevoir que notre machine virtuelle est créée.

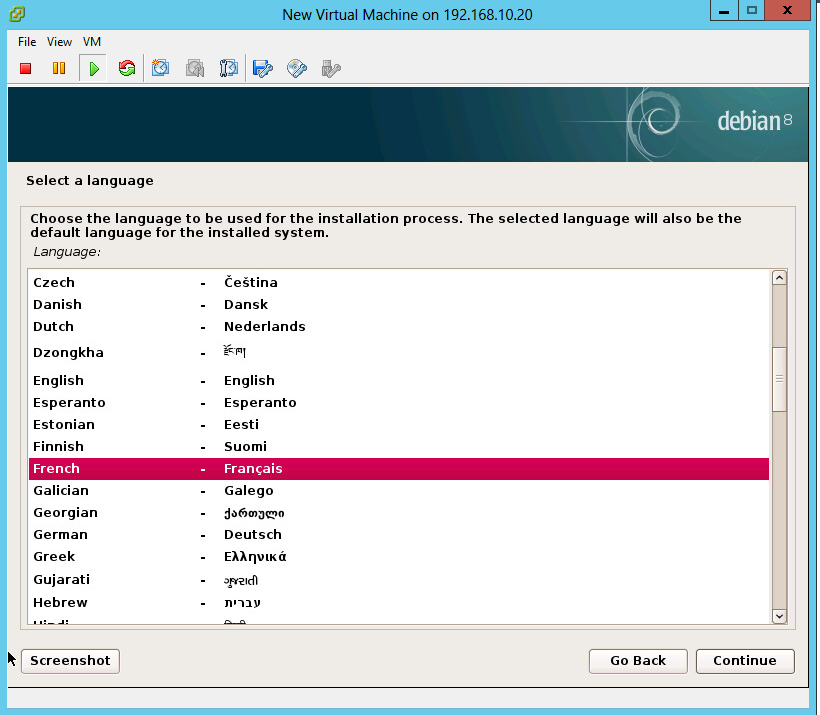


Effectuer un clique droit puis "power on", puis pour avoir un visuel effectuer un clique droit puis "open console".



Nous faisons l'installation en mode graphique.

Au niveaux de la selection des langues, choisir Français



Dans la page suivante renseignez votre pays, dans notre cas "France".

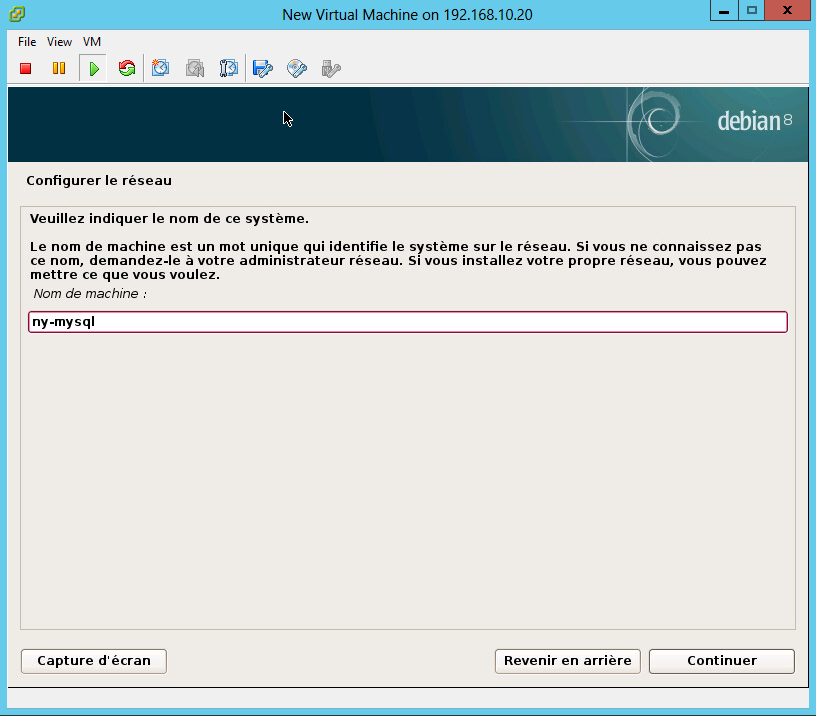


Puis clicker sur "Continuer".

Pour le clavier selectionner "Français".

Le CD est en train de charger les fichiers systémes puis essaie d'attribuer une adresse ip à la machine Debian.

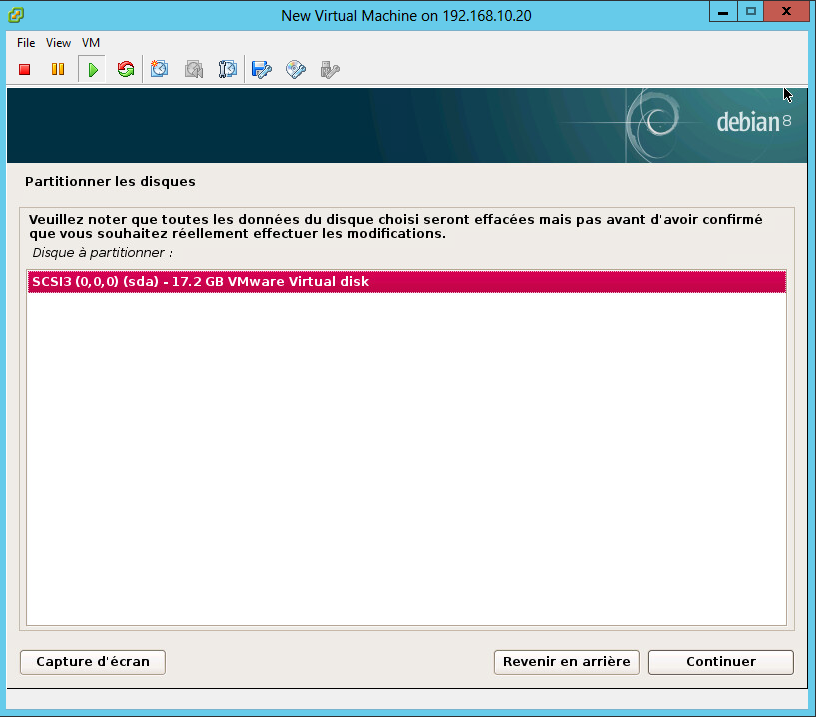
Une fois ceci fait, insérer le nom de la machine, dans notre cas : ny-mysql



