

MewPipe

Documentation technique
Architecture

Sommaire

- Introduction
- Réseau
- Architecture physique
- Architecture virtuelle
- Coûts de l'architecture
- SQL server AlwaysOn
- Web Load balancer
- MongoDB
- RabbitMQ
- Vyatta
- VOIP

I. Introduction

Ce document contient l'ensemble des informations demandées par le sujet du projet ainsi que certains détails que nous avons jugé importants.

La documentation pour installer l'environnement nécessaire pour faire tourner MewPipe se trouve dans le document "Instructions de déploiement".

Ce document est destiné à un lectorat "Technique". Cela signifie que l'utilisation de termes techniques liés au domaine de l'informatique, et plus particulièrement de l'administration technique, sera fréquente.

A titre de rappel, notre groupe est composé des personnes suivantes :

- Yilmaz Fatma - 161794
- Quinquis Simon - 143643
- Huynh Eddy - 144074
- Evieux Jean-Baptiste - 144897
- Cholin Théodore - 145731
- Chevalier Alexis - 123750

II. Réseau

Dans cette partie, nous allons expliquer l'adressage ip mais aussi la configuration de notre matériel réseau, c'est à dire les switchs et les routeurs. Pour notre infrastructure nous avons utilisé des **routeurs Cisco 7201** et des switchs Cisco de différentes séries :

- Pour les switchs de Vlans (excepté celui de la production), on utilisera le **Cisco Small Business SG500-28**. Pratique pour un réseau exigeant mais cependant à un prix abordable.
- Pour le switch de la production, il nous faut moins de ports mais d'avantage de puissance, nous avons donc choisi le **Cisco Small Business SG110D-16**
- Enfin pour le switch root, nous avons choisi le **Cisco Catalyst 2960C-8TC-L** qui s'adapte à l'évolution de l'entreprise.

L'adressage IP

Pour notre adressage IP (voir le schéma page suivante) nous avons choisi un adressage permettant pour chaque Vlan d'avoir 254 hôtes ce qui permet d'agrandir l'entreprise sans crainte. Seule le Vlan Production est limité à 14 hôtes car il ne devrait pas évoluer en taille. Nous avons divisé les différents pôles en Vlan.

Le schéma d'adressage IP est disponible en version de meilleure qualité dans le dossier de rendu.

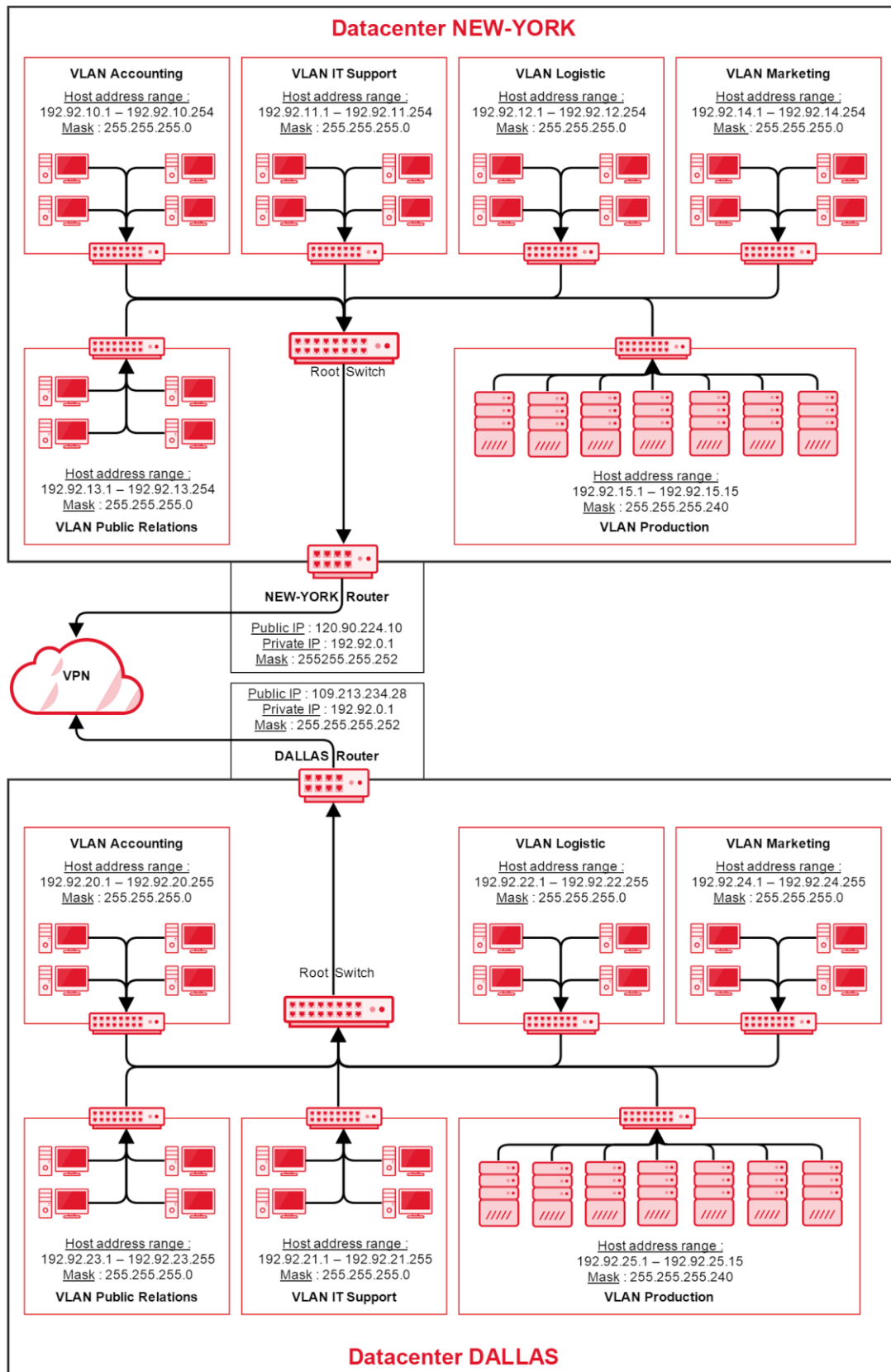


Schéma de l'adressage IP

Configuration d'un VPN IPSec Site-to-site

Quelques informations à savoir sur la topologie avant de commencer :

- Le protocole utilisé est esp
- Pour l'authentification, nous utiliserons la Clé pré-partagée
- Pour la confidentialité nous utiliserons AES 128 bits
- Pour Diffie-Helman, nous utiliserons le groupe 2
- Pour l'intégrité nous utiliserons le SHA-1

Nos routeurs auront 2 phases principales : IKE phase 1 et IKE phase 2

La phase 1 va permettre de configurer les politiques IKE et la phase 2 va permettre de configurer la sécurité IPSec.

Nous allons configurer le tunnel VPN entre les 2 routeurs des datacenters , Routeur NY et routeur TX. Voici les commandes pour la mise en place du VPN :

Autorisation la mise en place du tunnel

```
RouteurNY(config)#ip access-list extended MEWPIPEACL
RouteurNY(config-ext-nacl)#permit ahp host 120.90.224.10 host
109.213.234.28
RouteurNY(config-ext-nacl)#permit esp host 120.90.224.10 host
109.213.234.28
RouteurNY(config-ext-nacl)#permit udp host 120.90.224.10 host
109.213.234.28 eq isakmp
RouteurNY(config-ext-nacl)#exit
```

Définition la politique IKE

```
RouteurNY(config)#crypto isakmp policy 100
RouteurNY(config-isakmp)#encryption aes 128
RouteurNY(config-isakmp)#group 2
RouteurNY(config-isakmp)#hash sha
RouteurNY(config-isakmp)#lifetime 86400
RouteurNY(config-isakmp)#exit
```

Création de la clé pré-partagée sur routeur NY, nous allons utiliser SupinF0

```
RouteurNY(config)#crypto isakmp key SupinF0 address 109.213.234.28
```

Création de la politique IPSec pour la phase 2

```
RouteurNY(config)#crypto ipsec transform-set SUPIPSEC esp-aes 128 esp-sha-
hmac
```

Création de la crypto ACL

ACL qui va permettre d'identifier le trafic qui va passer dans le tunnel, on relie donc chaque VLAN d'un datacenter à ceux de l'autre datacenter

```
RouteurNY(config)#ip access-list extended MEWPIPEACL
RouteurNY(config-ext-nacl)#permit ip 192.92.10.0 0.0.0.255 192.92.21.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.10.0 0.0.0.255 192.92.22.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.10.0 0.0.0.255 192.92.23.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.10.0 0.0.0.255 192.92.24.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.10.0 0.0.0.255 192.92.25.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.11.0 0.0.0.255 192.92.21.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.11.0 0.0.0.255 192.92.22.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.11.0 0.0.0.255 192.92.23.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.11.0 0.0.0.255 192.92.24.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.11.0 0.0.0.255 192.92.25.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.12.0 0.0.0.255 192.92.21.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.12.0 0.0.0.255 192.92.22.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.12.0 0.0.0.255 192.92.23.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.12.0 0.0.0.255 192.92.24.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.12.0 0.0.0.255 192.92.25.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.13.0 0.0.0.255 192.92.21.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.13.0 0.0.0.255 192.92.22.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.13.0 0.0.0.255 192.92.23.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.13.0 0.0.0.255 192.92.24.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.13.0 0.0.0.255 192.92.25.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.14.0 0.0.0.255 192.92.21.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.14.0 0.0.0.255 192.92.22.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.14.0 0.0.0.255 192.92.23.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.14.0 0.0.0.255 192.92.24.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.14.0 0.0.0.255 192.92.25.0
0.0.0.255
```

```

RouteurNY(config-ext-nacl)#permit ip 192.92.15.0 0.0.0.255 192.92.21.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.15.0 0.0.0.255 192.92.22.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.15.0 0.0.0.255 192.92.23.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.15.0 0.0.0.255 192.92.24.0
0.0.0.255
RouteurNY(config-ext-nacl)#permit ip 192.92.15.0 0.0.0.255 192.92.25.0
0.0.0.255
RouteurNY(config-ext-nacl)#exit

```

Création de la crypto MAP qui définit le chemin qu'effectue le tunnel et qui fait la liaison avec le routeur

```

RouteurNY(config)#crypto map MAPMEWPIPE 100 ipsec-isakmp
RouteurNY(config-crypto-map)#set peer 109.213.234.28
RouteurNY(config-crypto-map)#set transform-set SUPIPSEC
RouteurNY(config-crypto-map)#match address MEWPIPEACL
RouteurNT(config-crypto-map)#exit

```

Application de notre crypto map et l'ACL pour établir le VPN

```

RouteurEntreprise(config)#interface s0/0/0
RouteurEntreprise(config-if)#crypto map MAPMEWPIPE
RouteurEntreprise(config-if)# ip access-group MEWPIPEACL out
RouteurEntreprise(config-if)#exit

```

La configuration est la même pour le routeur de Dallas, on inverse juste les IP.

Mise en place des Vlan et de spanning-tree sur les switchs

Créations des vlans

```

Switch-Mewpipe(config)#vlan 10
Switch-Mewpipe (config-vlan)#name accounting
Switch-Mewpipe (config-vlan)#ex
Switch-Mewpipe (config)#vlan 20
Switch-Mewpipe (config-vlan)#name itsupport
Switch-Mewpipe (config-vlan)#ex
Switch-Mewpipe (config)#vlan 30
Switch-Mewpipe (config-vlan)#name production
Switch-Mewpipe (config-vlan)#ex
Switch-Mewpipe (config)#vlan 40
Switch-Mewpipe (config-vlan)#name marketing
Switch-Mewpipe (config-vlan)#ex
Switch-Mewpipe (config)#vlan 50
Switch-Mewpipe (config-vlan)#name logistic
Switch-Mewpipe (config-vlan)#ex
Switch-Mewpipe (config)#vlan 60
Switch-Mewpipe (config-vlan)#name publicrelation
Switch-Mewpipe (config-vlan)#ex

```


Assignation des ports aux vlans, voici un exemple de commande

```
Switch-Mewpipe(config)#interface fastEthernet 0/1
Switch-Mewpipe(config-if)#switchport mode access
Switch-Mewpipe(config-if)#switchport access vlan 10
Switch-Mewpipe(config-if)#ex
```

Pour les ports switch vers switch on utilise le mode trunk

```
Switch-Mewpipe(config)#interface gigabitEthernet 1/0/1
Switch-Mewpipe(config)#switchport trunk encapsulation dot1q
Switch-Mewpipe(config-if)#switchport mode trunk
```

Nous allons maintenant installer le spanning-tree sur nos switches, ce protocole va permettre de gérer les boucles sur le réseau local dans le cas de lien redondant. Il faut savoir qu'un switch sera root .

Activation du spanning-tree sur le switch

```
Switch-Mewpipe(config)#spanning-tree mode rapid-pvst
```

Fixation du switch root

Voici la commande pour fixer le switch root, à savoir qu'un seul Switch par réseau doit être root, ici on le nomme root pour les différents Vlans de notre infrastructure :

```
Switch-Mewpipe(config)#spanning-tree vlan 10-60 root primary
Switch-Mewpipe(config)#end
```

III. Architecture physique

Dans cette partie, nous allons aborder l'architecture physique que nous avons prévu pour l'ensemble de MewPipe.

Schéma

Ce schéma est disponible en meilleure qualité dans le dossier de rendu.

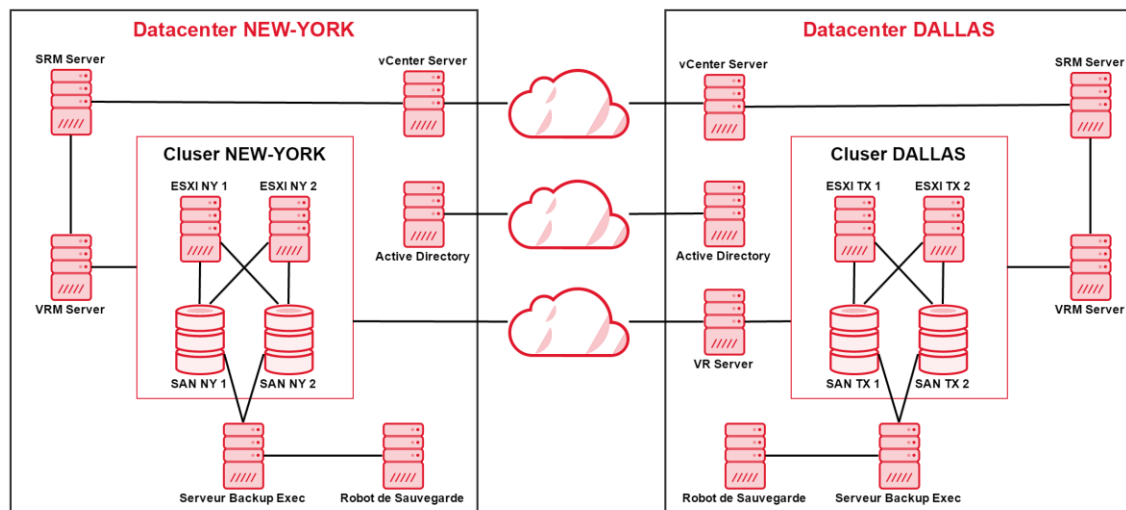


Schéma de l'architecture physique

Choix Techniques et fonctionnement

Pour nos choix techniques sur MewPipe, nous avons choisi d'utiliser :

- vCenter 6
- ESXI 6
- vSphere Replication Management Server
- Site Recovery Manager
- vSphere Replication Server
- Active Directory
- Backup Exec

Ces technologies sont en harmonie puisqu'elles sont toutes issues de VMware. C'est en partie les raisons pour lesquelles nous avons choisi ces technologies afin d'atteindre au maximum une synergie au sein de l'infrastructure.

vCenter va nous permettre de gérer les machines virtuelles, mais aussi les clusterings entre les ESXI. C'est aussi grâce à lui que nous pouvons configurer les ESXI (carte réseau , storage , monitoring ect.) En somme il va permettre de gérer nos Hyperviseurs. Il faudra cela vSphere Client pour se connecter à cette interface et pouvoir gérer ces différentes fonctionnalités.

ESXI est notre hyperviseur qui va héberger nos machines virtuelles mais aussi offrir les différentes ressources dont la machine virtuelle aura besoin.

vSphere Replication Management Server est le serveur qui instrumente la réplication entre les deux sites, c'est ici que vous allez choisir quels datastores vous voulez répliquer / restaurer.

Site Recovery Manager va permettre d'orchestrer automatiquement la reprise des services sur le second site si le premier site est HS.

vSphere Replication Server sera le service qui va récupérer les flux envoyés des ESXI pour répliquer les données vers le site passif.

Active Directory est notre annuaire d'utilisateur au sein de l'entreprise MewPipe.

Backup Exec va nous permettre d'effectuer des sauvegardes de nos SAN dans laquelle notre environnement MewPipe sera enregistré.

Ressource par machine

Chaque machine sur le schéma correspond à une machine physique. Nous avons donc au total 15 serveurs pour gérer notre infrastructure.

	Coeurs processeur	Mémoire (GO)	Stockage (GO)
vCenter	4	16	100
ESXI	24	128	100
vSphere Replication Management Server	4	8	100
Site Recovery Manager	4	8	100
vSphere Replication Server	4	8	100
AD Windows Server 2012 R2	4	8	100
Serveur Back up Exec	4	8	100

Stockage et Backup

Nous avons 4 SAN, 2 par site. Les 4 SAN doivent être identiques. Ceux-ci sont connectés directement en câble fibré à un serveur de Backup qui est Backup Exec. Les sauvegardes différentielles se feront tous les jours, une sauvegarde complète tous les vendredis soirs, ainsi que tous les mois, avec une rétention de 3 mois sera configurée. Les Backup execs sont directement connectés à des robots de sauvegarde.

Haute disponibilité sur les VMs

Chaque ESXI est dans un cluster qui va gérer la haute disponibilité entre les VMs. Si un hyperviseur tombe ayant une VM active, celle-ci remontera sur l'autre. C'est grâce à vSphere High-Availability que nous pouvons gérer cela.

IV. Architecture virtuelle

Nous allons à présent étudier l'architecture virtuelle nécessaire pour faire fonctionner MewPipe.

Schéma

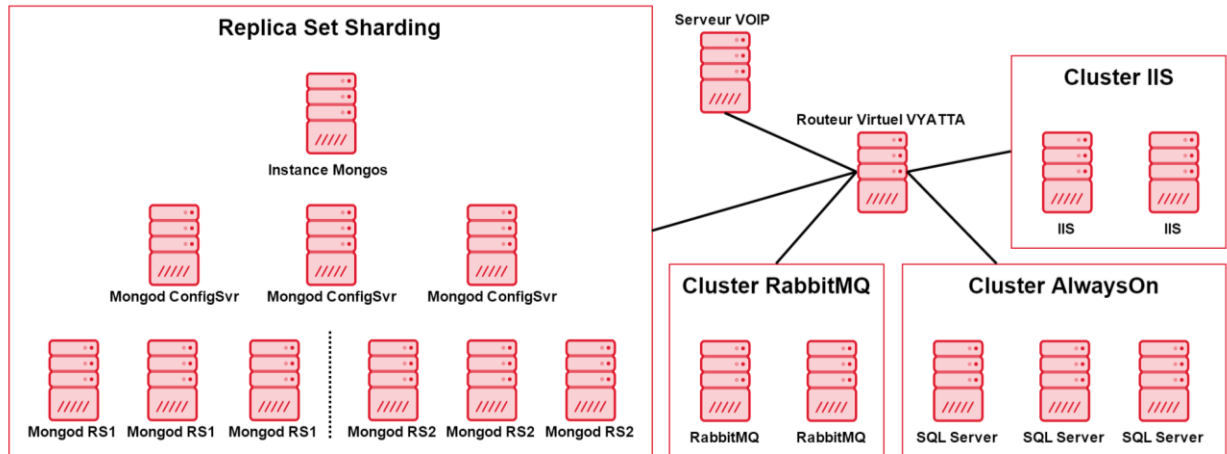


Schéma de l'architecture virtuelle

Ressources par machine virtuelle

	Coeurs processeur	Mémoire (GO)	Nombre de VMs
PFSENSE	1	2	1
IIS	2	4	2
RabbitMQ	2	4	2
SQL Server	2	8	3
Mongos	2	4	1
Mongod configsvr	2	4	3
Mongod Replica Set	2	4	6
Total	13	82	18

Le storage est géré par le SAN et sera extensible en fonction du nombre de vidéos sur MewPipe.

Pour plus d'informations concernant la configuration des services, se référer à la documentation prévue à cet effet.

