Années : 2012-2014

BTS SIO

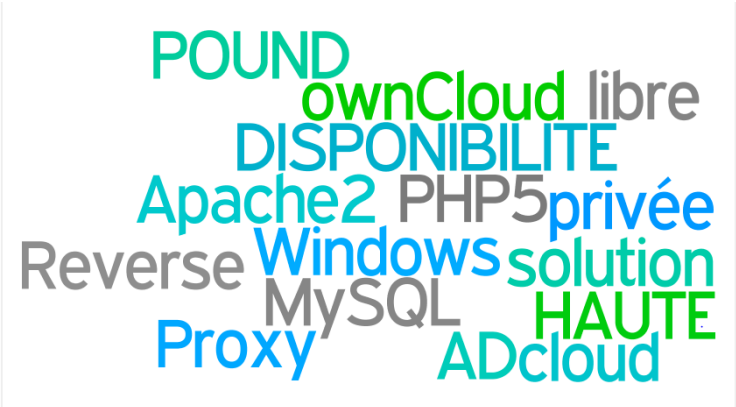
Situation professionnelle numéro 5

Architecture cloud privée sous linux avec liaison AD Windows 2008R2

Description :

Une architecture cloud privée permet à une entreprise de disposer de ses fichiers dans un environnement contrôlé et fiable. L'annuaire de l'AD peut être lié au cloud et peut contrôler les accès utilisateurs.

Mots-clés :



Validation de la situation professionnelle

Nom	Date	Tampon
	26/05/2014	

Plan de la situation

Le cahier des charges

L’expression des besoins

Notre société souhaite développer des services dans le cloud de type « privé ».
Nous serons hébergés chez OVH, avec deux serveurs EG-64 infrastructure et différentes options.
Dans un future proche notre infrastructure sera full SDN afin d'avoir une scalabilité horizontale et un PRA.
L'ensemble de l'infrastructure reposera sur Vmware Vcenter !

Cependant, nous voulons dans un 1er temps prendre conscience des possibilités offertes par un hebergeur (découvert dans la documentation précedente),mettre en œuvre cette solution cloud privée dans une infrastructure de test , immaginer notre cloud privée, ainsi que découvrir les solutions qui s’offrent à nous.

La description de l’existant

Nous ne possédons aucune solution cloud de type « privée » ni d’aucun type d’ailleurs.
Cependant, nous utilisons Mega, le service de Kim Dotcom pour le stockage de nos images ISO.

L’analyse des choix

Dans notre analyse, nous souhaitons disposer d’un cloud privé en centralisant des répertoires et des fichiers nomades afin que des « chefs de travaux » puissent rapidement « uploader » et « downloader » leurs documents.
L’analyse est de réfléchir aux avantages d'une architecture d'un cloud privé (simple) et de pouvoir fournir une disponibilité du service plus fiable, car nos clients travaillent également le weekend et aussi dans le monde entier !
Afin de garantir une bonne experience à nos clients, nous souhaitons :

• Disposer d’un reverse proxy pour sécuriser un peu plus notre cloud, et surtout décharger les pages web.

• Pouvoir faire une modification du stockage à la volée avec par exemple le protocole ISCSI.

• Une possibilité d’ajouter des « clones » de notre serveur cloud afin d’avoir une grande élasticité.

On parle d’une application en mode « SaaS » car notre client final aura directement accès à l’application.
Le nuage que nous utiliserons sera : Owncloud qui est une solution open-source.

Nous disposerons donc de trois serveurs pour notre architecture cloud privé :

- UTIUFA108-srv.reverse.proxy
- UTIUFA108-srv.web.1
- UTIUFA108-srv.web.2

Les offres du marché

Les offres Cloud privées sont nombreuses et tous les acteurs IT se sont lancés sur ce marché porteur .
Les offres sont très variées et abordables pour une entreprise et ce ,quelque soit sa taille, c'est un bussiness rentable !

Les offres de Cloud privés d'IBM se déclinent en plusieurs couches, du stockage en passant par la sécurité (filtrage de messagerie, détection de vulnérabilité...), jusqu'aux applications en mode SaaS, notamment collaboratives. L'offre recouvre également le test et la gestion de services IT, avec Tivoli Live. Six datacenters sont destinés à ces solutions, auxquels s'ajoute un centre de compétences spécialisés situé à La Gaude.

De son côté, EMC à mis à jour l'environnement matériel et applicatif de son infrastructure Cloud, Atmos. Concrètement, le fournisseur de stockage a installé dans ses serveurs Cloud des processeurs Intel Xeon 5500 complétés par des disques durs haute densité de 2 To, portant la capacité de stockage sur disque par cabinet à 720 To.