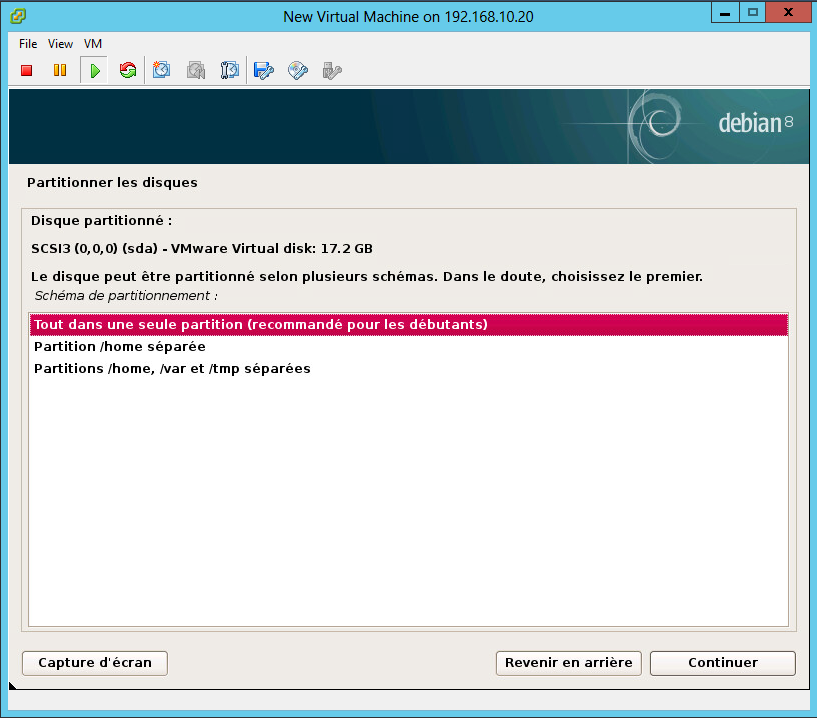
Maintenant nous devons configurer le mot de passe pour le superutilisateur (root) nous suggérons "Supinf0".

Ensuite il nous demande de créer un compte utilsiateur : appellons le mysql-server.

Pour le partitionnement, nous utilisons un partitionnement classique, c’est-à-dire utiliser un seul disque dur virtuel.



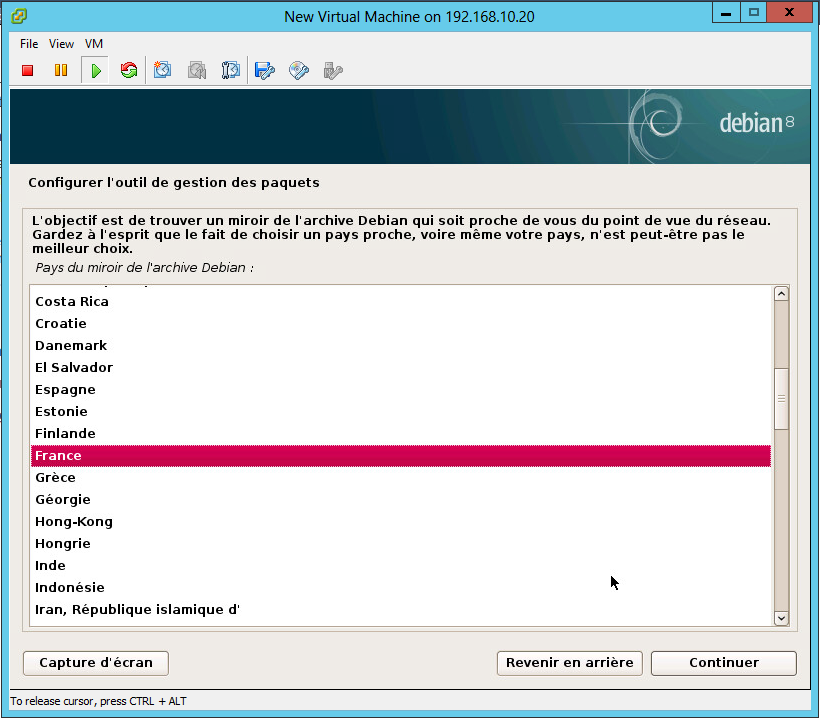
Nous mettons touts dans une seule partition.



Une fois ceci fait le système vous demandera de valider ce choix, validez-le. L'installation du système de base va débuter.

Configuration de l'outil de gestion des packets : afin de pouvoir améliorer la vitesse de téléchargement des packets des serveurs. Pour se faire il faut renseigner le serveur prioraitaire à utiliser.

Dans notre cas nous nous localisons en France donc nous selectionnons "France".