V. Coûts de l'architecture

Afin de vous donner une estimation du coût de déploiement de l'architecture en production, nous avons établi une liste de la configuration matérielle optimale pour faire fonctionner la totalité de l'architecture.

	Nombre de pièces	Prix à l'unité
Routeurs Cisco 7201	2	3 062,86€
Switch Cisco Small Business SG500-28	10	699,95€
Switch Cisco Small Business SG110D-16	2	209,95€
Switch Cisco Catalyst 2960C-8TC-L	2	639,95€
Serveur Dell PowerEdge R220 8 Go 4C	10	1 344€
Serveur Dell PowerEdge R220 16 Go 4C	2	1 506€
Serveur Dell PowerEdge R715 124 GO 24C	2	5 512€
SAN Dell PowerVault MD3220i 5x1T	4	10 858,15€
Total		85 733,42€

VI. SQL server AlwaysOn

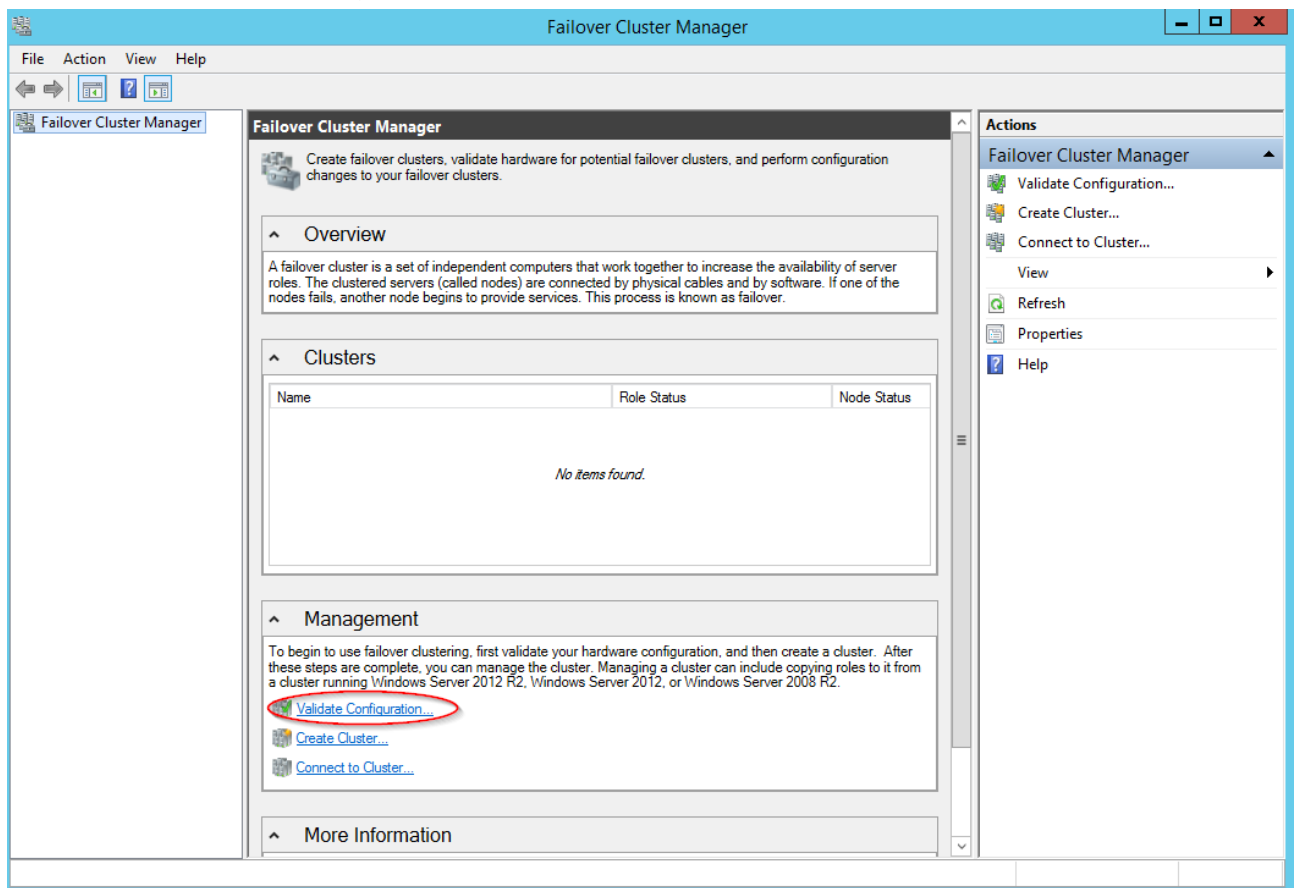
Dans cette section, nous allons voir comment configurer le cluster SQL Server AlwaysOn.

Pré requis

- 2 cœurs
- 4go de ram
- 30 go OS
- 40 go DATA
- 40 go temp
- 80 go Backup
- 40 go db
- Windows Server 2012 R2

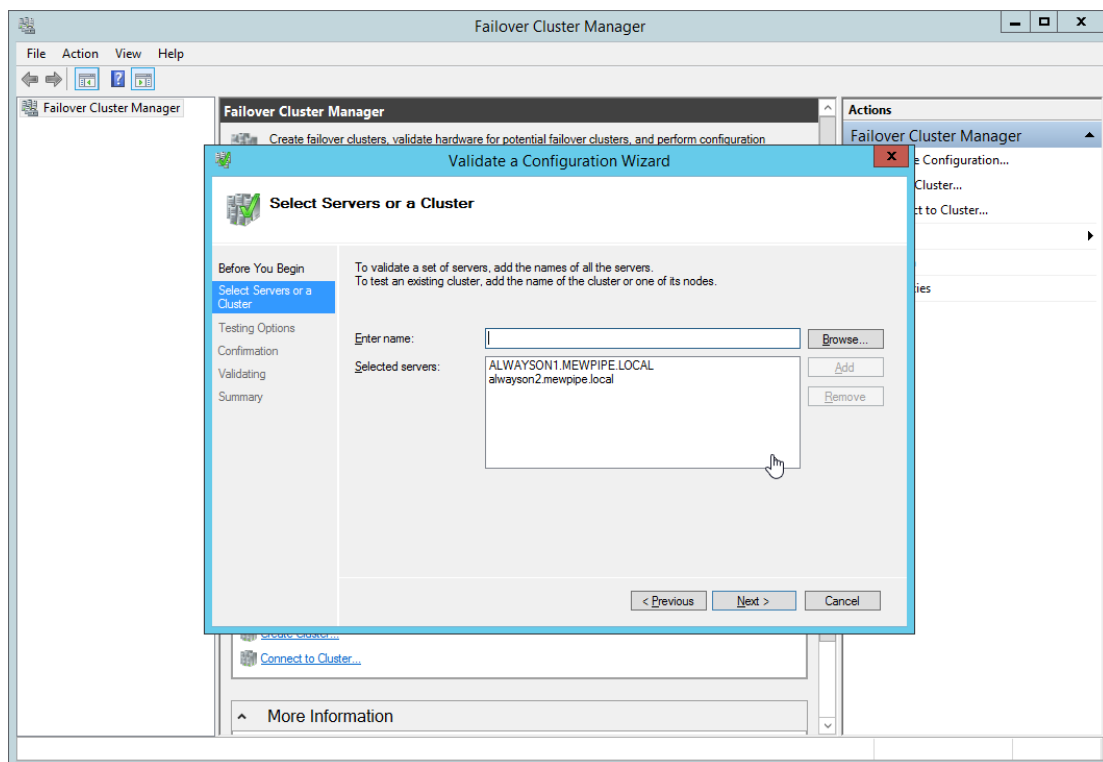
Il faut tout d'abord installer la feature Failover Clustering sur les deux machines qui vont servir d'Always On.

Validation de la configuration du cluster



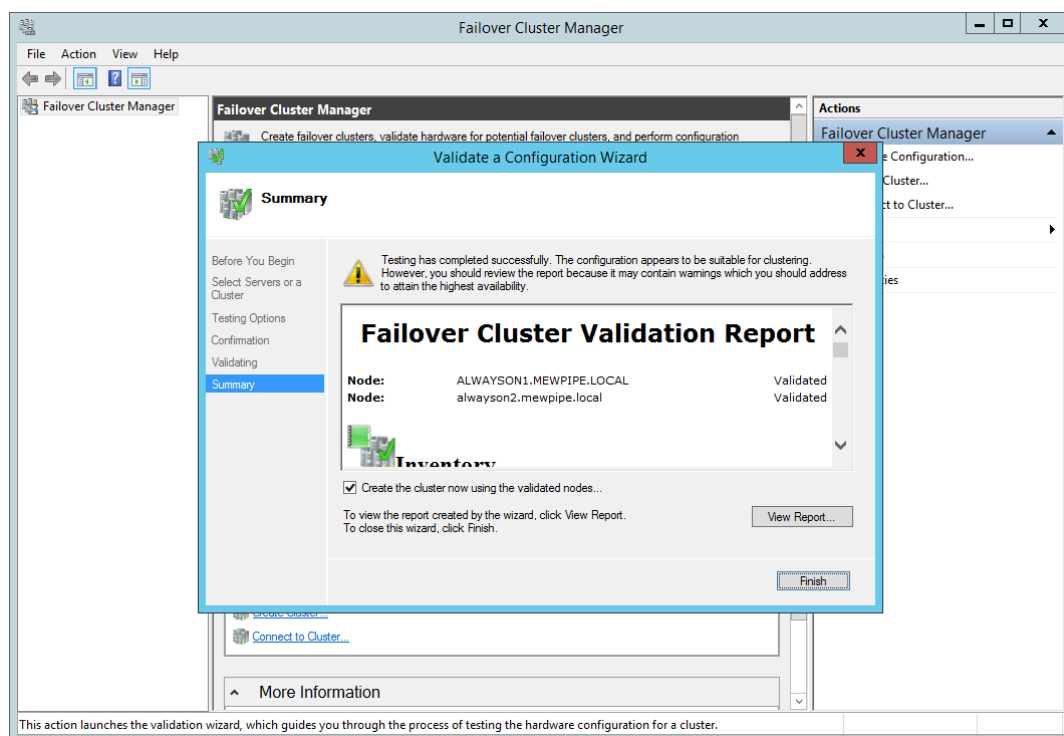
Visuel de la validation de la configuration du cluster

Ajout des deux noeuds du cluster



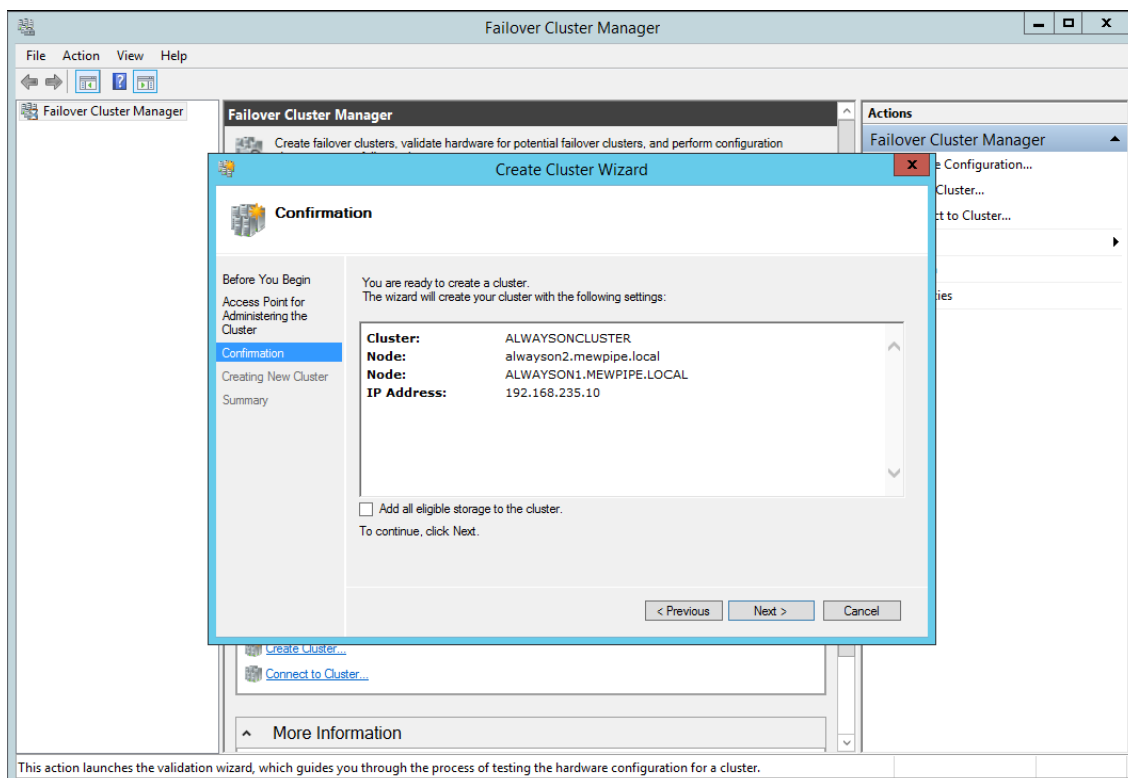
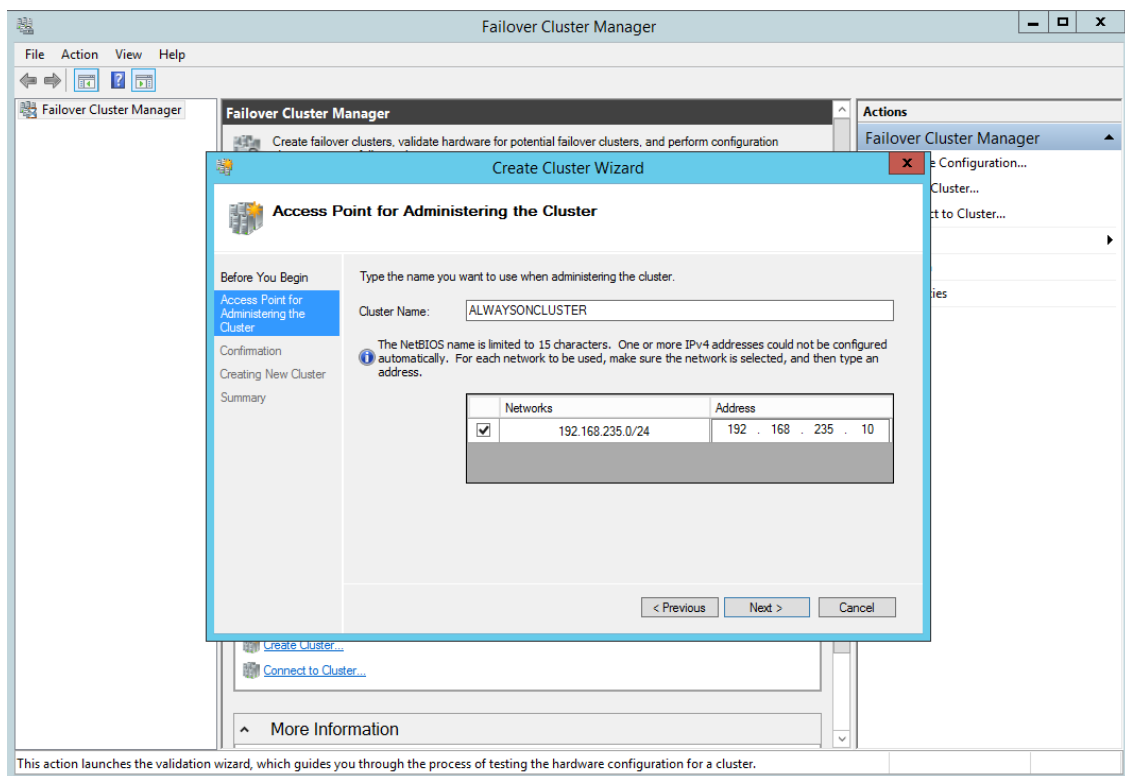
Visuel de l'ajout des noeuds au cluster

Création du cluster



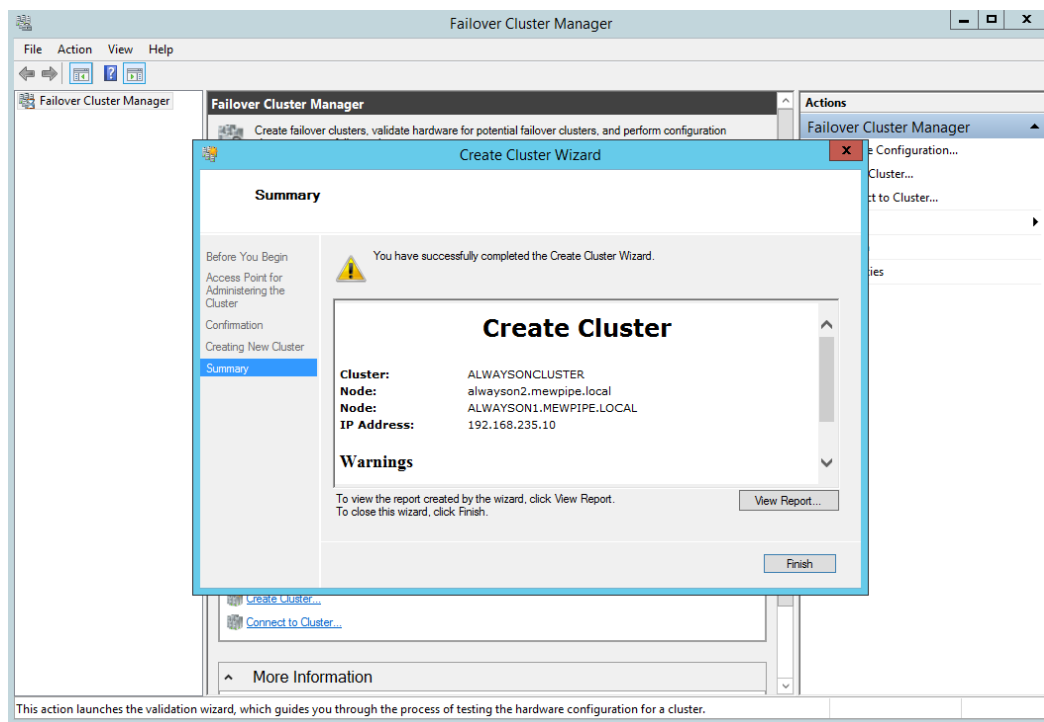
Visuel de la création du cluster

Définition de l'adresse IP du cluster



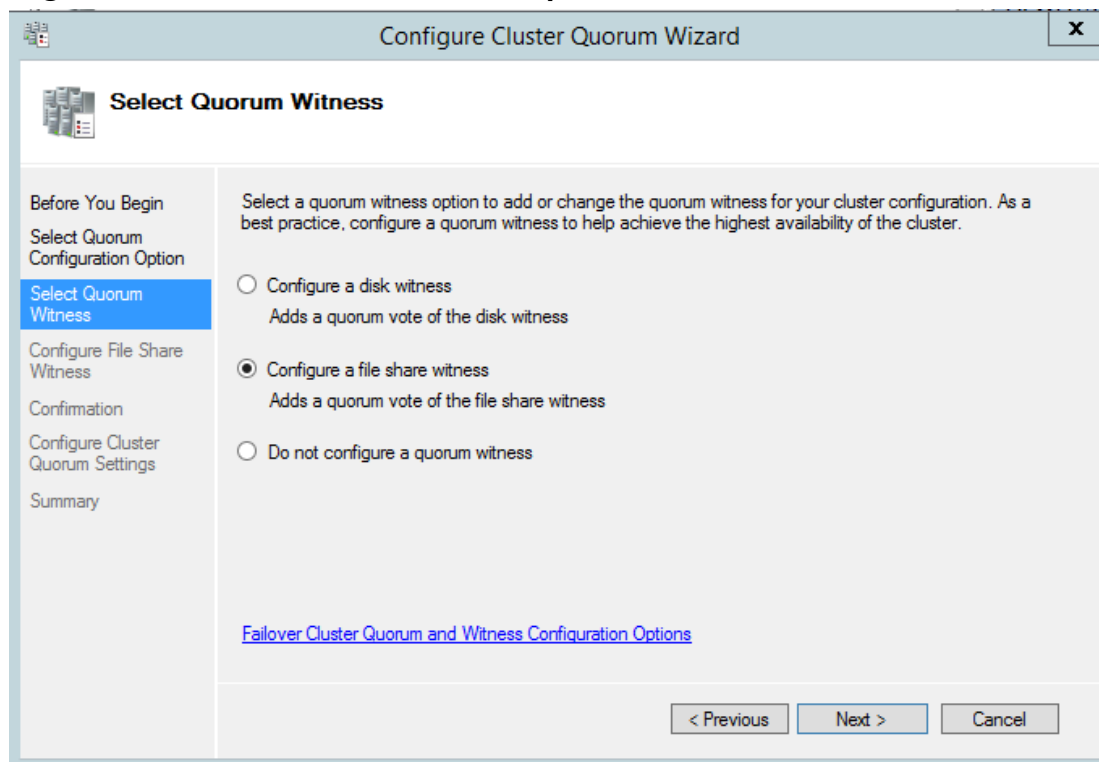
Visuels de la configuration IP du cluster