IBM, EMC, VMware, HP, Microsoft et Dell se partagent le marché

VMware a présenté lors de son évènement VMworld 2010 sa solution VMware vCloud Director. Complémentaire de l'hyperviseur de virtualisation VMware vSphere et du gestionnaire de machines virtuelles vCenter, ce nouveau service doit permettre à un administrateur de gérer plus facilement les architectures de Cloud. Y compris celles fonctionnant en mode multi-tenant, c'est-à-dire permettant à une seule instance logicielle de servir simultanément plusieurs requêtes clients.

HP s'est lui récemment positionné sur le segment de marché du Cloud privé. "Nous venons de lancer deux nouvelles offres centrées sur cette problématique. Flex Datacenter qui apporte une solution de centre de données modulaire optimisée d'un point de vue énergétique, et l'offre de services Cloud Start qui permet de déployer un Cloud privé rapidement, en moins de 30 jours", faisait ainsi savoir Jean-Paul Alibert interrogé dans les colonnes du journal du net en septembre dernier.

Autre américain à s'être lancé récemment sur le segment du Cloud Computing privé : Microsoft. Un acteur qui a d'ailleurs conclu un partenariat avec HP sur une autre offre packagée commune "Infrastructure-to-Applications" où ce dernier apporte ses solutions BladeSystem Matrix et un portail d'orchestration des ressources et la firme de Redmond System Center et HyperV et HP

Sources : http://www.journaldunet.com/solutions/systemes-reseaux/cloud-computing-prive/fournisseurs-cloud-prive.shtml

Le choix de la différence avec une solution open source

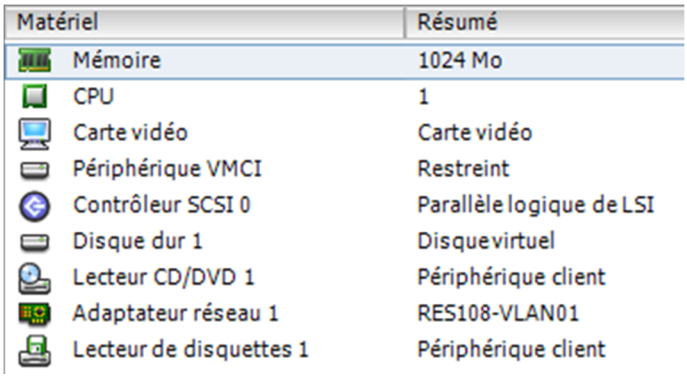
Afin de comprendre le cloud dans sa globalité, nous n'avons pas choisi les acteurs du marché cités ci-dessus.
En effet nous nous sommes orientés vers une solution open-source de type SaaS que nous allons intégrer.
Le nom de la solution que nous allons fournir à nos clients est OwnCloud.
La solution est très simple d'accès et jouit d'une communauté très réactive et productive.
Un forum ouvert et multilingue est à disposition : https://forum.owncloud.org/
Le site officiel : http://owncloud.org/

Mise en œuvre

Installation d'un serveur web avec« LAMP »

L'installation de Owncloud sera lancée après l'installation du serveur web « lamp ».
Nous partons du principe que notre machine Linux est déjà installée et que nous sommes en « root ».

La serveur utilise les ressources suivantes dans l'hyperviseur :



Matériel	Résumé
Mémoire	1024 Mo
CPU	1
Carte vidéo	Carte vidéo
Périphérique VMCI	Restreint
Contrôleur SCSI 0	Parallèle logique de LSI
Disque dur 1	Disque virtuel
Lecteur CD/DVD 1	Périphérique client
Adaptateur réseau 1	RES108-VLAN01
Lecteur de disquettes 1	Périphérique client

Linux

Nous allons commencer par configurer la carte réseau avec nano« eth0 » (adresse statique).
Il faut pour cela se rendre dans : `/etc/network/interfaces` et configurons le comme ceci :

```
auto eth0

    iface eth0 inet static
        address 192.0.2.7
        netmask 255.255.255.0
        gateway 192.0.2.254
```

Nous quittons nano avec : « ctrl +x » puis nous validons par « yes »
Pour relancer le service réseau : `/etc/init.d/networking restart`
La mise en place du proxy peut être necessaire pour l'update et l'upgrade de nos paquets.
Pour cela utilisons la commande suivante :

```
export http_proxy='http://172.16.0.4 :3128'
```

Procédons aux mises à jour de la distribution et du « sources.list » :

```
apt-get update
apt-get upgrade
```