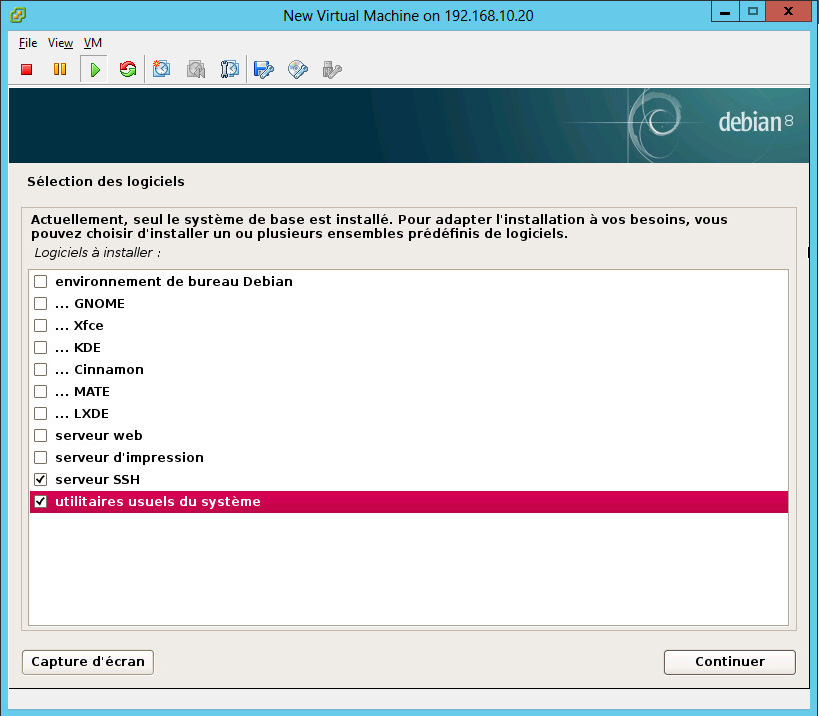
Le serveur ftp.debian.org est un serveur de référence Français.



Au prochain écran, laisser les valeurs par défault nous n'avons pas besoin de passer par un proxy.

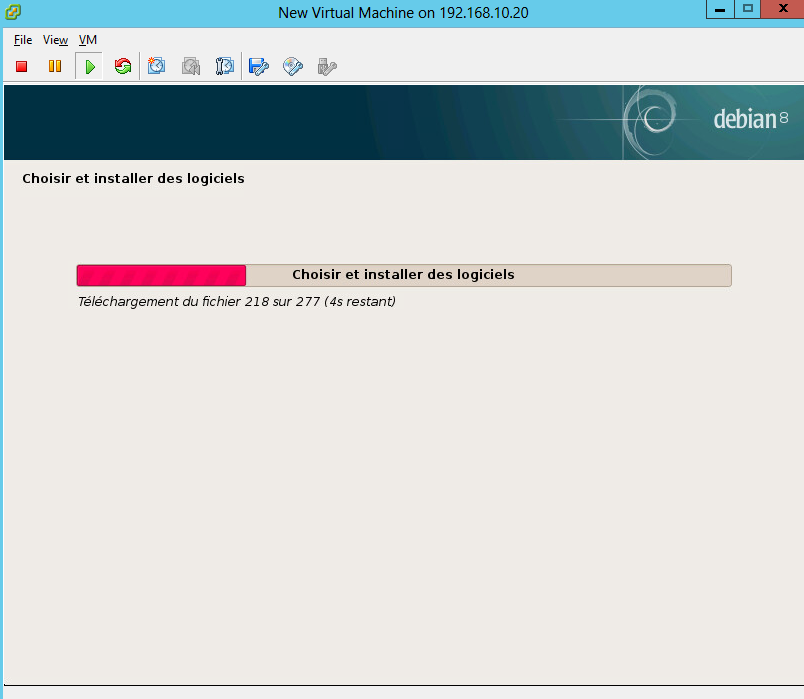
A la question "voulez-vous participez à l'étude de l'utilisation des packets selectionner "non" puis continuer.

Nous arrivons à la fin de l'installation il faut cependant décocher l'environnement de bureau car nous n'avons pas besoin d'interface graphique, cela le rendra plus léger et fiable. Penser à décocher le serveur d'impressions, puis activer le serveur ssh ce qui permettra de se connecter à distance avec d'autres ordinateus, rendant aisé la configuration de celui-ci.



Valider en cliquer "continuer".

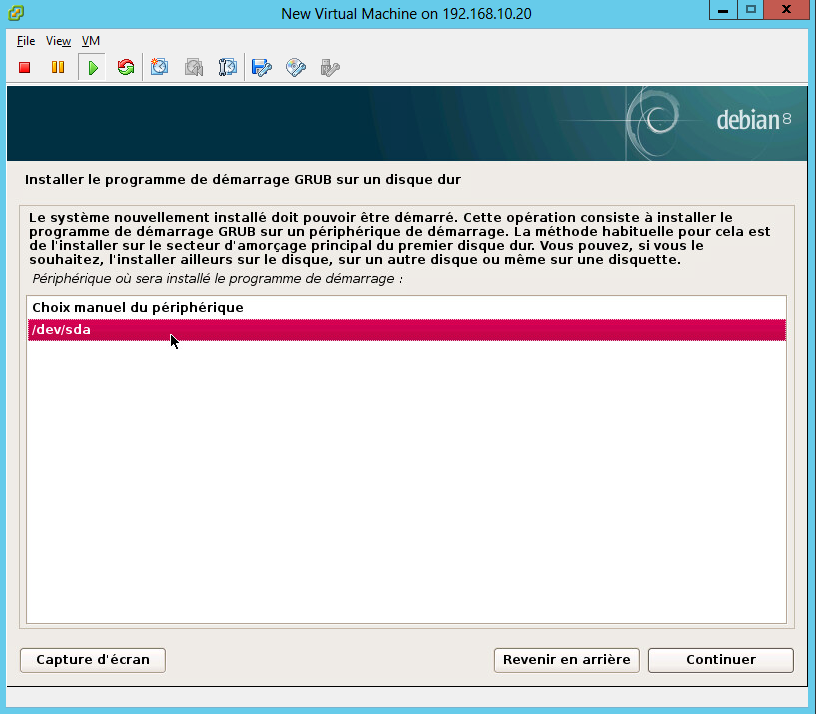
Le système va maintenant effectuer une serie de mise à jours du système et des applications.