Validation de la création du cluster



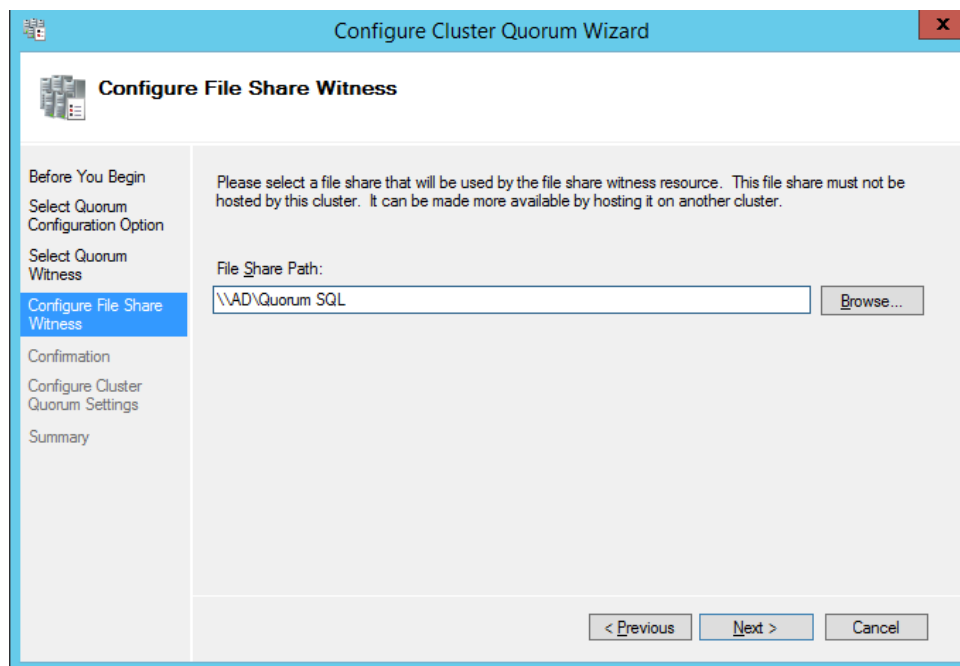
Visuel de la confirmation de création

Configuration du file share witness qui est le Quorum du cluster



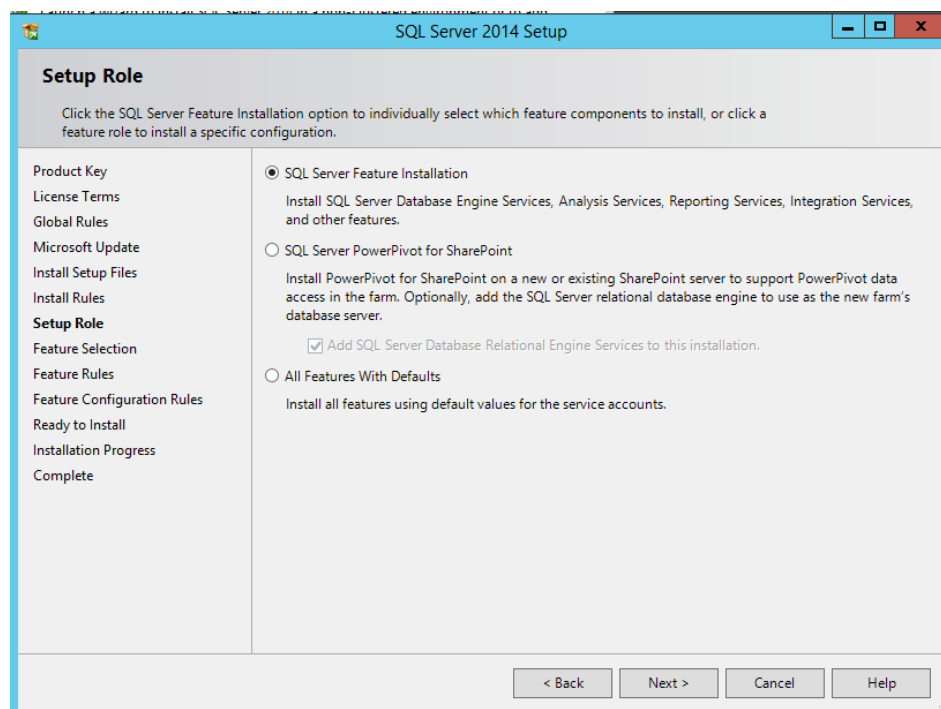
Visuel de la création du share witness

Celui-ci doit être un dossier partagé sur un autre serveur.



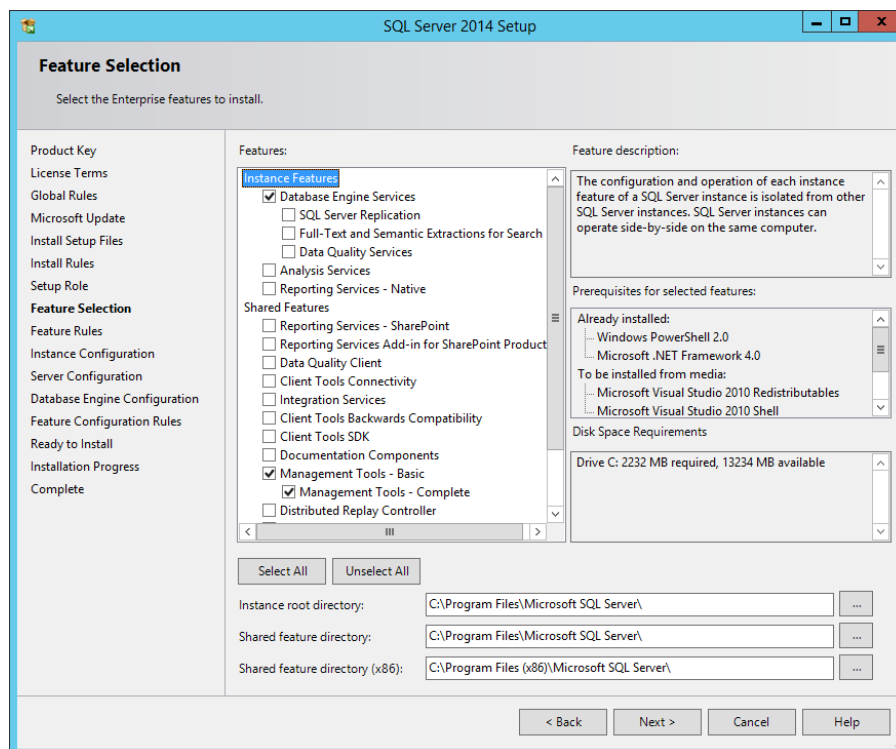
Visuel du choix du chemin du share witness

Installation de SQL Server



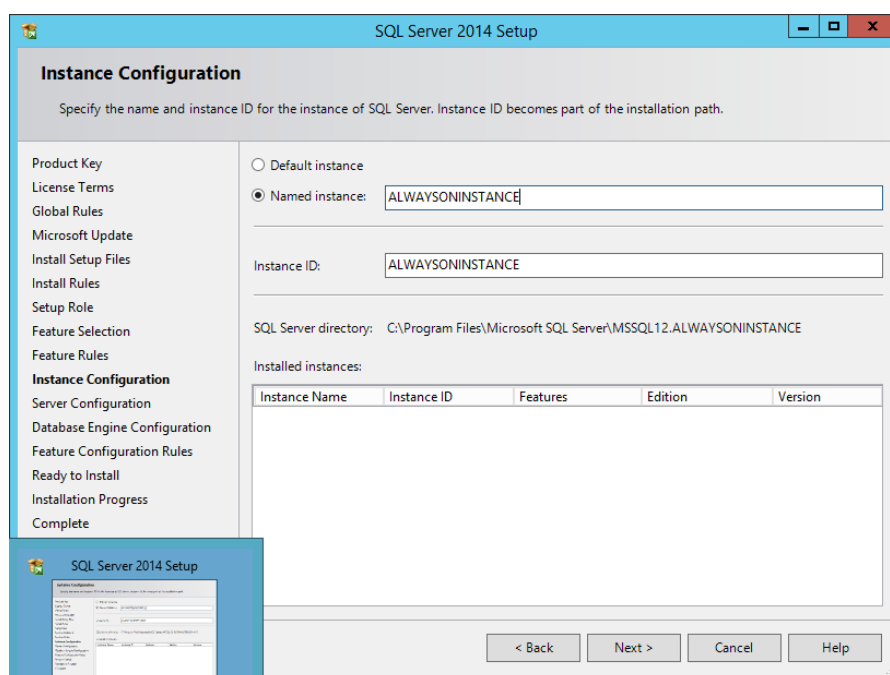
Visuel de l'installateur de SQL Server

Sélectionnez les fonctionnalités à installer comme dans le visuel ci-dessous.



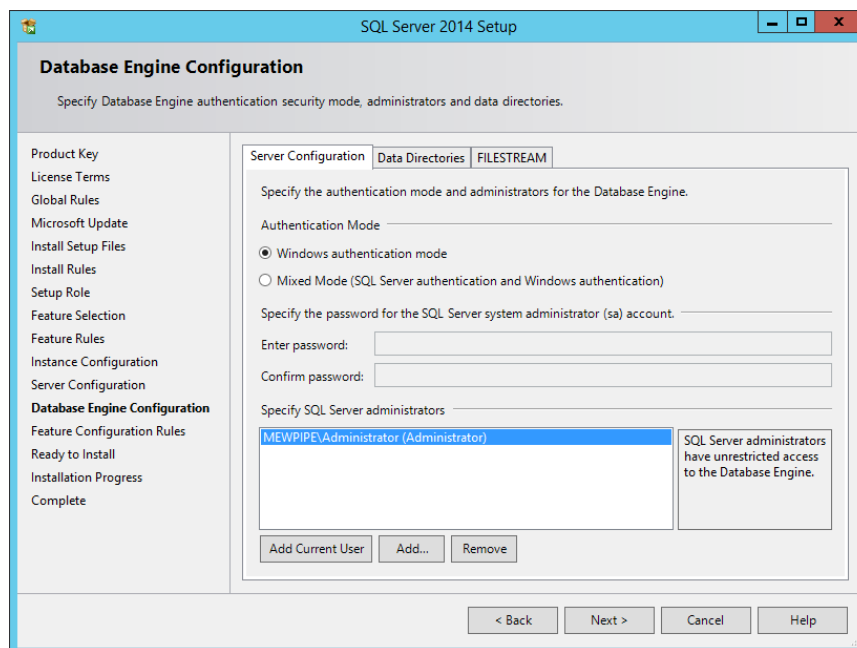
Visuel des fonctionnalités à sélectionner lors de l'installation

Choix du nom de l'instance



Visuel de la définition du nom de l'instance

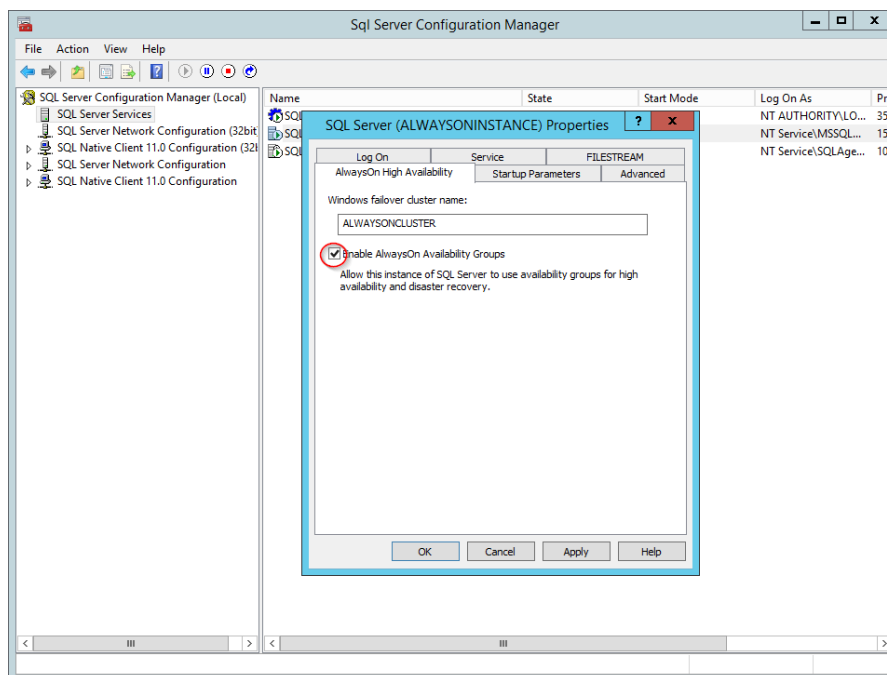
Ajout des comptes administrateurs



Visuel de l'ajout des comptes

Activation de AlwaysOn Availability

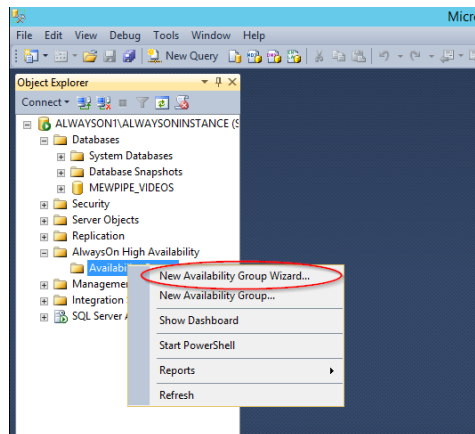
Cocher la case Enable AlwaysOn Availability Groups dans SQL Server Configuration Manager sur votre instance.



Visuel de la configuration de la haute disponibilité

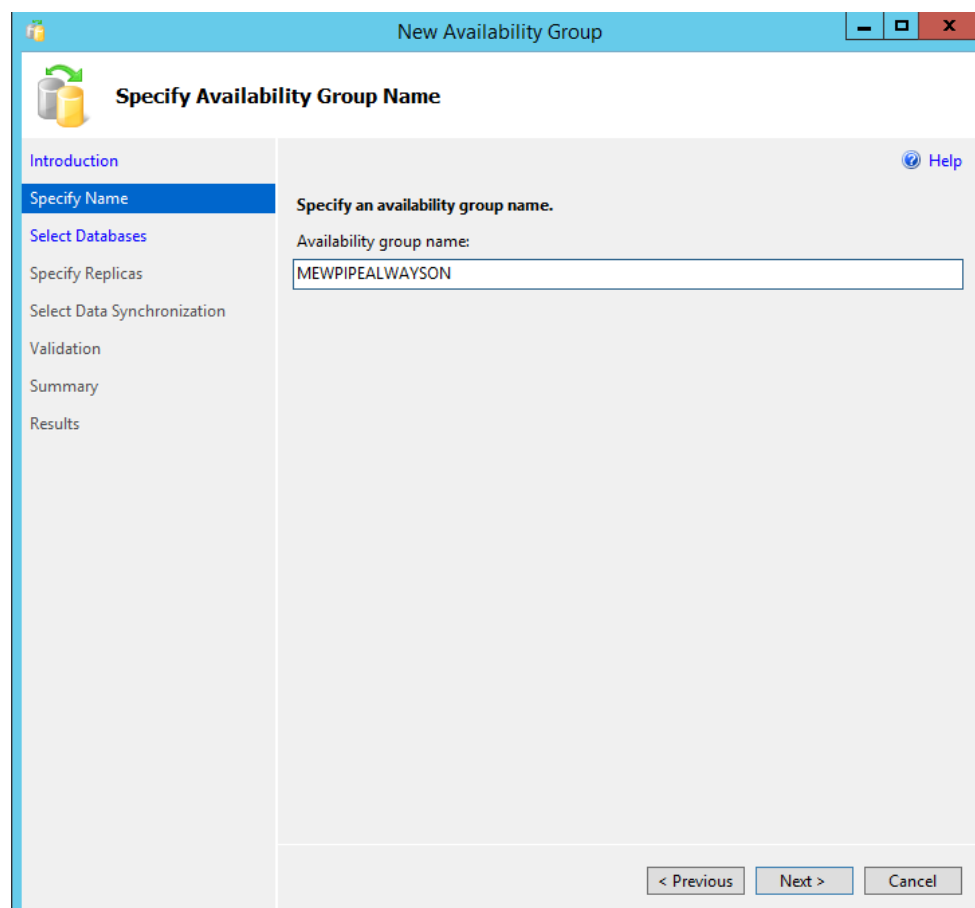
Création de l'Availability Group

Lancer SQL Server Management Studio et dans AlwaysOn High-Availability, cliquez sur New Availability Group Wizard.



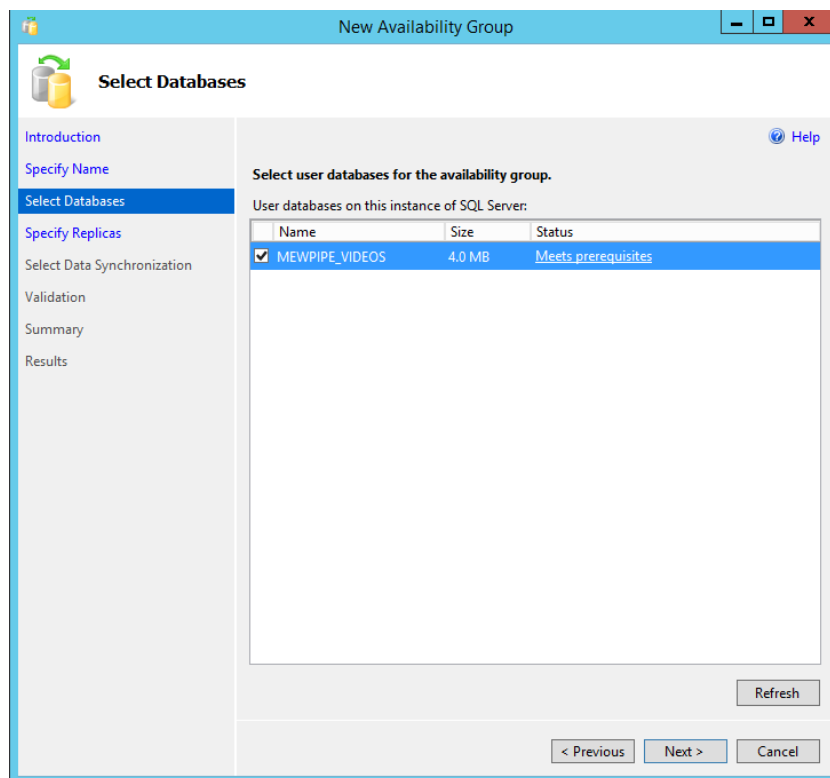
Visuel de la création de l'Availability Group

Spécification du nom du Availability Group



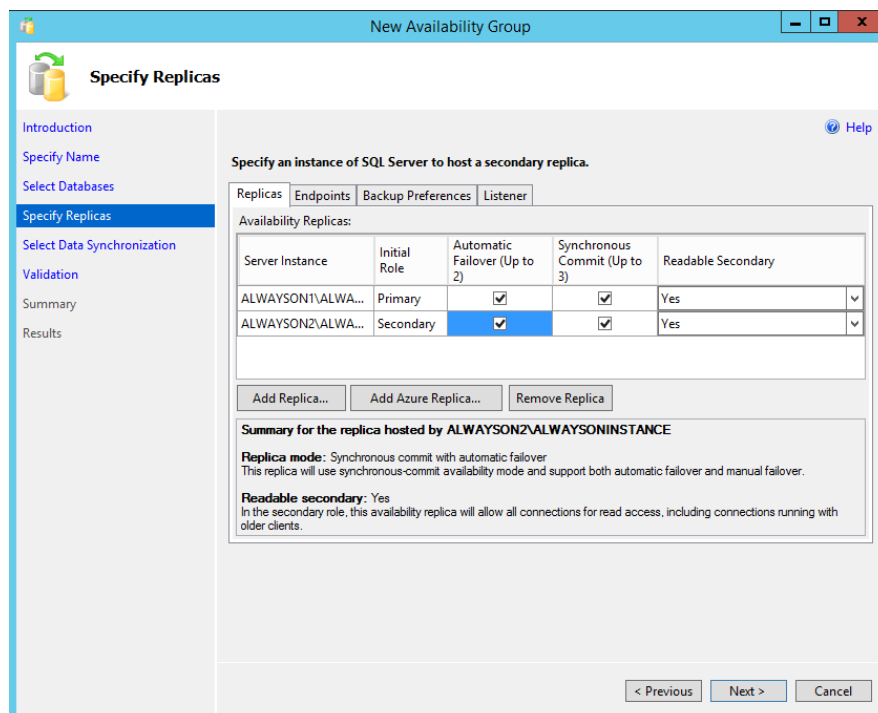
Visuel de la définition du nom de l'Availability Group

Sélection des bases de données hautement disponibles



Visuel de la sélection des bases de données hautement disponibles

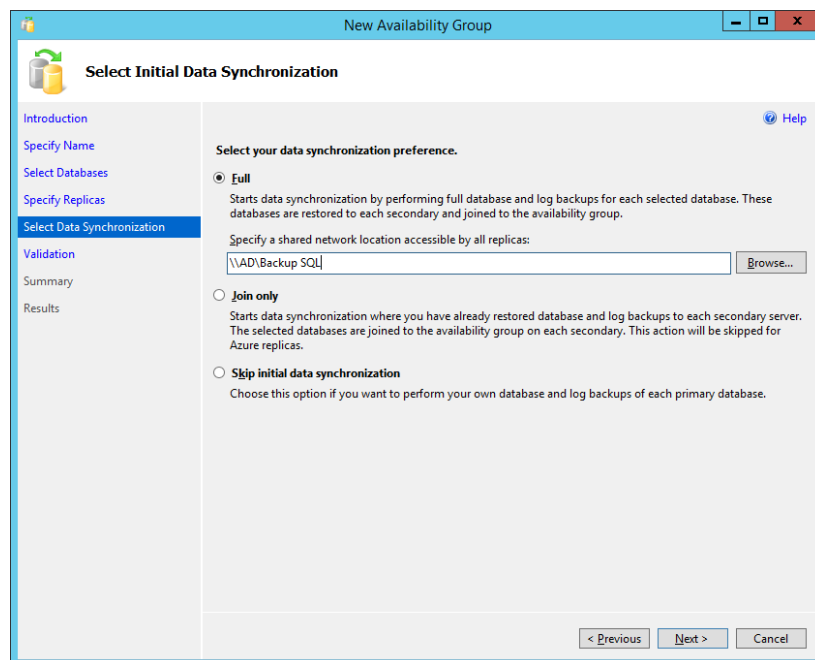
Spécification des noeuds du groupe



Visuel de la sélection des noeuds

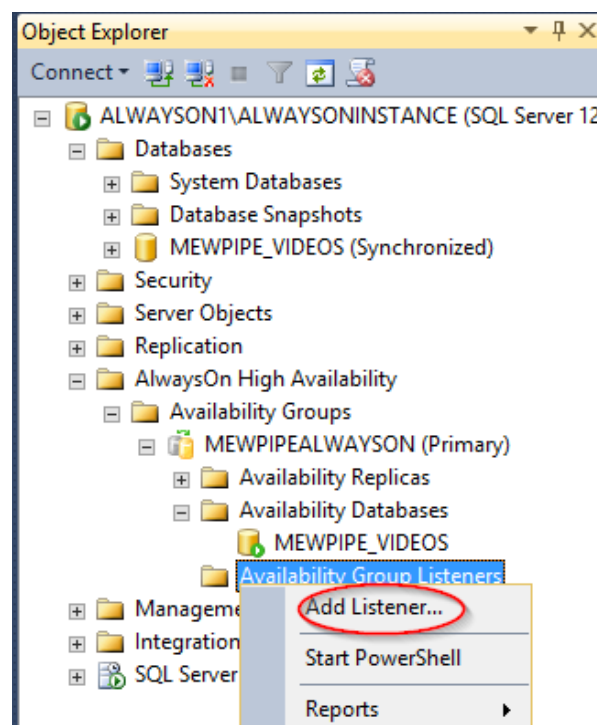
Choix du dossier partagé pour les backups

Spécifier un dossier partagé pour les backups des bases de données et votre base de données est prête.



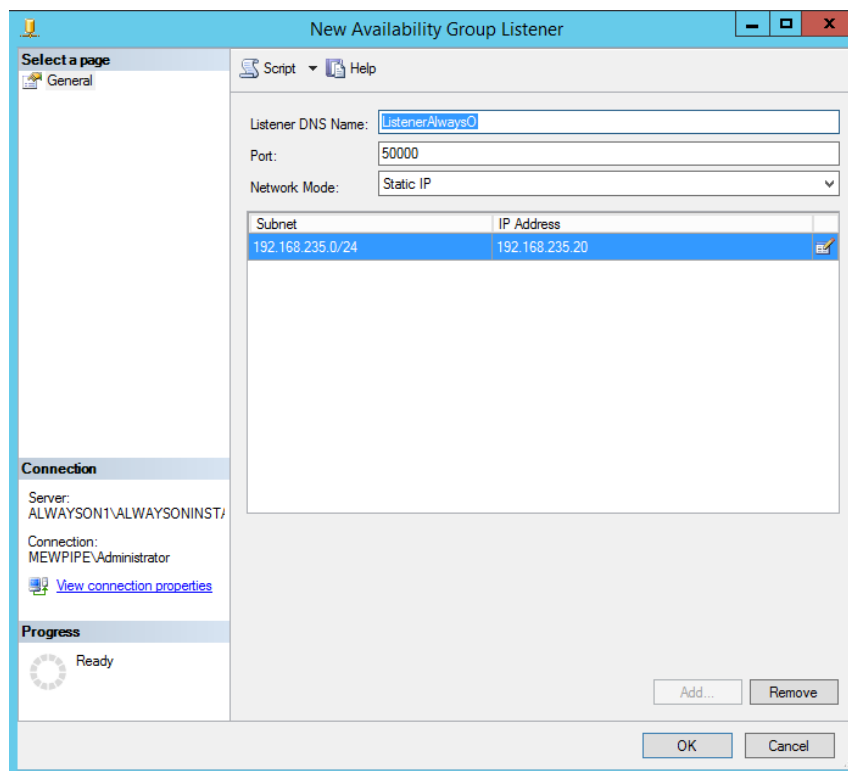
Visuel de la configuration du dossier de backups

Ajout du listener qui va pointer sur l'AlwaysOn

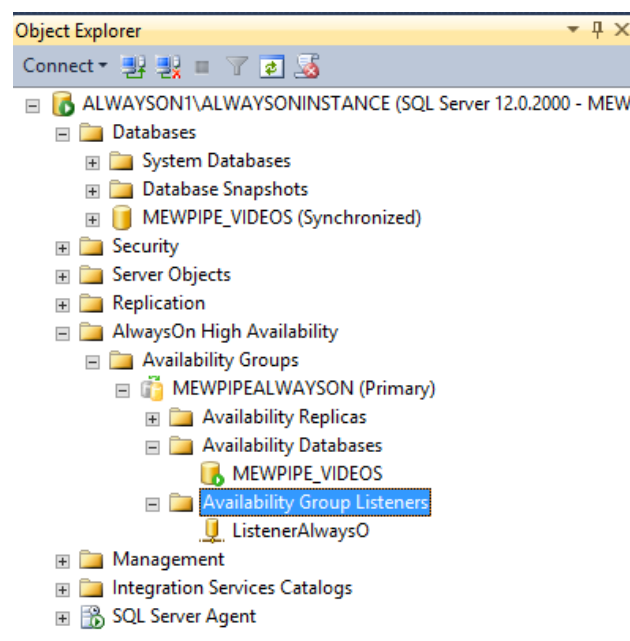


Visuel de la création du listener

Configuration du listener (IP et Nom)



L'instance SQL AlwaysOn est désormais fonctionnelle !



VII. Web Load balancer

Ce document vous permet de comprendre la mise en place du load balancing (répartition des charges) sur nos serveurs web.

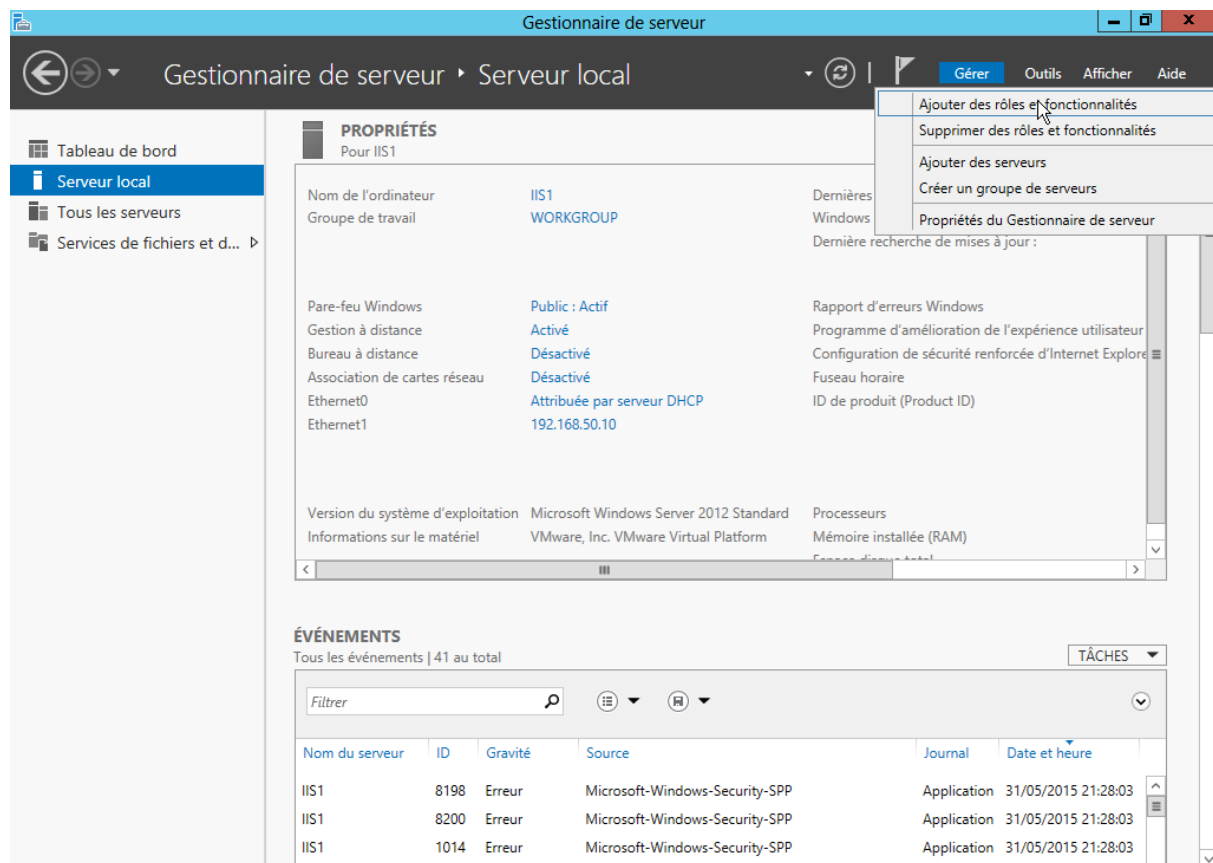
- 2 machines qui ont un rôle de serveur web
- 1 machine qui a un rôle de load balancer

Les trois machines ont 2 cartes réseaux (une pour internet et une pour communiquer avec les autres VMs). Nous utilisons pour ces 3 machines Windows Server 2012.

Mise en place des serveurs web avec IIS

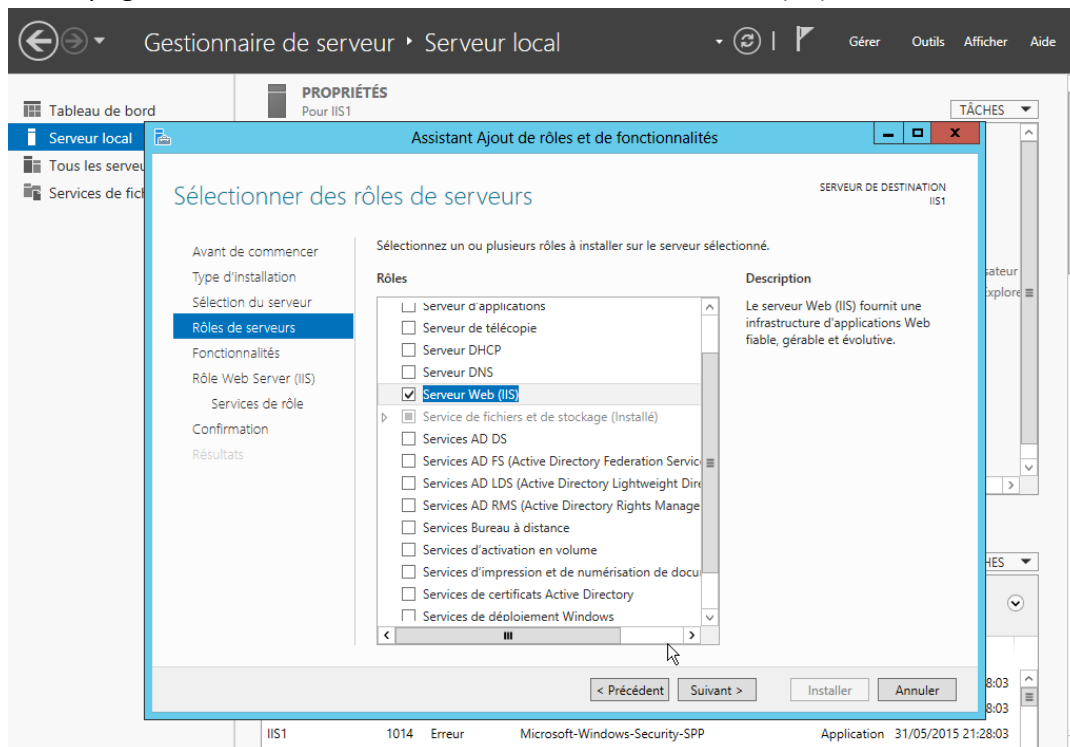
Nous allons mettre en place les serveurs web. Pour cela, nous utiliserons le rôle IIS. Voici la démarche à suivre pour l'installation de celui-ci.

Depuis le gestionnaire de serveur, on clique sur gérer puis ajouter des rôles et fonctionnalités.



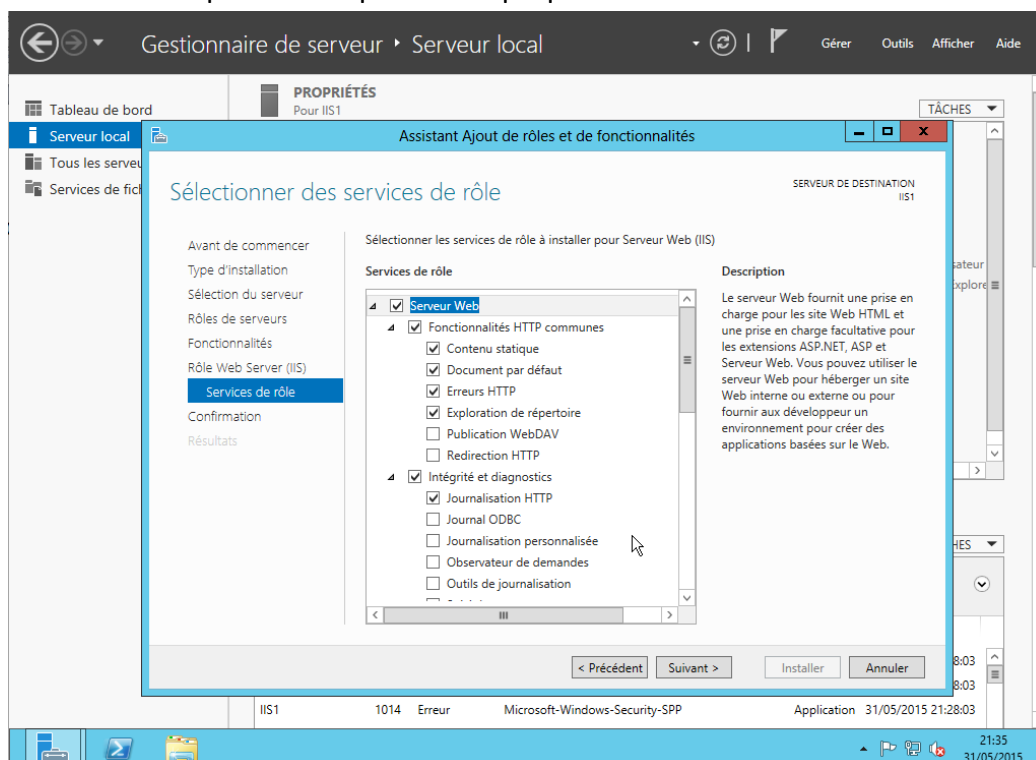
Visuel de l'ajout de rôle et fonctionnalités

Sur la page de sélection des rôles, on coche « Serveur Web (IIS) »



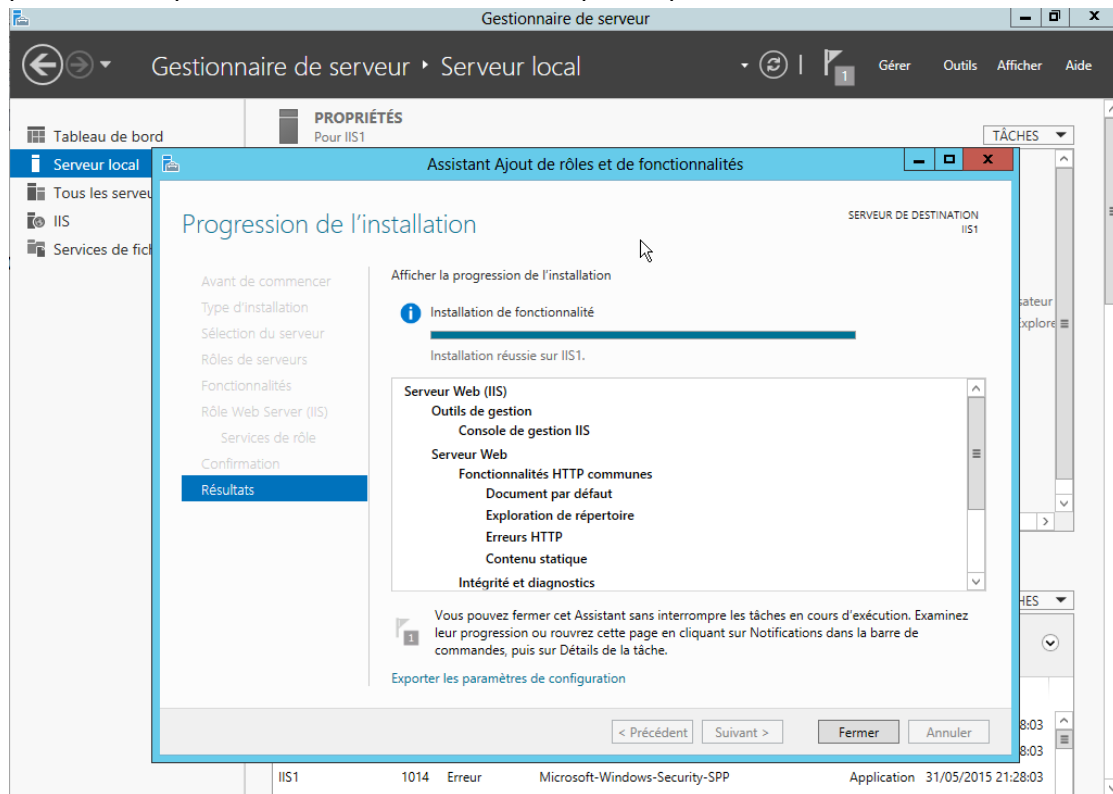
Visuel du choix des rôles

On garde les services par défaut qu'on nous propose.