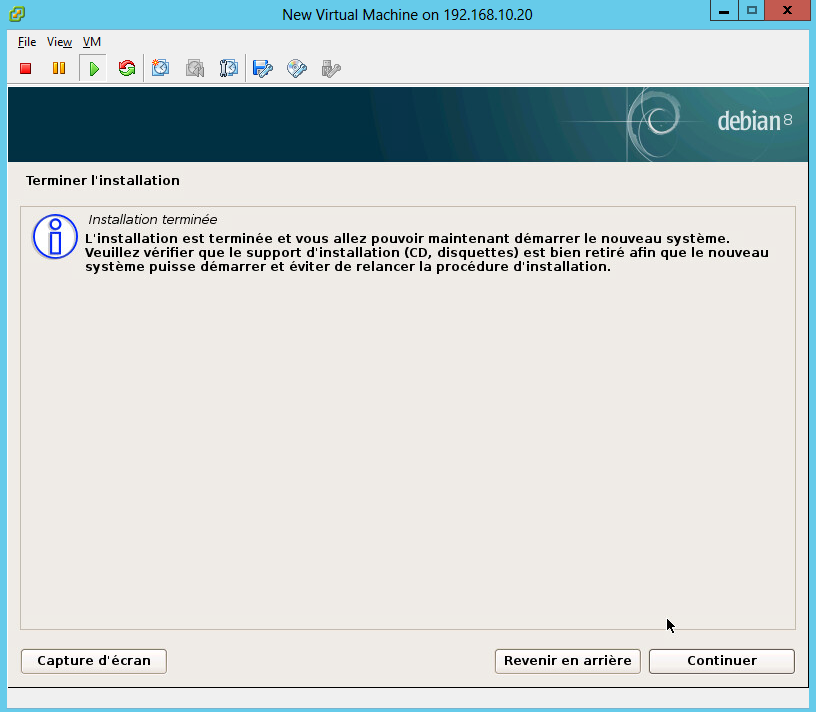
Une fois le système mis à jours il faudra installer GRUB qui permettra de selectionner l'OS à démarrer.



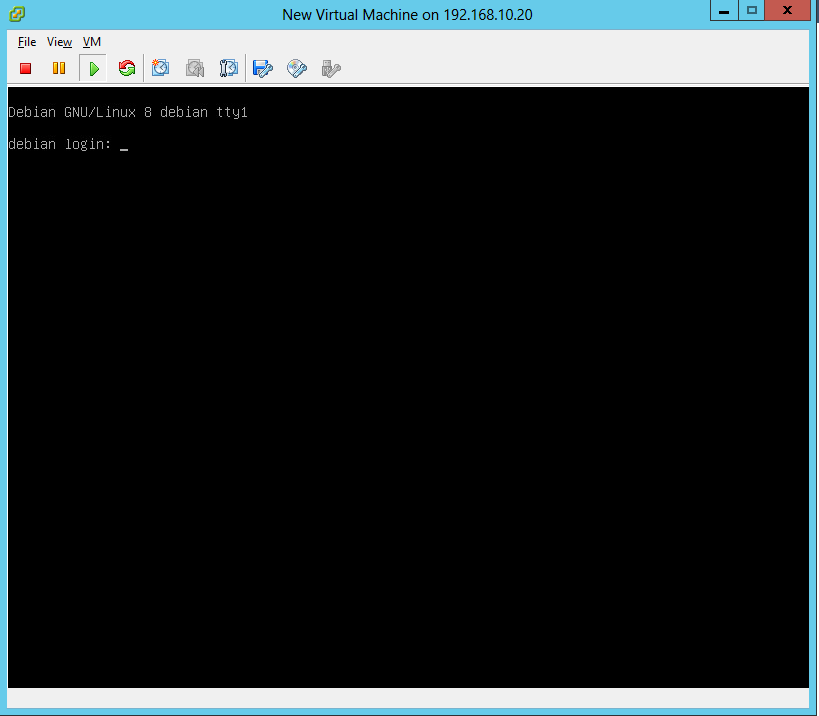
Continuer, et selectionnez /dev/sda



Puis cliquer sur continuer.



La machine redémarera alors et ainsi nous pourrons effectuer la configuration des diverses configurations.



Vous pouvez donc vous connecter avec le nom d'utilisateur et mot de passe que vous avez renseignez durant l'installation.

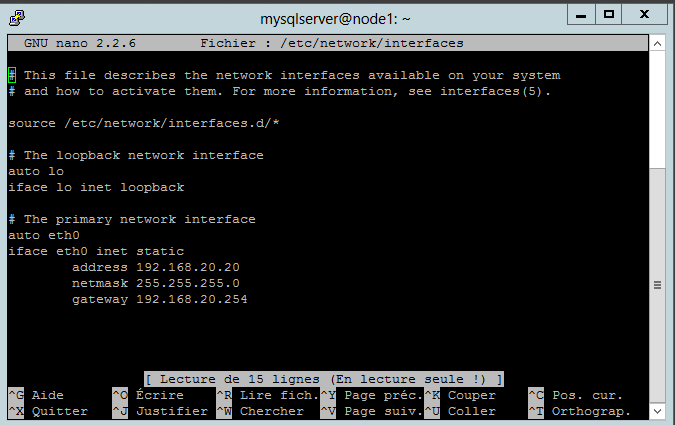
Veuillez répéter ces opérations pour les machines suivantes :