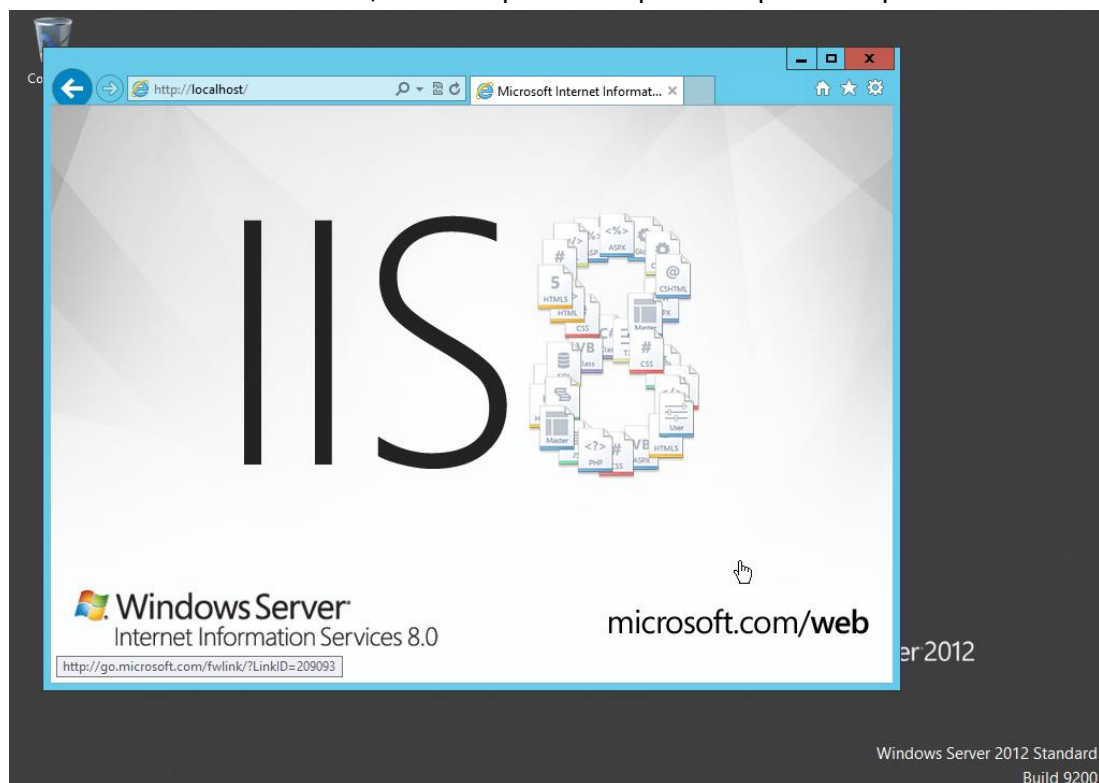
Visuel du choix des services

On passe les étapes en laissant les différentes options par défaut.



Visuel des étapes d'installation

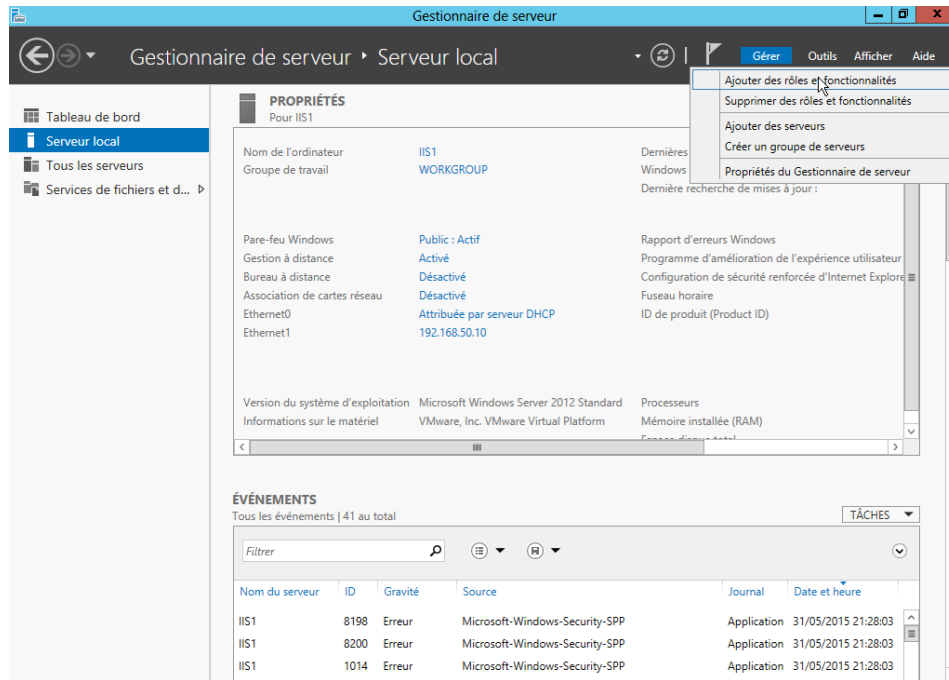
L'installation de IIS est terminée, cette étape est à reproduire pour chaque serveur web.



Visuel de la confirmation d'installation

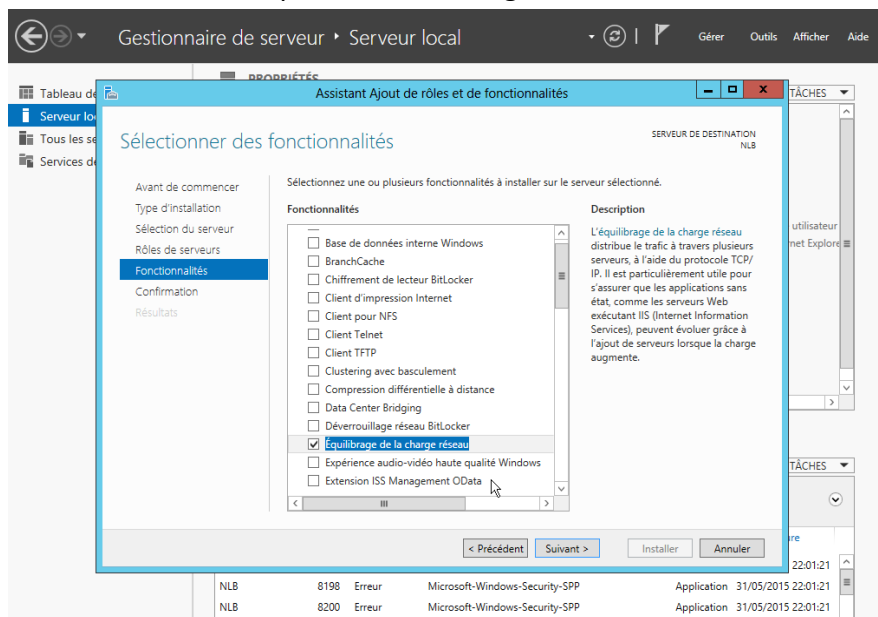
Mise en place de l'équilibrage de la charge réseau

Nous allons maintenant mettre en place sur chaque machine la fonctionnalité « Équilibrage de la charge réseau », c'est cette fonctionnalité qui permettra le load balancing. Depuis le gestionnaire de serveur, on clique sur « ajouter des rôles et fonctionnalités ».



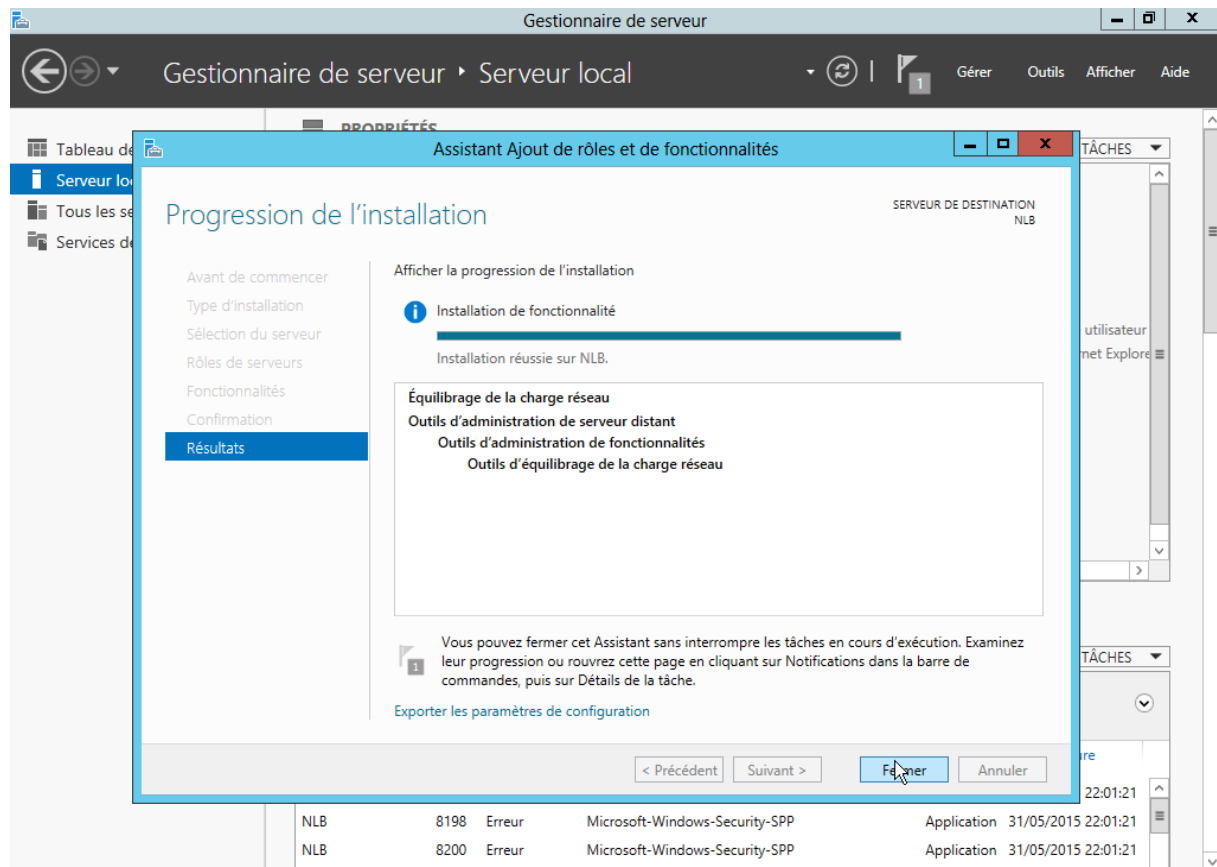
Visuel de l'ajout des rôles

On passe toutes les étapes sans modifier quoi que ce soit. Lorsqu'on arrive sur la page des fonctionnalités, on coche sur « Équilibre de la charge réseau ».



Visuel du choix des fonctionnalités

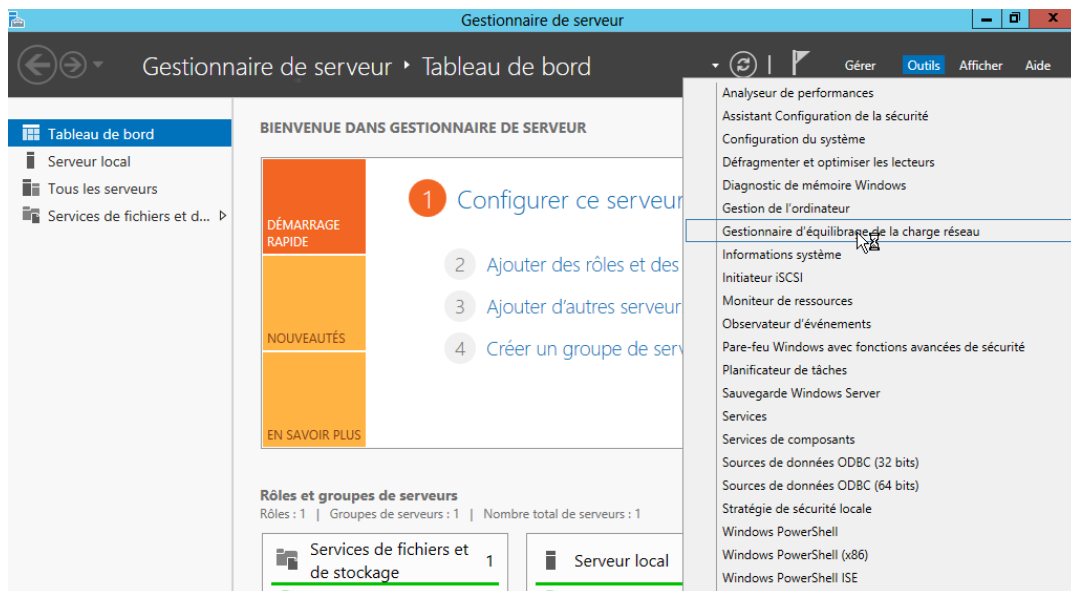
On passe les étapes et on finit l'installation. Ceci est à effectuer sur les serveurs web et sur le serveur qui servira de load balancer.



Mise en place du cluster

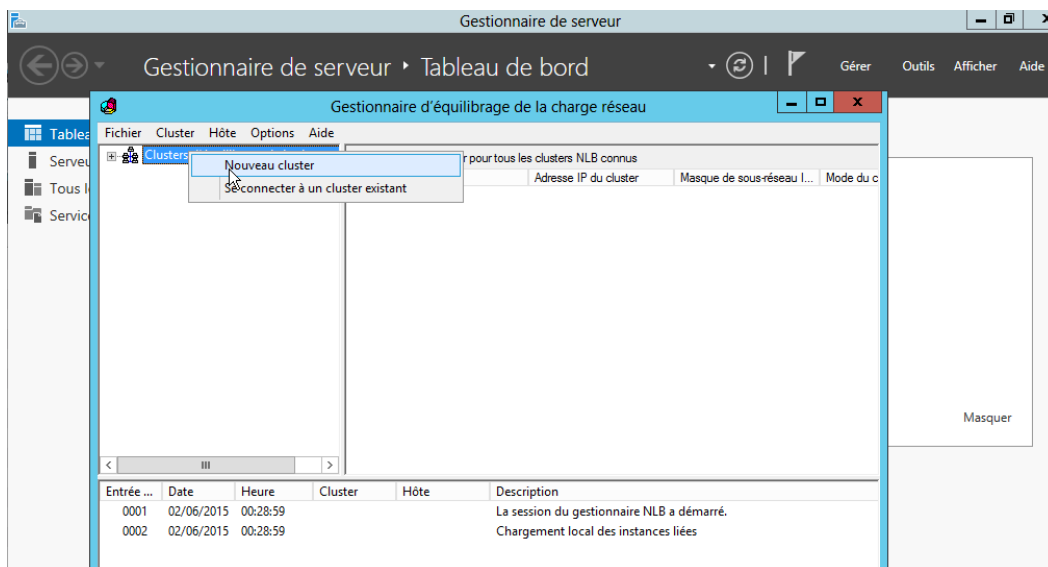
Il ne reste plus qu'à mettre en place l'équilibrage sur le load balancer. Pour cela nous allons créer un cluster contenant nos deux serveurs web.

Pour cela sur notre load balancer, on va sur le gestionnaire de serveur puis on clique sur « Outils » et « Gestionnaire d'équilibrage de la charge réseau ».



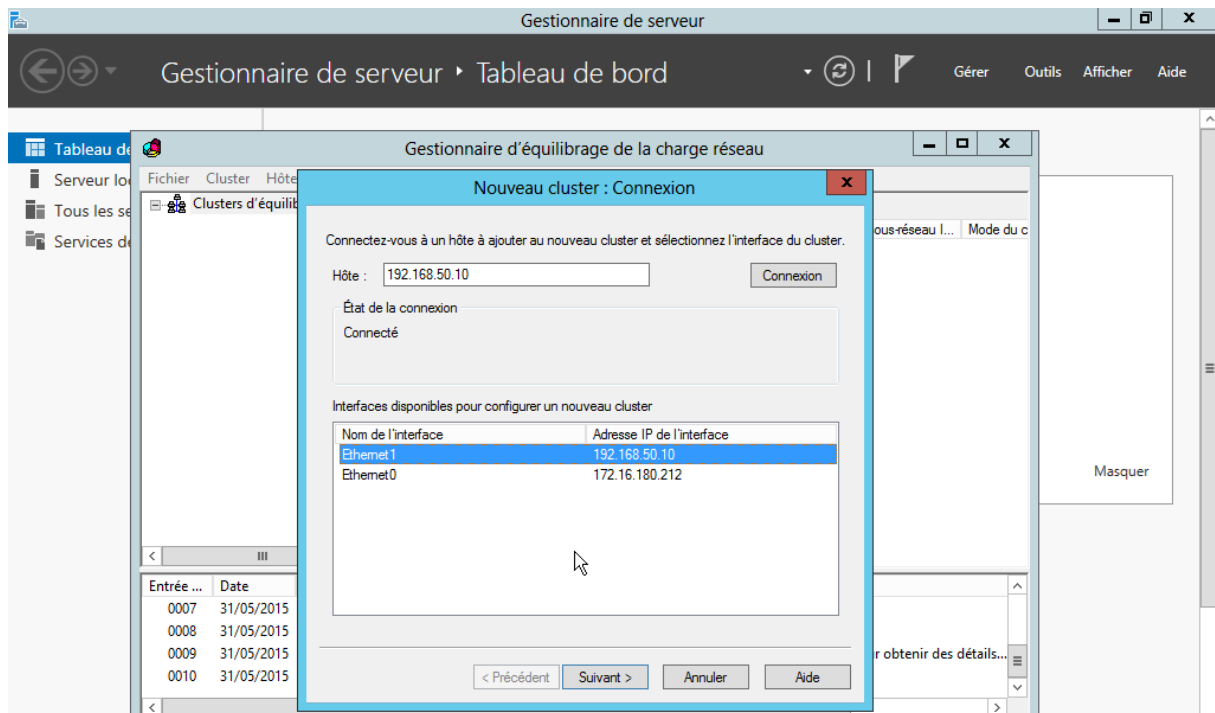
Visuel de l'accès à l'équilibrage de la charge réseau

On fait un clic droit sur « Clusters d'équilibrage de la charge réseau » puis « nouveau cluster ».



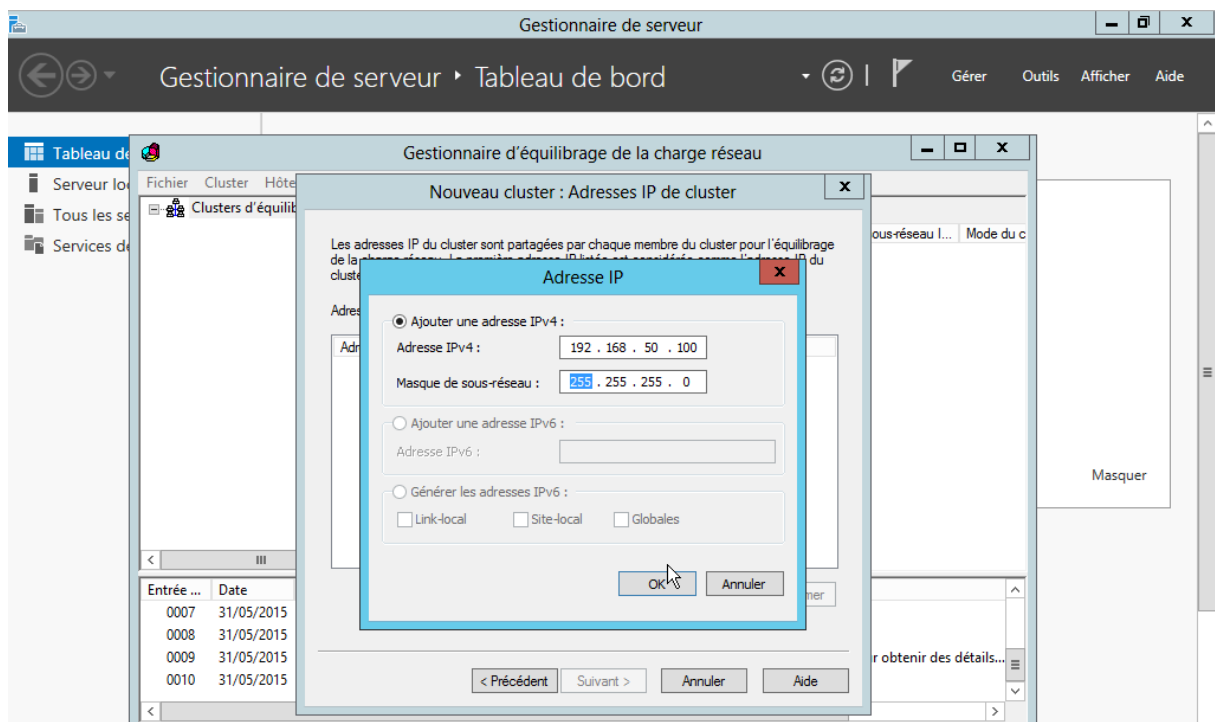
Visuel de l'ajout du cluster

On va ajouter les différents hôtes au cluster ici, on entre l'IP du premier serveur IIS puis on coche « Ethernet 1 », c'est la carte réseau utilisée pour communiquer dans le réseau.



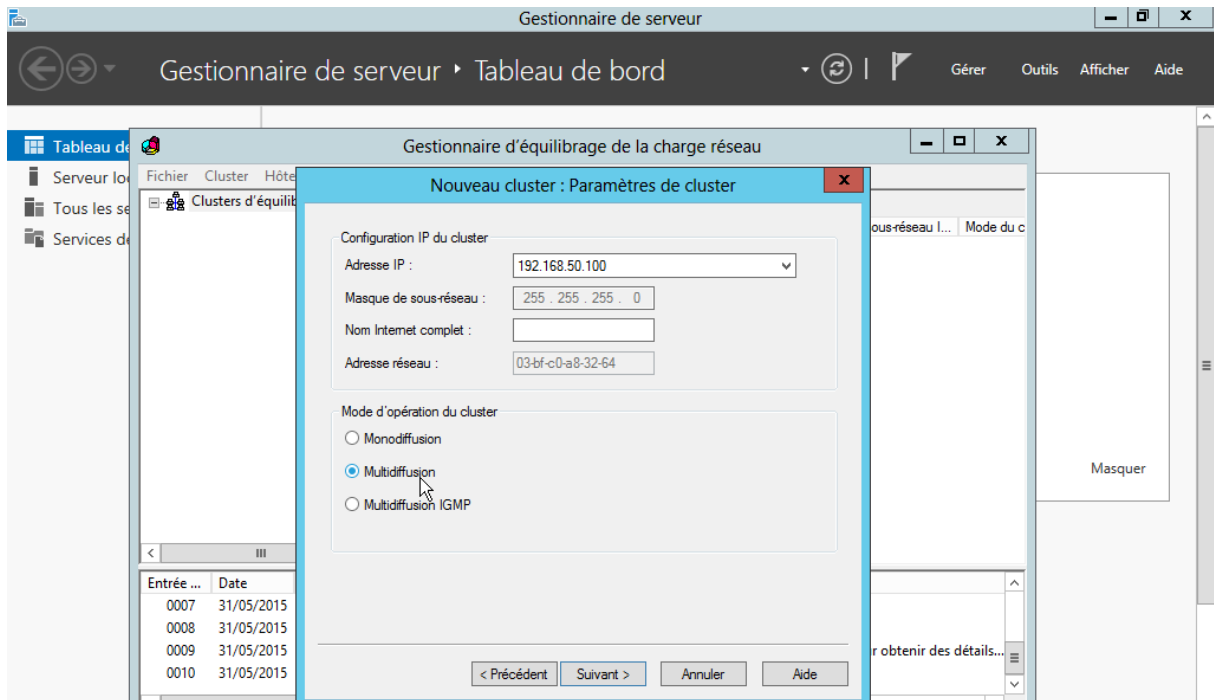
Visuel de l'ajout des hôtes au cluster

Ensuite on entre l'IP qu'on veut donner à notre cluster et on choisit son masque de sous-réseau.



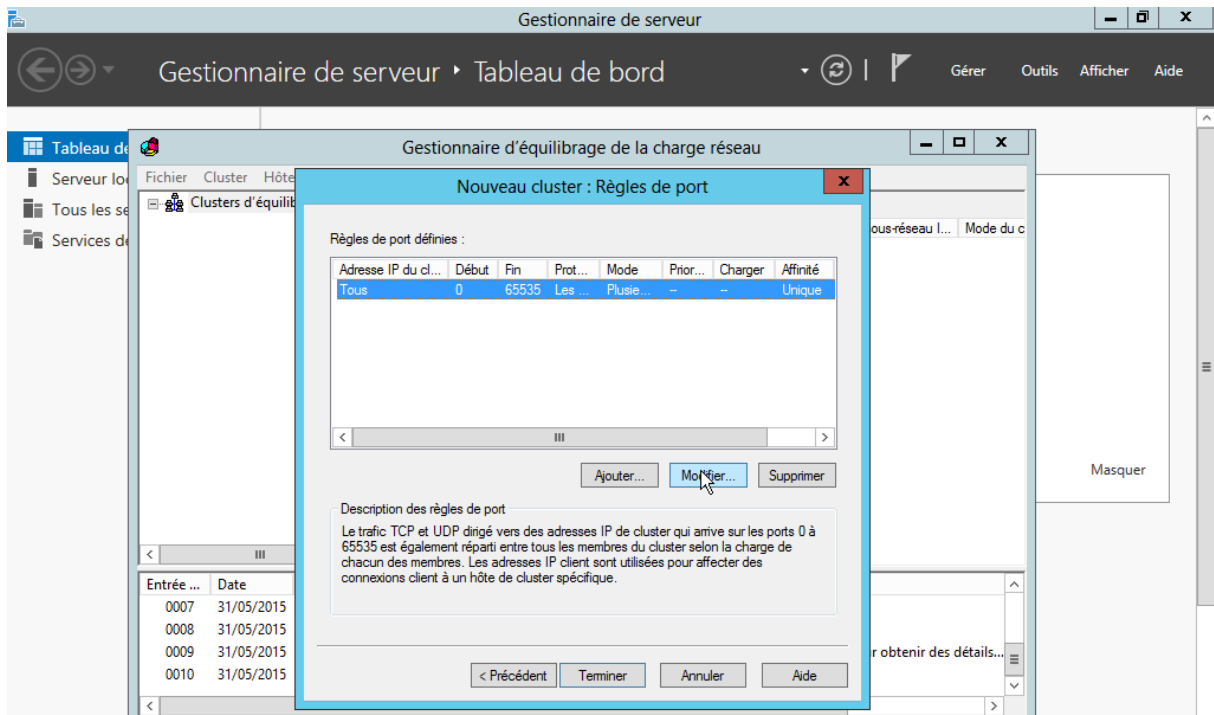
Visuel de la configuration IP

Par la suite on choisit le mode « Multidiffusion ».



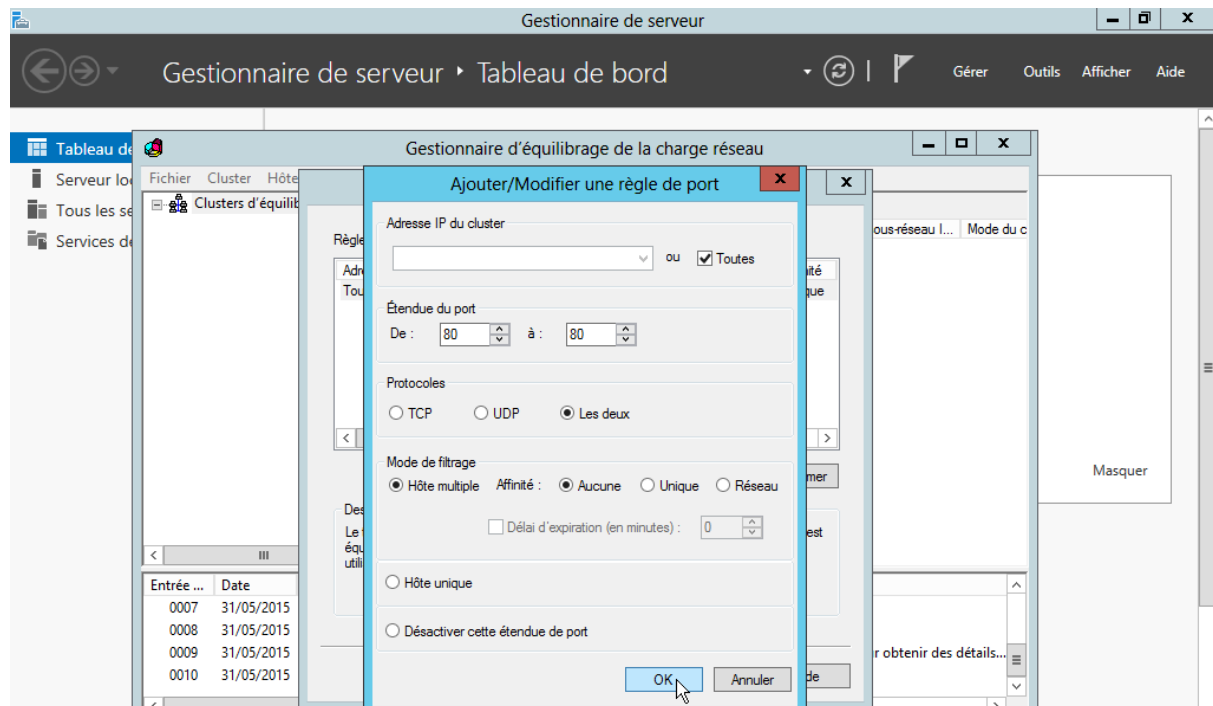
Visuel de la configuration Multidiffusion

Enfin, on choisit les règles du port, ici nous allons entrer les 2 ports utilisés principalement pour le web, le port 80 (http) et le port 443 (https)



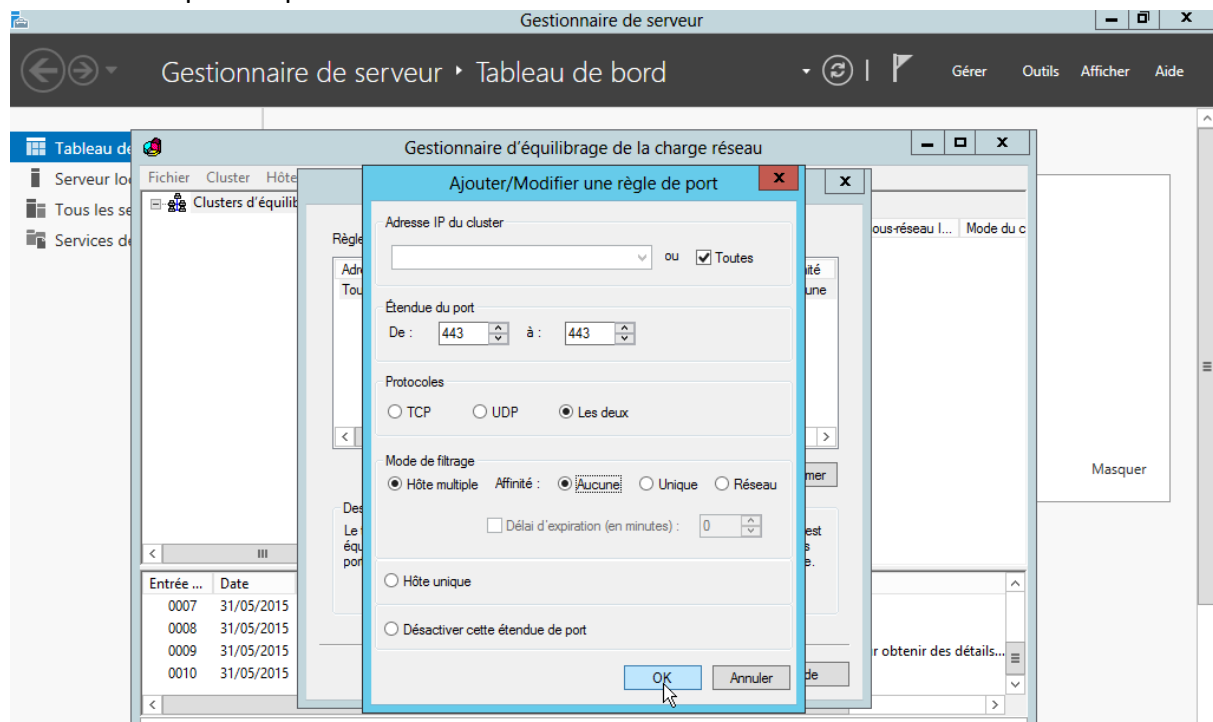
Visuel de la configuration des règles de port

Lors de l'ajout de la règle de port, on coche protocoles sur « les deux » et ne met aucun mode de filtrage.



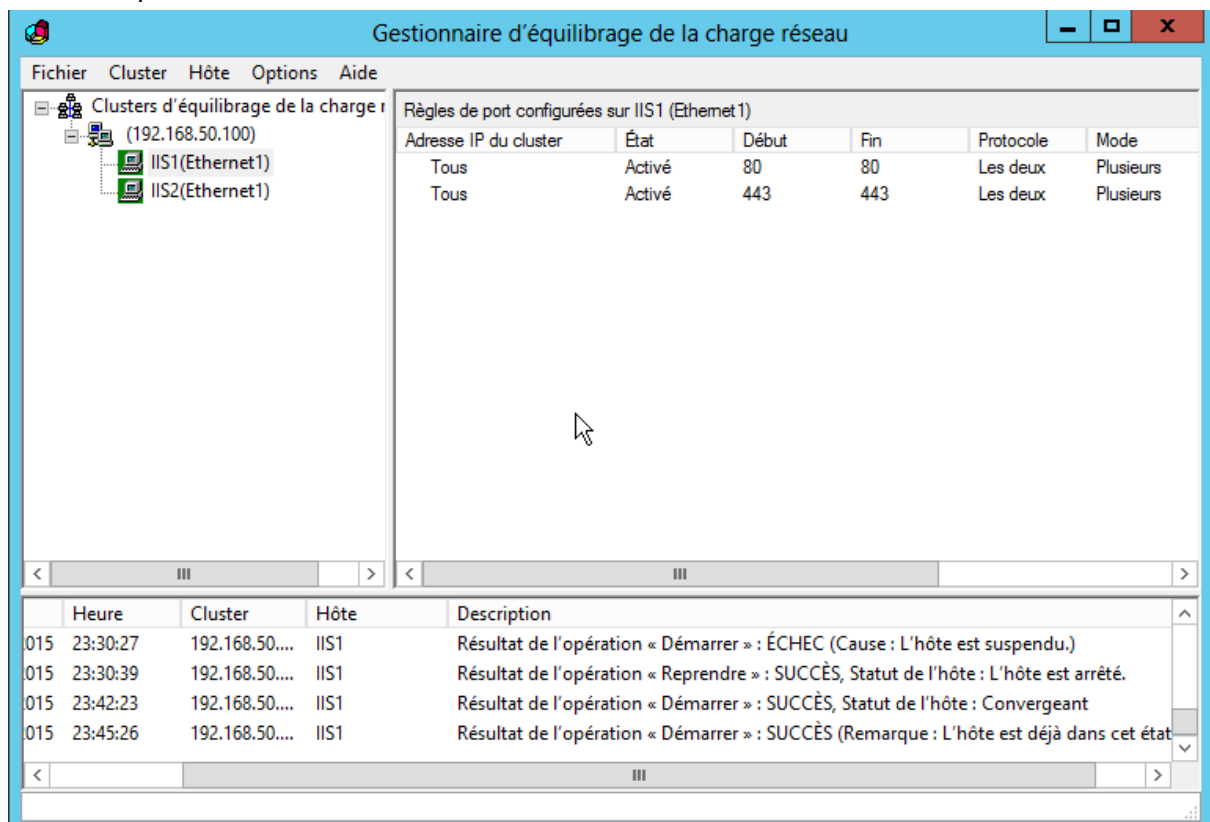
Visuel de la configuration de la règle pour le port 80

Même chose pour le port 443.



Visuel de la configuration de la règle pour le port 443

Nos deux hôtes sont bien connectés au cluster, la mise en place est réussie ! Cependant vérifions que tout fonctionne correctement.



Visuel de la configuration terminée

Démonstration

Nous allons tester le load balancing, pour cela on modifie la page d'accueil des deux serveurs web afin de les différencier.

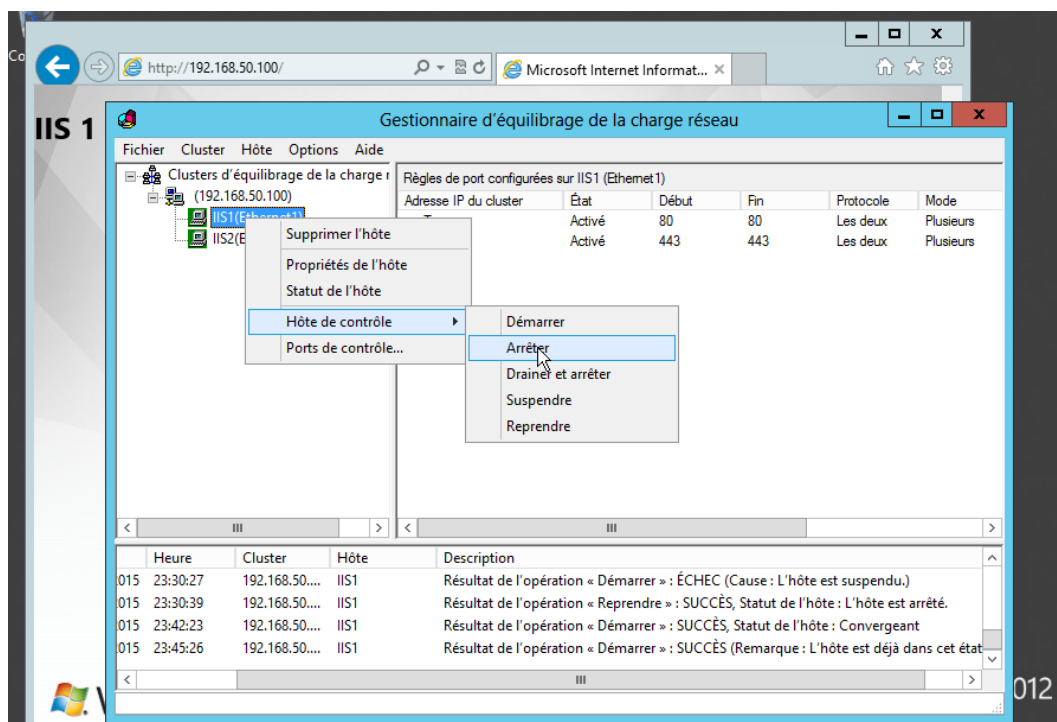
Nous avons tout simplement rajouté une balise h2 sur la page d'accueil.



Visuel de la page modifiée

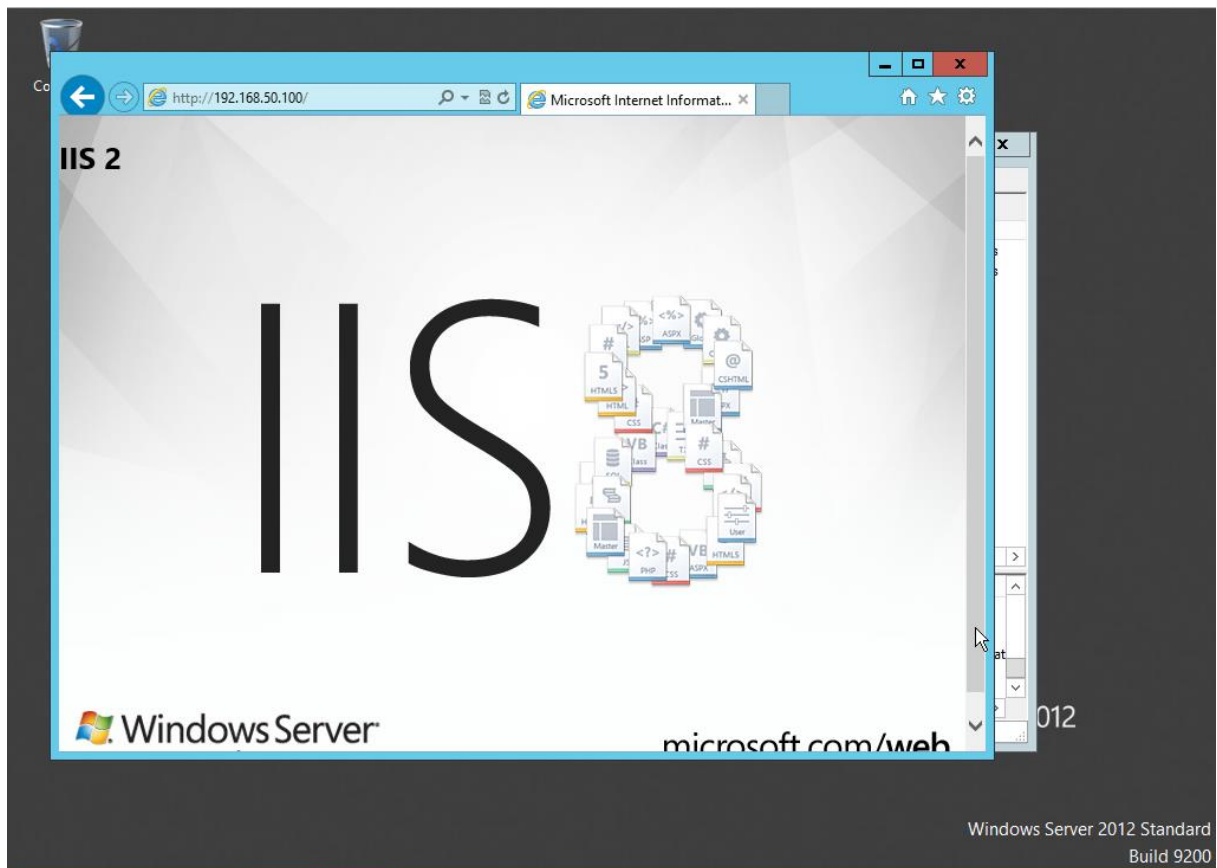
Nous avons maintenant testé notre cluster. La conclusion est que par défaut lorsque les deux serveurs web sont disponibles, notre cluster choisi notre premier serveur.