* Nom de la machine : tx-mysql sur l'hotel tx.qwirk.lan
* Nom de la machine ny-webserver sur hote ny.qwirk.lan

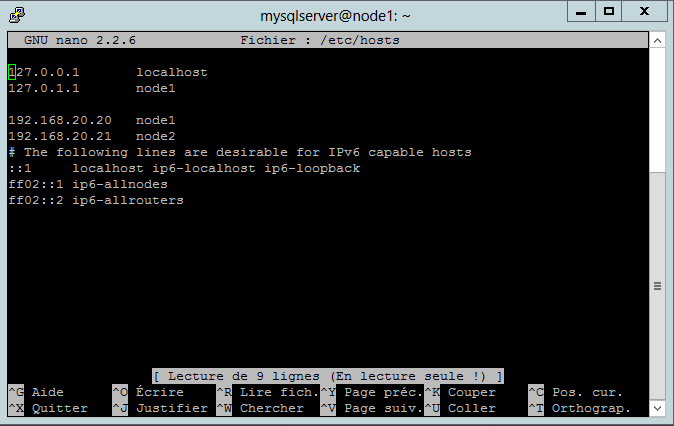
Une fois ceci fait, connectez-vous à l'aide de votre compte 'root' et effectuer les opérations suivantes :

Sur le serveur ny-mysql:

* Nano /etc/networks/interfaces tel que configuré ci-dessous :

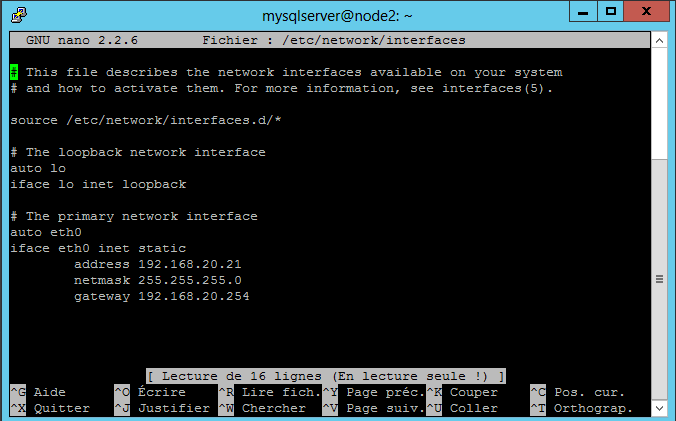


* Nano /etc/hosts configuré tel que configuré ci-dessous :

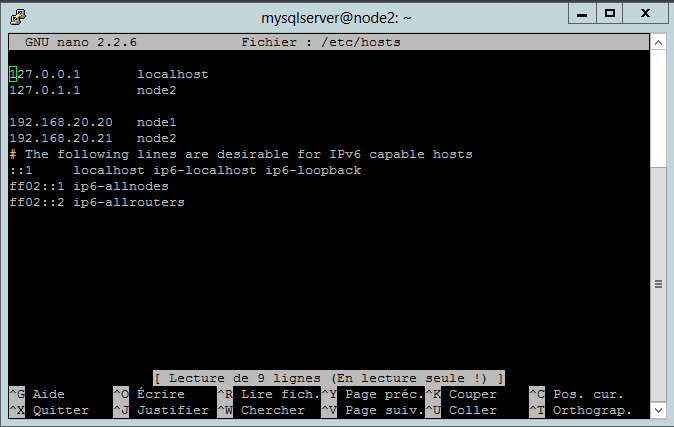


Sur le serveur tx-mysql:

* Nano /etc/networks/interfaces tel que configuré ci-dessous :



* Nano /etc/hosts tel que configuré ci-dessous :



Sur le serveur ny-webserver:

* Nano /etc/networks/interfaces tel que configuré ci-dessous :
* Nano /etc/hosts tel que configuré ci-dessous :

### 1.2 Installation du cluster mysql //pierre

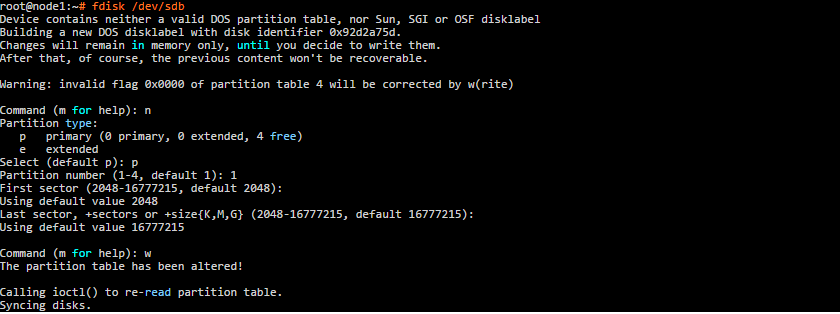
Tout d'abord assurez vous que les serveurs ny-mysql et tx-mysql sont bien allumés et connectez-vous avec votre compte administrateur 'root'.

Rajoutez un disque dur par serveur afin de répliquer la base de données des serveurs. Pour se faire dirigez-vous sur votre machine virtuelle puis rajouter les disques dur.

Premiére étape (à faire sur chaque serveurs) MySQL :

Entrez la commande suivante : fdisk /dev/sdb

Lorsqu'on vous demande la commande, entrez sur 'n' puis 'p' et mettez les valeurs par défaut. Appuyer sur 'w' pour enregistrer.



Puis nous allons installer DRBD pour synchroniser les disques durs des deux serveurs. Pour faire cela, vous devez entrer

|  |
| --- |
| apt-get install drbd8-utils |

Sur chaque serveurs.

Puis nous activons DRBD à l'aide de la commande suivante : modprobe drbd

Nous sommes maintenant prêt à créer une ressource, effectuez la commande sur les deux serveurs :

nano /etc/drbd.d/drbd0.res



Et nous terminons par activer la ressource créée à l'aide des commandes .

Sur le premier node (ny-mysql) :



On passe manuelle le serveur en primaire.

Sur le deuxiéme node (tx-mysql)



On passe le deuxiéme serveur en secondaire.

Afin de suivre la progression de la synchronisation des disques vous pouvez le faire à l'aide de la commande suivante sur le node : 

Une fois celle-ci fini il est temps de formater la partition : sur le note 1 entré la commande



Maintenant nous allons rajouter la haute disponibilité, nous devons installer hearthbeat :

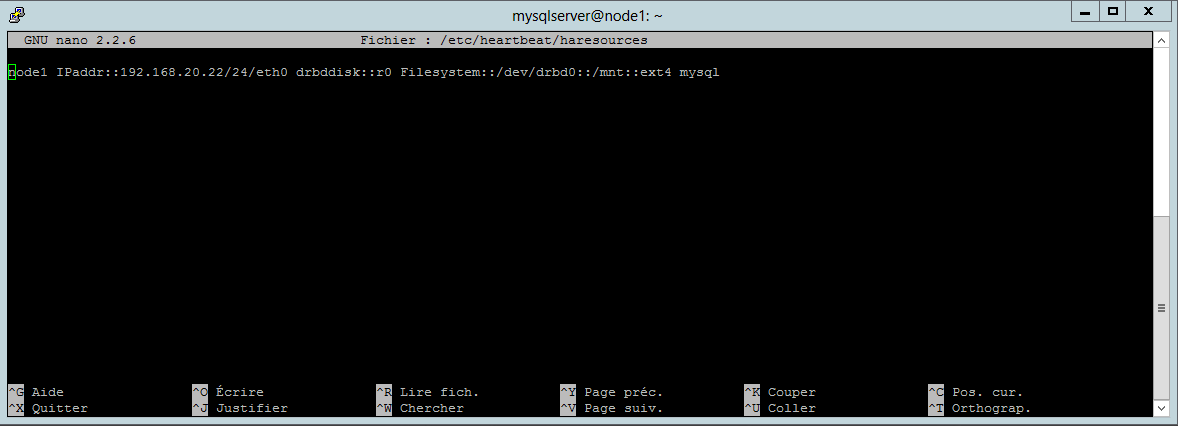
Nous devons entrer la commande suivante sur tout les serveurs : 

Puis nous devons éditer à l'aide de la commande nano /etc/hearthbeat/ha.cf avec les paramètres suivants, sur les 2 serveurs :

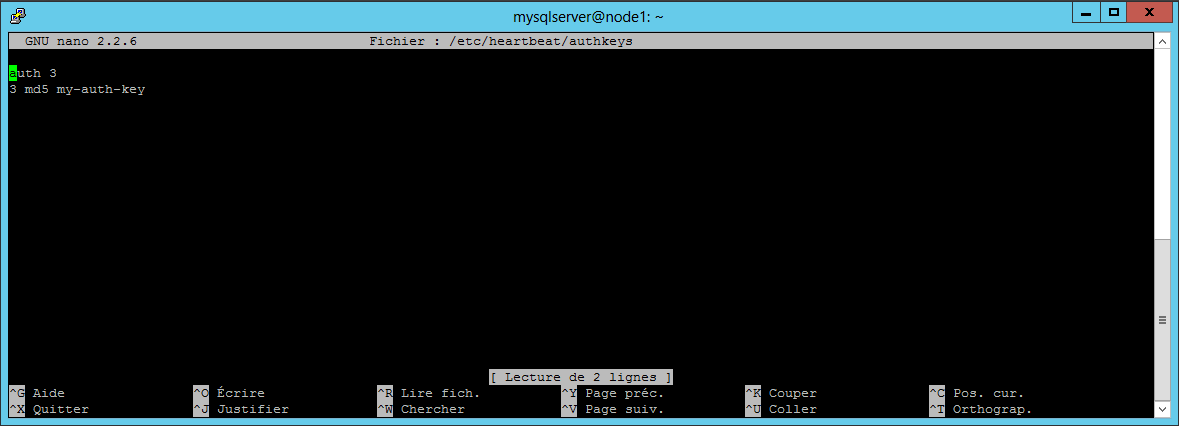


Puis nous allons déclarer l'adresse IP virtuelle du service que l'on veux assurer et utilisée par le cluster.

Pour se faire, il faut éditer le fichier /etc/hearthbeat/haresources avec nano sur les deux serveurs.



Il nous reste à éditer le fichier /etc/hearthbeat/**authkeys sur les deux serveurs avec nano**



Puis nous appliquons les droits avec la commande chmod 600 /etc/ha.d/authkeys sur les deux serveurs

Sur le premier serveur, il nous faut maintenant démarrer hearthbeat à l'aide de la commande

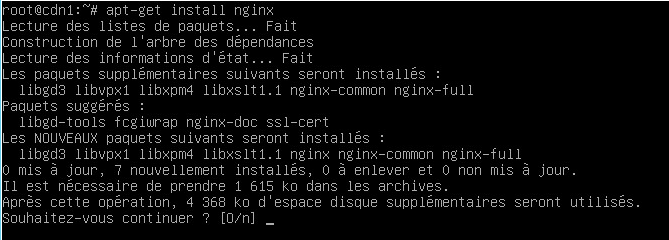
|  |
| --- |
| /etc/init.d/heartbeat start |

Attendez quelques minutes et démarrez le second serveur.