On va donc arrêter la liaison du premier serveur web (IIS1) avec le cluster pour vérifier qu'il se connecte bien sur le deuxième serveur.



Visuel de l'arrêt de l'hôte

C'est bien le cas, le load balancing fonctionne !

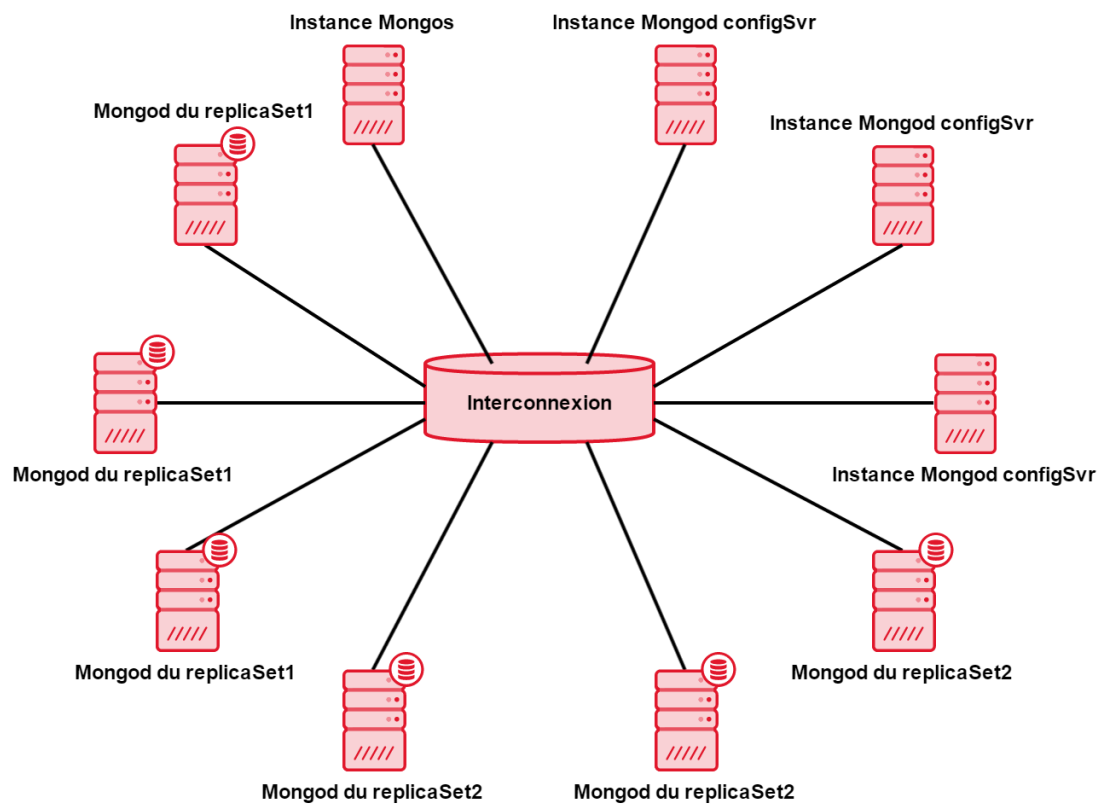


Visuel de la confirmation de changement d'hôte

VIII. MongoDB

Dans cette section nous allons étudier la mise en place de MongoDB et de sa configuration en haute disponibilité et réplication.

Architecture



Prérequis par machine

- Windows Server 2012 R2
- CPU 4 cores
- 8 go de ram
- 30 go OS
- Connecté sur SAN extensible pour données

Installation de MongoDB sur Windows Server 2012 R2

Lanceur l'installateur téléchargé au préalable sur <http://www.mongodb.org/downloads>.

Choisir son dossier de destination, ce dossier sera l'endroit où les commandes mongoDB seront lancées.

Créer un dossier dans lequel vous mettrez la base de données. Par exemple C:\data\db.

Pour instancier une première base de données dans ce dossier, lancez cette commande (se positionner dans le dossier d'installation dans l'invite de commande) :

```
mongod --dbpath "C:\data\db"
```

Pour effectuer des opérations sur la base de données :

```
mongo
```

Mise en place du Replica Set

Tout d'abord, il faut instancier les base de données dans un Replica Set (à effectuer sur les différentes machines):

```
mongod --replSet "rs0"
```

Connectez-vous sur les différentes instances avec cette commande :

```
mongo
```

Initialisez la configuration par défaut du réplica set :

```
rs.initiate()
```

Pour ajouter une machine dans le replica set il faut (n'oubliez pas d'instancier chaque base de données dans le même replica set) :

```
rs.add("mongoDB1")  
rs.add("mongoDB2")  
rs.add("mongoDB3")
```

Pour vérifier votre configuration de replica set :

```
rs.status()
```


Mise en place du Sharding

Dans un premier temps, créer un dossier dans lequel les fichiers de configuration du cluster seront mis en place.

Lancer les trois instances de configuration du cluster sur vos 3 serveurs, le port 27019 étant le port par défaut :

```
mongod --configsvr --dbpath C:\data\confDB --port 27019
```

Il faut maintenant lancer le service de routing de mongoDB sur une des machines :

```
mongos --configdb mongoDBcf1:27019,mongoDBcf2:27019,mongoDBcf3:27019
```

Enfin, il faut ajouter les shards à notre configuration, nous allons donc nous connecter sur notre instance de routing sharding via mongo :

```
mongo --host mongoShard --port 27017
```

Puis ajouter les membres (ici nos replica set) :

```
sh.addShard( "rs0/mongoDB1:27017" )  
sh.addShard( "rs1/mongoDB4:27017" )
```

Il ne reste plus qu'à ajouter les collections à « Sharder ».

```
sh.enableSharding("mewpipeStorage")
```

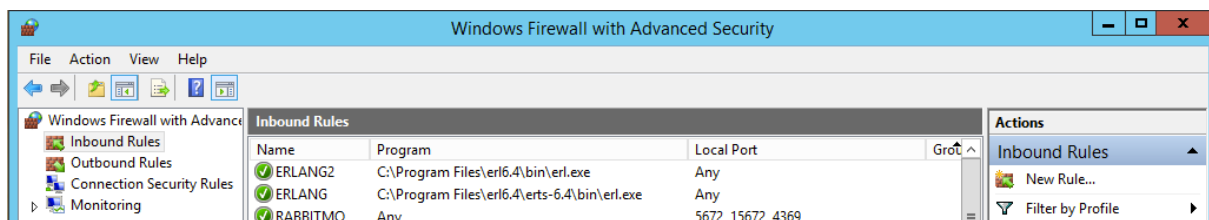
IX. RabbitMQ

Dans cette partie nous allons voir comment configurer RabbitMQ en cluster et en haute disponibilité. Pour rappel, MongoDB nous sert à faire parvenir des messages dans notre système, notamment pour traiter des tâches lourdes en arrière plan.

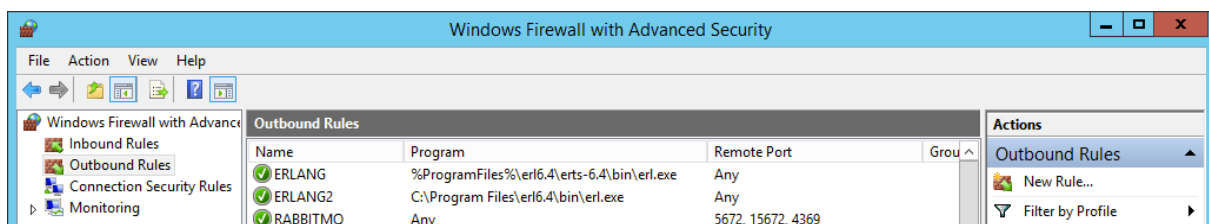
Configuration du pare-feu

Avant de commencer, il est nécessaire d'assurer que tous les membres du futur cluster aient leur pare-feu configuré de la sorte :

- Ports 5672, 15672, 4369 en TCP ouverts en entrée et en sortie
- Programme "C:\Program Files\erl6.4\erts-6.4\bin\erl.exe" autorisé en entrée et en sortie
- Programme "C:\Program Files\erl6.4\bin\erl.exe" autorisé en entrée et en sortie



Règles de pare-feu entrantes



Règles de pare-feu sortantes

Installation sur un premier serveur

La seconde étape est de choisir un serveur qui initialisera la procédure.

Une fois ce serveur choisi, installez les programmes suivants en tant qu'administrateur sur le serveur :

- Erlang - <http://www.erlang.org/>
- RabbitMQ - <https://www.rabbitmq.com/>

Une fois ces programmes installés, ouvrez une interface de ligne de commande, et déplacez-vous dans le dossier suivant (remplacez le numéro de version si vous n'avez pas le même) :

```
C:\Program Files (x86)\RabbitMQServer\rabbitmq_server-3.5.3\sbin
```

Une fois dans ce dossier, nous allons passer le serveur RabbitMQ en mode cluster en saisissant la commande suivante :

```
rabbitmq-server -detached
```

RabbitMQ est prêt pour le cluster, il a également initialisé un “cookie Erlang” sur le serveur. Ce cookie doit être le même sur tous les noeuds du cluster, afin de préparer ceci, copiez le fichier suivant (sur le premier serveur) :

```
C:\Windows\.erlang.cookie
```

Vers les dossiers suivants (sur les autres serveurs que nous n'avons pas encore préparés) :

```
C:\Windows\.erlang.cookie
```

```
%HOMEPATH%\erlang.cookie
```

La configuration du premier serveur est dorénavant terminée, nous allons passer à la suite.

Installation sur les autres serveurs

Les étapes suivantes seront à effectuer sur chaque serveur (excepté celui que nous avons configuré précédemment).

Installez les programmes suivants en tant qu'administrateur sur les serveurs :

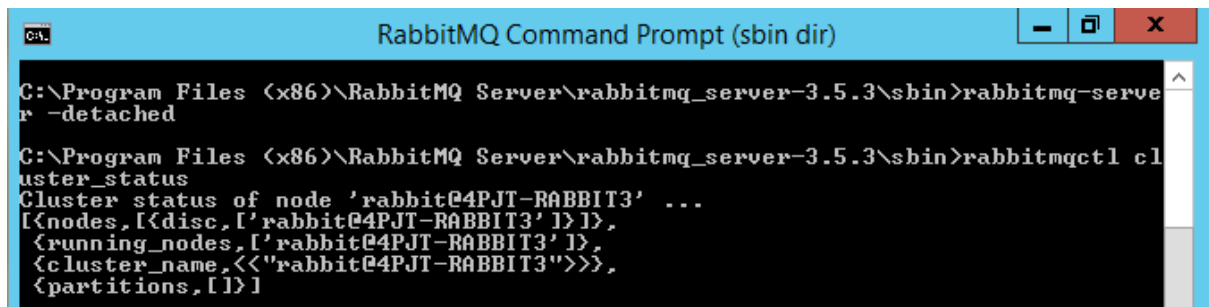
- Erlang - <http://www.erlang.org/>
- RabbitMQ - <https://www.rabbitmq.com/>

Une fois ces programmes installés, ouvrez une interface de ligne de commande, et déplacez vous dans le dossier suivant (remplacez le numéro de version si vous n'avez pas le même) :

```
C:\Program Files (x86)\RabbitMQServer\rabbitmq_server-3.5.3\sbin
```

Une fois dans ce dossier, nous allons passer le serveur RabbitMQ en mode cluster en saisissant la commande suivante :

```
rabbitmq-server -detached
```



```
C:\Program Files (x86)\RabbitMQ Server\rabbitmq_server-3.5.3\sbin>rabbitmq-server -detached

C:\Program Files (x86)\RabbitMQ Server\rabbitmq_server-3.5.3\sbin>rabbitmqctl cluster_status
Cluster status of node 'rabbit@4PJT-RABBIT3' ...
[{nodes,[{disc,['rabbit@4PJT-RABBIT3']}]}],
 {running_nodes,['rabbit@4PJT-RABBIT3']}],
 {cluster_name,<<"rabbit@4PJT-RABBIT3">>}},
 {partitions,[]}]
```

Le noeud est prêt pour le cluster, mais il est pour l'instant seul

Former le cluster

Les étapes suivantes seront à effectuer sur chaque serveur (excepté celui que nous avons configuré au début).

Récupérez le nom d'hôte de votre premier serveur, dans les exemples suivants, le nom d'hôte de notre premier serveur sera **4PJT-RABBIT1**, faites bien attention à le remplacer par le vôtre.

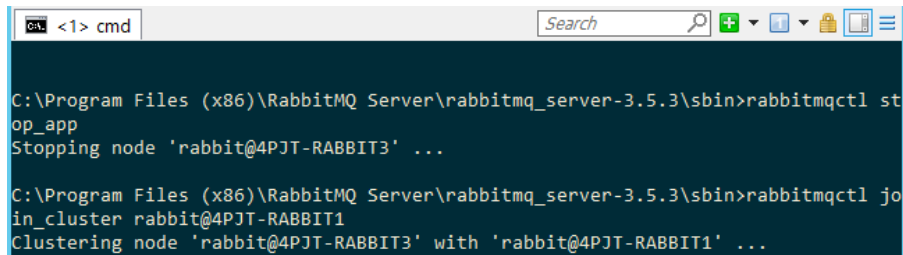
Ouvrez une interface de ligne de commande, et déplacez vous dans le dossier suivant (remplacez le numéro de version si vous n'avez pas le même) :

```
C:\Program Files (x86)\RabbitMQServer\rabbitmq_server-3.5.3\sbin
```

Nous allons arrêter le noeud, l'ajouter au cluster et le redémarrer. Pour cela, saisissez les commandes suivantes :

```
rabbitmqctl stop_app
```

```
rabbitmqctl join_cluster rabbit@4PJT-RABBIT1
```



```
C:\Program Files (x86)\RabbitMQ Server\rabbitmq_server-3.5.3\sbin>rabbitmqctl stop_app
Stopping node 'rabbit@4PJT-RABBIT3' ...

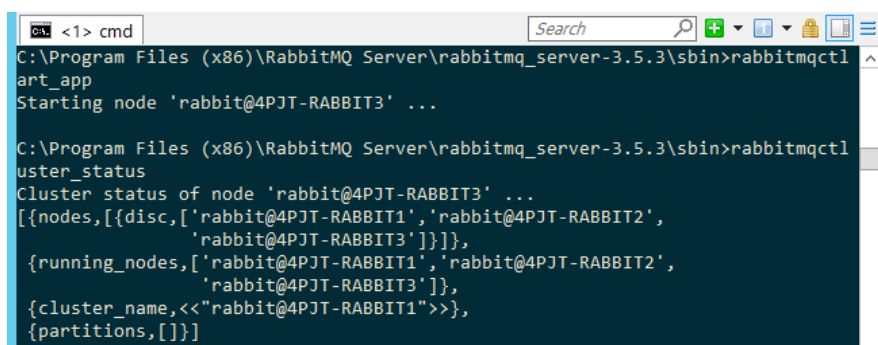
C:\Program Files (x86)\RabbitMQ Server\rabbitmq_server-3.5.3\sbin>rabbitmqctl join_cluster rabbit@4PJT-RABBIT1
Clustering node 'rabbit@4PJT-RABBIT3' with 'rabbit@4PJT-RABBIT1' ...
```

Résultat de la commande, nous allons tester plus ensuite

A présent, redémarrons le noeud et vérifions son ajout au cluster. Pour cela, saisissez les commandes suivantes :

```
rabbitmqctl start_app
```

```
rabbitmqctl cluster_status
```



```
C:\Program Files (x86)\RabbitMQ Server\rabbitmq_server-3.5.3\sbin>rabbitmqctl start_app
Starting node 'rabbit@4PJT-RABBIT3' ...

C:\Program Files (x86)\RabbitMQ Server\rabbitmq_server-3.5.3\sbin>rabbitmqctl cluster_status
Cluster status of node 'rabbit@4PJT-RABBIT3' ...
[{nodes,[{disc,['rabbit@4PJT-RABBIT1','rabbit@4PJT-RABBIT2','rabbit@4PJT-RABBIT3']}]},
 {running_nodes,['rabbit@4PJT-RABBIT1','rabbit@4PJT-RABBIT2','rabbit@4PJT-RABBIT3']},
 {cluster_name,<<"rabbit@4PJT-RABBIT1">>},
 {partitions,[]}]
```

Nous voyons que le noeud est dans le cluster, et fonctionne

Le résultat vous montre le nombre de noeuds dans le cluster, et le nombre de noeuds actuellement fonctionnels.

Configuration de la haute disponibilité

Actuellement le cluster fonctionne bien, les messages sont accessibles depuis chaque noeud, cependant en cas de chute d'un noeud qui héberge une queue, cette queue est également perdue, pour cela nous devons configurer la haute disponibilité.

Sur votre premier serveur, ouvrez une interface de ligne de commande, et déplacez vous dans le dossier suivant (remplacez le numéro de version si vous n'avez pas le même) :

```
C:\Program Files (x86)\RabbitMQServer\rabbitmq_server-3.5.3\sbin
```

Saisissez la commande suivante :

```
rabbitmqctl set_policy ha-all "." '{"ha-mode":"all"}"
```

Tests

Si vous souhaitez faire des tests ou visualiser le comportement de votre cluster, nous conseillons d'installer l'interface web de gestion.

Pour ce faire, sur chacun de vos serveurs ou vous voulez installer l'interface, suivez ces étapes :

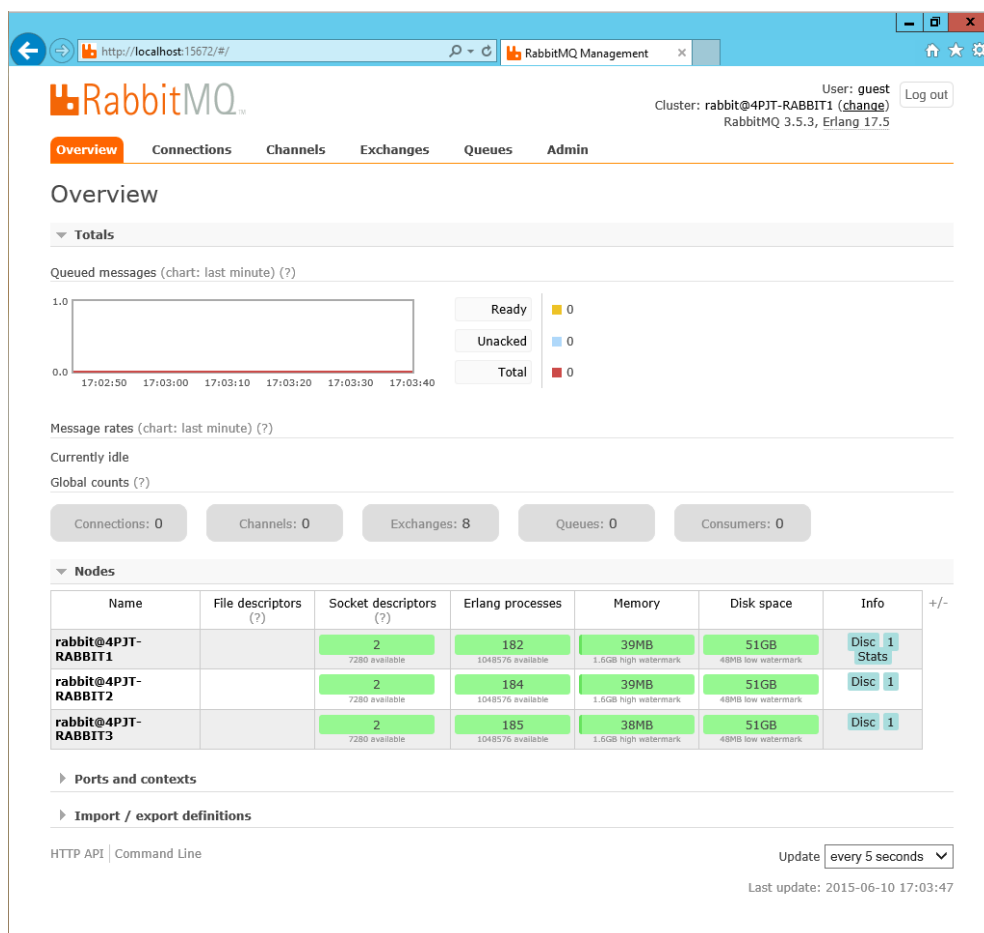
Une fois ces programmes installés, ouvrez une interface de ligne de commande, et déplacez vous dans le dossier suivant (remplacez le numéro de version si vous n'avez pas le même) :

```
C:\Program Files (x86)\RabbitMQServer\rabbitmq_server-3.5.3\sbin
```

Une fois dans ce dossier, nous allons passer le serveur RabbitMQ en mode cluster en saisissant la commande suivante :

```
rabbitmq-plugins enable rabbitmq_management
```

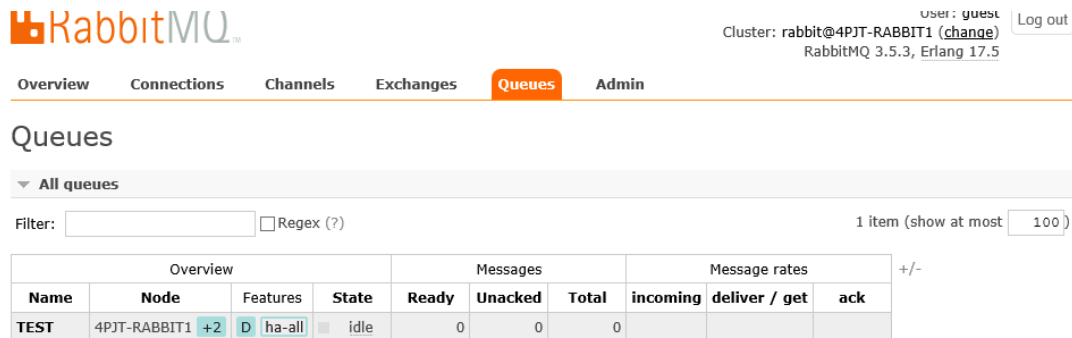
Vous pouvez désormais accéder à l'url <http://node-hostname:15672/> depuis votre navigateur pour avoir les informations.