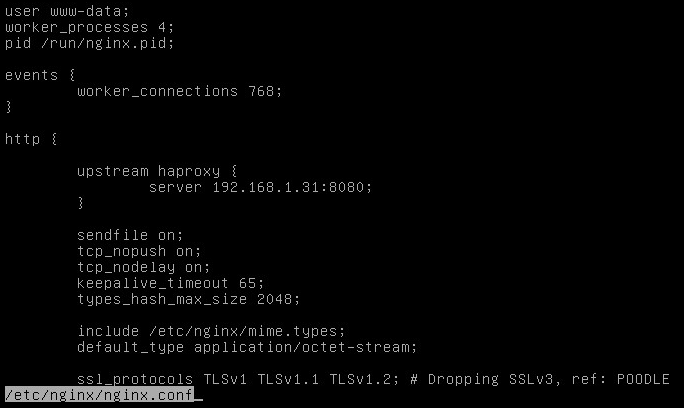
Le cluster de mysql est désormais configuré.

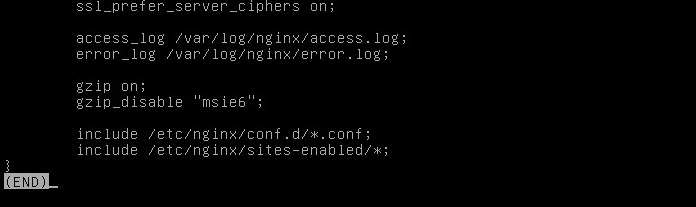
## Installation et configuration des serveurs WSGI & Nginx

*Ce chapitre présente la configuration du reverse-proxy Nginx qui aura le rôle de Content Delivery Network.*

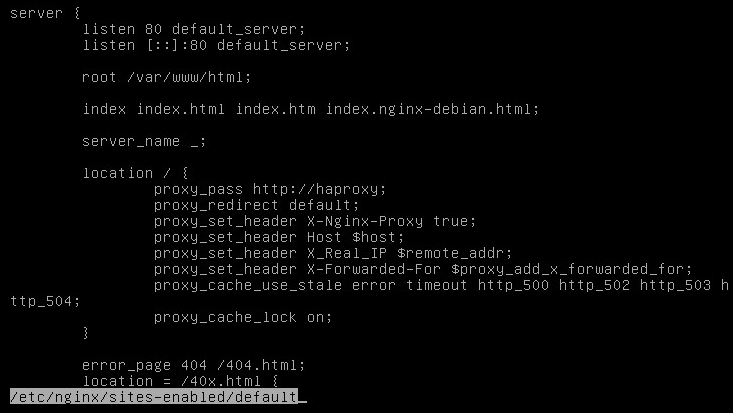


**/etc/nginx/nginx.conf**





**/etc/nginx/sites-enabled/default**





**Serveurs WSGI et NGINX**

Le serveur de développement *ne peut pas* être utilisé en production. En effet, celui-ci n'apporte pas les conditions de sécurité et de performances suffisantes pour garantir un service stable. Le rôle d'un framework n'est pas de distribuer les pages web, c'est au serveur web qu'incombe ce travail.

Le protocole WSGI est une spécification qui permet à un serveur web et une application web Python de communiquer ensemble et ainsi récupérer les pages web générés par votre projet Django.

On distingue ainsi plus clairement les différents services et protocoles qui entrent en jeux.

1. **Navigateur Web → *HTTP* → Serveur WSGI → *WSGI* → Application**
2. **Application → *WSGI* → Serveur WSGI → *HTTP* → Navigateur Web**

D'un côté, vous avez l’application dédié à Qwirk, par exemple. Cette application va recevoir une requête HTTP par le protocole [WSGI](https://en.wikipedia.org/wiki/Web_Server_Gateway_Interface) sous forme d'objet Python. Elle va la traiter, générer une réponse (une page web, par exemple), et envoyer cette réponse sous forme d'objet Python par le protocole WSGI.

De l'autre côté, vous avez le serveur (d'application), qui doit donc être capable de recevoir des requêtes HTTP envoyées par les navigateurs web des visiteurs, de les transformer en objets Python, de les transmettre par le protocole WSGI, d'attendre les réponses sous forme d'objet Python et de les renvoyer au client en HTTP par le réseau.

Nous avons choisi le serveur web [Nginx](http://nginx.com/) pour de très nombreuses raisons, et notamment pour son efficacité et sa simplicité de configuration en tant que proxy. Malheureusement, il ne gère pas WSGI, et ne peut donc pas communiquer directement avec notre application. Il faut un intermédiaire : nous avons opté pour [uWSGI](http://uwsgi-docs.readthedocs.org/en/latest/), qui est à la fois le plus rapide en cas de forte charge et celui qui a la plus faible empreinte mémoire. Il est aussi extrêmement simple à configurer.

**Le proxy**

Nginx est un excellent proxy qui, en plus de transférer les requêtes à uWSGI, va pouvoir délivrer directement les fichiers statiques (images, css, js…), va faire office de cache, va pouvoir communiquer en HTTPS, et va même compresser les réponses avec gzip. De plus, il sait très bien discuter en HTTP avec les navigateurs, c'est sa spécialité.

Nginx va donc se charger de parser les requêtes HTTP pour les transférer à uWSGI via un autre protocole : uwsgi

En résumé, le fonctionnement ressemble à cela :

1. **Navigateur web → *HTTP(s)* → Nginx → *uwsgi* → uWSGI → *WSGI* → Application**
2. **Application → *WSGI* → uWSGI → *uwsgi* → Nginx → *HTTP(s)* → Navigateur web**

Procédure d’installation sur un système Debian ou dérivé :

## Installez les dépendances

sudo apt-get install python3-dev python-pip git redis-server nginx openssl redis-server

sudo pip install virtualenvwrapper

**Virtualenv**

Virtualenv va créer un dossier avec un environnement complet dedans : l’interpreteur Python, les libs, des commandes, etc. C’est votre installation isolée.