Nous voyons notre cluster et ses détails

Il est également possible de tester les queues avec des messages depuis l'onglet "Queues".

Vous pourrez également voir ici que la haute disponibilité fonctionne car vos queues sont réparties sur tous les noeuds.



Overview				Messages			Message rates		
Name	Node	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack
TEST	4PJT-RABBIT1	+2 D ha-all	idle	0	0	0			

Interface des queues

Vous pouvez créer une nouvelle queue et tester plusieurs scénarios pouvant mettre en danger votre infrastructure (Crash serveur, coupure de courant, coupure de réseau, shutdown, etc...)

Afin de tester la chute d'un noeud sans éteindre le serveur, vous pouvez utiliser ces commandes :

Démarrer le noeud :

```
rabbitmqctl start_app
```

Éteindre le noeud :

```
rabbitmqctl stop_app
```

Attention : Dans le cas d'une coupure réseau, RabbitMQ n'est pas capable de se réparer automatiquement, vous serez dans le cas d'une erreur "NETWORK PARTITION".

Vous devrez résoudre manuellement ce conflit en choisissant une partition de confiance que vous utiliserez pour répliquer les données de cette partition sur les partitions en faute.

Une fois que vous avez choisi la partition de confiance, redémarrez tous les noeuds dans les partitions en faute avec les commandes listées précédemment.

Afin d'avoir plus de détails sur ce cas, veuillez vous référer au guide officiel :

<http://www.rabbitmq.com/partitions.html>

Note : Nous avons vu comment configuré le cluster manuellement, mais sachez qu'il est possible d'automatiser ce processus grâce à un fichier de configuration. Nous vous invitons à vous informer sur le site officiel : <https://www.rabbitmq.com/clustering.html> (Section "Auto-configuration of a cluster")

X. Vyatta

Nous avons décidé en solution de recours, si la nôtre ne vous convient pas, la mise en place Vyatta en VPN site to site IPSEC.

Configuration de l'IPSEC

Nous considérons que la configuration de base a été effectuée.

Ici l'adresse publique de routeur virtuel à New-York sera : 80.10.10.10 ; et celle de Dallas : 70.10.10.10. Ces adresses publiques sont à titre indicatif puisque nous n'avons pas encore les adresses ip publiques de l'infrastructure.

En adresse réseau privé, nous allons utiliser : 192.92.10.0/24 pour New-York et pour Texas : 192.92.20.0/24

Tout d'abord, connectez-vous en SSH sur le routeur NY:

```
ssh root@80.10.10.10
```

Entrer en mode de configuration :

```
root@routeurNY: configure
root@routeurNY#:
```

Activer le VPN sur l'interface connectée à votre WAN :

```
Set vpn ipsec ipsec-interfaces interface eth0
```

Créer la configuration ike-group :

```
Set vpn ipsec ike-group IKE-NY proposal 1
```

Mettre en place l'encryption :

```
Set vpn ipsec ike-group IKE-NY proposal 1 encryption aes256
```

Mettre en place le hash :

```
Set vpn ipsec ike-group IKE-NY proposal 1 hash sha1
```

Enfin mettre le lifetime du groupe :

```
Set vpn ipsec ike-group IKE-NY proposal 1 lifetime 3600
```

Maintenant, nous allons créer l'esp group :

```
Set vpn ipsec esp-group ESP-NY proposal 1
```

Choisir l'encryption :

```
Set vpn ipsec esp-group ESP-NY proposal 1 encryption aes256
```

Choisir l'algorithme :

```
Set vpn ipsec esp-group ESP-NY proposal 1 hash sha1
```

Et enfin le lifetime :

```
Set vpn ipsec esp-group ESP-NY proposal 1 lifetime 1800
```

Il faut maintenant configurer la connexion vers l'autre routeur de Dallas, en indiquant l'ip public de Dallas :

```
Edit vpn ipsec site-to-site peer 70.10.10.10
```

Configurer le mode d'authentification :

```
Set authentication mode pre-shared-secret
```

Indiquer le mot de passe de l'authentification :

```
Set authentication pre-share-secret Supinf0
```

Spécifier le group esp pour tous les tunnels :

```
Set default-esp-group ESP-NY
```

Spécifier maintenant le groupe ike :

```
Set ike-group IKE-NY
```

Identifier maintenant l'adresse ip du routeur local :

```
Set local-address 80.10.10.10
```

Créer maintenant le tunnel du sous réseau local :

```
Set tunnel 1 local prefix 192.92.10.0/24
```

Indiquer dès à présent le réseau local coté Dallas :

```
Set tunnel 1 remote prefix 192.92.20.0/24
```

Il faut maintenant configurer le routeur Dallas :

Connectez-vous en SSH sur le routeur TX:

```
ssh root@70.10.10.10
```

Entrer en mode de configuration :

```
root@routeurTX: configure
root@routeurTX#:
```

Activer le VPN sur l'interface connectée à votre WAN :

```
Set vpn ipsec ipsec-interfaces interface eth0
```

Créer la configuration ike-group :

```
Set vpn ipsec ike-group IKE-TX proposal 1
```

Mettre en place l'encryption :

```
Set vpn ipsec ike-group IKE-TX proposal 1 encryption aes256
```

Mettre en place le hash :

```
Set vpn ipsec ike-group IKE-TX proposal 1 hash sha1
```

Enfin mettre le lifetime du groupe :

```
Set vpn ipsec ike-group IKE-TX proposal 1 lifetime 3600
```

Maintenant, nous allons créer l'esp group :

```
Set vpn ipsec esp-group ESP-TX proposal 1
```

Choisir l'encryption :

```
Set vpn ipsec esp-group ESP-TX proposal 1 encryption aes256
```

Choisir l'algorithme :

```
Set vpn ipsec esp-group ESP-TX proposal 1 hash sha1
```

Et enfin le lifetime :

```
Set vpn ipsec esp-group ESP-TX proposal 1 lifetime 1800
```

Il faut maintenant configurer la connexion vers l'autre routeur de New York, en indiquant l'ip public de New York :

```
Edit vpn ipsec site-to-site peer 80.10.10.10
```

Configurer le mode d'authentification :

```
Set authentication mode pre-shared-secret
```

Indiquer le mot de passe de l'authentification :

```
Set authentication pre-share-secret Supinf0
```

Spécifier le group esp pour tous les tunnels :

```
Set default-esp-group ESP-TX
```

Spécifier maintenant le groupe ike :

```
Set ike-group IKE-TX
```

Identifier maintenant l'adresse ip du routeur local :

```
Set local-address 70.10.10.10
```

Créer maintenant le tunnel du sous réseau local :

```
Set tunnel 1 local prefix 192.92.20.0/24
```

Indiquer dès à présent le réseau local coté New-York :

```
Set tunnel 1 remote prefix 192.92.10.0/24
```

Vérifier maintenant que le tunnel est up :

```
Show vpn ipsec sa
```

La configuration est dorénavant terminée !

XI. VOIP

Afin de clore cette documentation, nous allons aborder la configuration d'Asterisk pour la mise en place la VOIP.

Introduction

Tout d'abord, voici les prérequis à effectuer avant de configurer Asterisk :

- Debian
- Installer les librairies requises
- Installer Dahdi sur cette machine
- Installer Asterisk

Création des Users

Modifier le fichier suivant `/etc/asterisk/users.conf`, en créant tout d'abord les templates :

```
[mewpipe_users] (!)

type=friend
disallow=all
allow=ulaw
nat=never
host=dynamic
dtmfmode=rfc2833
hassip=yes

[it_support_template] (!,mewpipe_users)
context=it_support

[accounting_template] (!,mewpipe_users)
context=accounting

[logistic_template] (!,mewpipe_users)
context=logistic

[public_relation_template] (!,mewpipe_users)
context=public_relation

[marketing_template] (!,mewpipe_users)
context=marketing
```

Maintenant il faut ajouter les users :

```
[134] (it_support_template)
secret=itSupp0rt1
```

```
[135] (it_support_template)
secret=itSupp0rt2
```

(répéter cela jusqu'à 146)

```
[200] (accounting_template)
secret=acc0unt1ng1
```

(Répéter cela pour les users jusqu'à 251)

```
[400] (logistic_template)
secret=l0gistic1
```

(Répéter cela pour les users jusqu'à 478)

```
[480] (public_relation_template)
secret=pubrel@ti0n1
```

(Répéter cela pour les users jusqu'à 492)

```
[500] (marketing_template)
secret=m@rket1ng1
```

(Répéter cela pour les users jusqu'à 634)

Configuration des Voicemail

Nous allons désormais créer des Voicemail pour que l'utilisateur puisse avoir une messagerie en cas de non-réponse à un appel. Éditer le fichier suivant :
`/etc/asterisk/voicemail.conf`

```
[it_support_vm]

134 => 5468,itperson1,itperson1@mewpipe.com

135 => 5469,itperson2,itperson2@mewpipe.com
```

Etc pour le reste des users `it_support`.

[accounting_vm]

200 => 5610, accperson1, accperson1@mewpipe.com

Etc pour le reste des users accounting.

[logistic_vm]

400 => 5321, logperson1, logperson1@mewpipe.com

Etc pour le reste des users logistic.

[public_relation_vm]

480 => 3651, prperson1, prperson1@mewpipe.com

Etc pour le reste des users public relation.

[marketing_vm]

500 => 4510, marketperson1, marketperson1@mewpipe.com

Etc pour le reste des users marketing.

Configuration des Dialplans

Entrer dans le fichier de conf suivant `/etc/asterisk/extensions.conf`

```
[it_support]
exten => _13[4-9],1,Dial(SIP/${EXTEN},30)
same => n,VoiceMail(${EXTEN}@it_support_vm)
exten => _14[0-6],1,Dial(SIP/${EXTEN},30)
same => n,VoiceMail(${EXTEN}@it_support_vm)
exten => _2[0-4]X,1,Goto(accounting,${EXTEN},1)
exten => _25[0-1],1,Goto(accounting,${EXTEN},1)
exten => _4[0-6]X,1,Goto(logistic,${EXTEN},1)
exten => _47[0-8],1,Goto(logistic,${EXTEN},1)
exten => _48X,1,Goto(public_relation,${EXTEN},1)
exten => _49[0-2],Goto(public_relation,${EXTEN},1)
exten => _5XX,Goto(marketing,${EXTEN},1)
exten => _6[0-2]X,Goto(marketing,${EXTEN},1)
exten => _63[0-4],Goto(marketing,${EXTEN},1)

[accounting]
exten => _2[0-4]X,1,Dial(SIP/${EXTEN},30)
same => n,VoiceMail(${EXTEN}@accounting_vm)
exten => _25[0-1],1,Dial(SIP/${EXTEN},30)
same => n,VoiceMail(${EXTEN}@accounting_vm)
exten => _13[4-9],1,Goto(it_support,${EXTEN},1)
exten => _14[0-6],1,Goto(it_support,${EXTEN},1)
exten => _4[0-6]X,1,Goto(logistic,${EXTEN},1)
exten => _47[0-8],1,Goto(logistic,${EXTEN},1)
exten => _48X,1,Goto(public_relation,${EXTEN},1)
exten => _49[0-2],Goto(public_relation,${EXTEN},1)
exten => _5XX,Goto(marketing,${EXTEN},1)
exten => _6[0-2]X,Goto(marketing,${EXTEN},1)
exten => _63[0-4],Goto(marketing,${EXTEN},1)

[logistic]
exten => _4[0-6]X,1,Dial(SIP/${EXTEN},30)
same => n,VoiceMail(${EXTEN}@logistic_vm)
exten => _47[0-8],1,Dial(SIP/${EXTEN},30)
same => n,VoiceMail(${EXTEN}@logistic_vm)
exten => _13[4-9],1,Goto(it_support,${EXTEN},1)
exten => _14[0-6],1,Goto(it_support,${EXTEN},1)
exten => _2[0-4]X,1,Goto(accounting,${EXTEN},1)
exten => _25[0-1],1,Goto(accounting,${EXTEN},1)
exten => _48X,1,Goto(public_relation,${EXTEN},1)
exten => _49[0-2],Goto(public_relation,${EXTEN},1)
exten => _5XX,Goto(marketing,${EXTEN},1)
exten => _6[0-2]X,Goto(marketing,${EXTEN},1)
exten => _63[0-4],Goto(marketing,${EXTEN},1)
```

```

[public_relation]
exten => _48X,1,Dial(SIP/${EXTEN},30)
same => n,VoiceMail(${EXTEN}@public_relation_vm)
exten => _49[0-2],1,Dial(SIP/${EXTEN},30)
exten => n,VoiceMail(${EXTEN}@public_relation_vm)
exten => _13[4-9],1,Goto(it_support,${EXTEN},1)
exten => _14[0-6],1,Goto(it_support,${EXTEN},1)
exten => _2[0-4]X,1,Goto(accounting,${EXTEN},1)
exten => _25[0-1],1,Goto(accounting,${EXTEN},1)
exten => _4[0-6]X,1,Goto(logistic,${EXTEN},1)
exten => _47[0-8],1,Goto(logistic,${EXTEN},1)
exten => _5XX,Goto(marketing,${EXTEN},1)
exten => _6[0-2]X,Goto(marketing,${EXTEN},1)
exten => _63[0-4],Goto(marketing,${EXTEN},1)

```

```

[marketing]
exten => _5XX,1,Dial(SIP/${EXTEN},30)
same => n,VoiceMail(${EXTEN}@marketing_vm)
exten => _6[0-2]X,1,Dial(SIP/${EXTEN},30)
same => n,VoiceMail(${EXTEN}@marketing_vm)
exten => _63[0-4],1,Dial(SIP/${EXTEN},30)
same => n,VoiceMail(${EXTEN}@marketing_vm)
exten => _13[4-9],1,Goto(it_support,${EXTEN},1)
exten => _14[0-6],1,Goto(it_support,${EXTEN},1)
exten => _2[0-4]X,1,Goto(accounting,${EXTEN},1)
exten => _25[0-1],1,Goto(accounting,${EXTEN},1)
exten => _4[0-6]X,1,Goto(logistic,${EXTEN},1)
exten => _47[0-8],1,Goto(logistic,${EXTEN},1)
exten => _48X,1,Goto(public_relation,${EXTEN},1)
exten => _49[0-2],Goto(public_relation,${EXTEN},1)

```

Configuration du Trunk

Dans le fichier `/etc/asterisk/sip.conf` : (Dans notre exemple nous mettons la syntaxe des sip que le fournisseur doit vous donner)

```
Register=> login :password@host/DID
```

```
[mewpipeTrunk]
```

```

Type=peer
host=host
fromuser=login
username=login
secret=password
insecure=invite,port
qualify=yes
nat=never
context=incoming_trunk

```

Configuration des appels venant de l'extérieur

Dans /etc/asterisk/extensions.conf

```
[incoming_trunk]
exten=> 0XXXXXXXX,1,Goto(ivr,s,1)
same=> n, Hangup() ;
```

Configuration des appels vers l'extérieur + numéro urgence

Dans /etc/asterisk/extension.conf

```
[outgoing_calls]

exten => _0XXXXXXXX,1,Dial(SIP/mewpipeTrunk/${EXTEN})

exten => _0XXXXXXXX,2,Goto(trunk_error,s,1)

exten => _[1-3]XXX,1,Dial(SIP/mewpipeTrunk/${EXTEN})

exten => _[1-3]XXX,2,Goto(trunk_error,s,1)

exten => _1[578],1,Dial(SIP/mewpipeTrunk/${EXTEN})

exten => _1[578],2,Goto(trunk_fail,s,1)
```

```
[trunk_error]

exten => s,1,Answer()

exten => s,2,Playtones(congestion)

exten => s,3,Congestion()
```

```
[it_support]

include => outgoing_calls
```

```
[accounting]

include => outgoing_calls
```

```
[logistic]

include => outgoing_calls
```

```
[public_relation]

include => outgoing_calls
```

```
[marketing]

include => outgoing_calls
```

Configuration de l'IVR

Il faut d'abord installer les services sur votre VMs !

```
Apt-get install perl libwww-perl sox mpg123
```

```
Wget -O GoogleTTS.tar.gz http://github.com/zaf/asterisk-googleletts/tarball/master --no-check-certificate
```

```
tar -xvf GoogleTTS.tar && cd zaf-asterisk-googleletts-51c2db5 && cp
googleletts.agi /var/lib/asterisk/agi-bin/
```

Maintenant la configuration dans `/etc/asterisk/extensions.conf` :

```
[ivr_number]

exten => 900,1,Goto(ivr,s,1)
```

```
[it_support]

include => ivr_number
```

```
[accounting]

include => ivr_number
```

```
[logistic]

include => ivr_number
```

```
[public_relation]

include => ivr_number
```

```
[marketing]

include => ivr_number
```

```

[ivr]

exten => s,1,Answer()

exten => s,2,Set(TIMEOUT(response)=10)

exten => s,3,agi(googletts.agi,"Welcome to MewPipe industry, may the
creation be with you !",en,any)

exten => s,4,agi(googletts.agi,"Which department would you
like to speak to?",en,any)

exten => s,5,agi(googletts.agi,"Press 1 to contact IT Support
department",en,any)

exten => s,6,agi(googletts.agi,"Press 2 to contact Accounting
department",en,any)

exten => s,7,agi(googletts.agi,"Press 3 to contact Logistic
department",en,any)

exten => s,8,agi(googletts.agi,"Press 4 to contact Public Relation
department",en,any)

exten => s,9,agi(googletts.agi,"Press 5 to contact Marketing
department",en,any)

exten => s,10,agi(googletts.agi,"Press # to listen again this
message",en,any)

exten => s,11,WaitExten()


exten => 1,1,Goto(rings,1101,1)

exten => 2,1,Goto(rings,1102,1)

exten => 3,1,Goto(rings,1103,1)

exten => 4,1,Goto(rings,1104,1)

exten => 5,1,Goto(rings,1105,1)

exten => _[05-9#*],1,Goto(ivr,s,3)

exten => t,1,Goto(ivr,s,3)

```

[rings]

exten => 1101,1,Dial(SIP/134&SIP/135..etc,30) ;correspond au ring it support

exten => 1102,1,Dial(SIP/200&SIP/201..etc,30) ;ring Accounting

exten => 1103,1,Dial(SIP/400&SIP/401..etc,30) ;ring Logistic

exten => 1104,1,Dial(SIP/480&SIP/481..etc,30) ;ring Public Relation

exten => 1105,1,Dial(SIP/500&SIP/501..etc,30) ;ring Marketing