Si vous installez une bibliothèque avec pip, il l’installera dans cet environnement virtuel, et elle ne sera pas accessible ailleurs. Si vous lancez la commande python, le shell Python aura accès à toutes les libs de cet environnement virtuel.

Vous pouvez sortir de l’environnement virtuel avec la commande deactivate.

**virtualenvwrapper**   
  
virtualenvwrapper permet de faciliter une procédure rébarbative de changement d’environnement

|  |
| --- |
| pip install --user virtualenvwrapper |

Ensuite il faut rajouter quelques lignes dans votre script d’init de shell, par exemple dans *~/.bashrc*:

|  |
| --- |
| export WORKON\_HOME=~/.virtualenvs  mkdir -p $WORKON\_HOME  source ~/.local/bin/virtualenvwrapper.sh |

Ces lignes vont lancer virtualenvwrapper en permanence dans le shell. Relancez le terminal pour l’activer.

Ensuite, vous avez accès à de nouvelles commandes :

1. mkvirtualenv nom\_env va créer un virtualenv dans le dossier *~/.virtualenvs*, où que vous soyez.
   * workon nom\_env va automatiquement activer un env, où que vous soyez.
   * rmvirtualenv nom\_env va surpprimer l’env du même nom.

Les options de mkvirtualenv sont les mêmes que pour la commande virtualenv, vous n’avez juste plus à vous souciez d’où sont vos envs, ni d’où vous êtes.

source ~/.bashrc

mkvirtualenv qwirk-python -p /usr/bin/python3

#Si l’environnement n’est pas déjà activé  
 workon qwirk-python

pip install Django pytz djangorestframework channels asgi\_redis uwsgi django-cors-headers

## Configuration du SSL Nginx

sudo rm /etc/nginx/sites-enabled/default

sudo rm /etc/nginx/sites-available/default // bonnes pratiques (lien symbolique)

sudo mkdir /etc/nginx/ssl

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/server.key -out /etc/nginx/ssl/server.crt

**Configuration du server.conf**

sudo nano /etc/nginx/sites-available/server.conf

[…]

## Reverse proxy Nginx : Load-balancing

## I. Présentation

Les serveurs et les architectures web actuelles se doivent d'avoir une performance est une disponibilité optimale pour répondre aux besoins des entreprises dont la survie dépend parfois, et pour le bien de l'expérience utilisateur également. Cela passe bien souvent par la mise en place d'un **load-balancing** et d'un **fail-over** avec des éléments qui ont pour rôle de gérer la charge des requêtes et de la répartir selon la disponibilité et l'occupation des ressources. C'est ce que nous allons voir ici en utilisant les serveurs web Nginx.

## II. Reverse proxy, Load-Balancing

**Le Load-Balancing** est le fait de répartir les requêtes qui arrivent sur un service entre plusieurs serveurs (deux ou plus). On peut alors avoir une **diminution de la charge de travail des serveurs** gérés en Load-Balancing ce qui permet d'éviter les surcharges, des ralentissements et des plantages. Un Load-balancing requiert le plus souvent un hôte dit **"front-end"** dont le rôle va être de réceptionner les requêtes et de les répartir entre les hôtes membres du cluster de load-balancing. Ces membres du cluster peuvent être identifiés comme des **"back-end".**

**Un reverse proxy** est un serveur qui va servir d'intermédiaire entre les utilisateurs et les serveurs web. Dans notre cas, le reverse proxy aura pour rôle de jouer les **load-balancer** entre nos serveurs web. De plus, un reverse proxy va également avoir pour mission de rendre invisible toute la partie back end, les utilisateurs s'adresseront à un seul et unique serveur qui gérera ensuite vers quel back-end envoyer la requête.

## III. Présentation de l'architecture Nginx

## IV. Installation & configuration d'Nginx en reverse Proxy

## V. Tests

## VI. Ajustement des paramètres

# Descriptif sauvegarde

## Stratégie

Le sauvegardes sont automatiques. Elles sont générés tout les soir à 2h00 du matin incluant les machines virtuelles mais la base de donnée permettant le bon fonctionnement du projet.

## 

# Descriptif sécurité

## Configuration du HTTPS

https://www.digicert.com/ssl-certificate-installation-microsoft-iis-8.htm

# Annexe 1 : comptes administratifs

|  |  |  |  |
| --- | --- | --- | --- |
| Compte d'administration | Mot de passe | Utilisation | url de connexion |
| administrateur | Ji1&hhzy4 | vCenter Esxi site primaire |  |
| administrateur | kAh7^s3kl | vSphere Client site secondaire |  |
| administrateur | Nfeoifen9 | IIS1 |  |
| administrateur | Jnjnden( | IIS2 |  |
| administrateur | Dede\_Kd | IIS3 |  |
| administrateur | Jnen7(=1 | IIS4 |  |
| administrateur | Djbncjkj6» | CDN1 |  |
| administrateur | Dieni%0 | CDN2 |  |
| administrateur | Huh46hd\_ | CDN3 |  |
| administrateur | 3!ffrJdd | CDN4 |  |
| admin | Oq02°ds | Load Balancer site primaire | http://192.168.1.31:4444 |
| admin | cVhQ10) | Load Balancer site secondaire | http://192.168.1.150:4444 |
| itsolutions | Qdnnfe=3 | DNS MADE EASY | https://cp.dnsmadeeasy.com/ |
| itsolutions | qjYh6$kd | DBaaS Azure | https://azure.microsoft.com/fr-fr/account/ |
| itsolutions | qjYh6$kd | SendGrid Azure | https://azure.microsoft.com/fr-fr/account/ |