Apache2

L'installation du serveur web peut désormais commencer :

```
apt-get install apache2
```

Pour verifier que le serveur est lancé, la commande suivante permet de connaître les connexions ouvertes :

```
netstat –tan
```

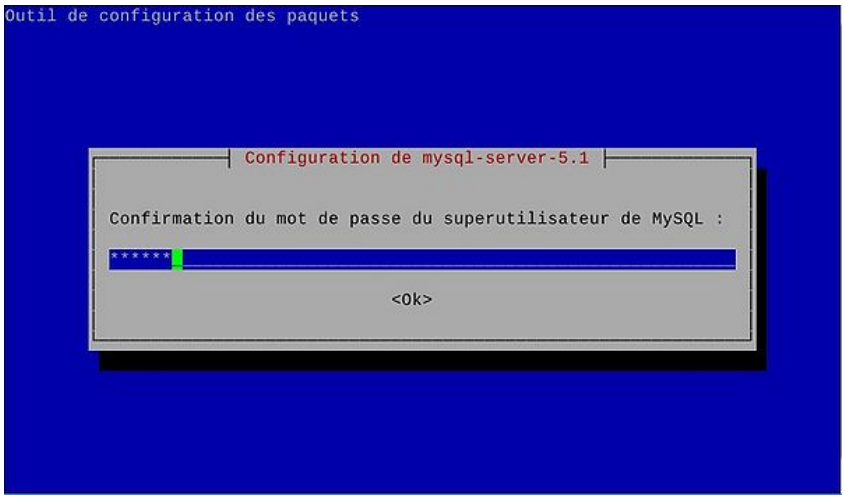
Ou encore, rendons nous sur l'url de notre serveur web qui affiche le message :

```
It works!
```

Mysql

L’installation du serveur de base de données (SGBD) :
`apt-get install mysql-server`

Durant l'installation nous allons définir le mot de passe de notre accès root à la base de données :



Après l’installation, nous lancons mysql de la façon suivante :

```
mysql –u root –p
Enter password : ********
```

Ensuite, nous allons simplement créer un utilisateur, une base avec l’accès à notre utilisateur :

```
CREATE DATABASE owncloud;
CREATE USER 'owncloduser'@'localhost' IDENTIFIED BY 'p4ssw0rd'
GRANT ALL PRIVILEGES ON owncloud.* TO owncloduser'@’localhost’;

OU

CREATE USER 'owncloduser'@'localhost' IDENTIFIED BY 'p4ssw0rd';
CREATE DATABASE IF NOT EXISTS owncloud;
GRANT ALL PRIVILEGES ON owncloud.* TO owncloduser'@'localhost' IDENTIFIED BY ‘p4ssw0rd';
```

PHP-5

Pour que notre serveur comprenne le language de programmation PHP il faut lui installer la version 5 et quelques dépendances :
`apt-get install php5 php5-common php5-gd`

Nous avons terminé l’installation du serveur web sous lamp, owncloud peut enfin être mis en place.

Installation de OwnCloud 6

Nous sommes fin prêt à déployer owncloud sous Debian 7.
Afin de pouvoir récuperer owncloud, il nous faut le depôt de owncloud dans notre source.list :

```
echo 'deb http://download.opensuse.org/repositories/isv:/ownCloud:/community/Debian_7.0/
/' >> /etc/apt/sources.list.d/owncloud.list
```

Il est conseillé de récupérer la clé du depôt pour éviter des problèmes et de l'ajouter à linux :

```
wget
http://download.opensuse.org/repositories/isv:ownCloud:community/Debian_7.0/Release.key
apt-key add - < Release.key
```

Puis, on lance une synchronisation des depôts de notre source.list :

```
apt-get update
```

Ensuite nous installons owncloud :

```
apt-get install owncloud
```

L'installation est rapide, ensuite il faut se rendre dans le répertoire de ownlcoud et définir le groupe "**www-data**" en tant que groupe propriétaire sur les répertoires utiles au fonctionnement d'ownCloud :

```
cd /var/www/owncloud
mkdir data
chgrp www-data data –R
chgrp www-data config –R
chgrp www-data apps –R
```

Modification des permissions sur les différents répertoires (avec récursivité)

```
chmod 770 data -R
chmod 770 config -R
chmod 770 apps –R
```

Activons les modules rewrite et headers d'Apache2 :

Le Rewrite permet la réécriture d'URL, et, headers permet de gérer les en-têtes des requêtes/réponses HTTP/.

```
a2enmod rewrite
a2enmod headers
```

Modifier notre « DocumentRoot » et « Directory » dans notre vhost par défaut d'Apache2 :

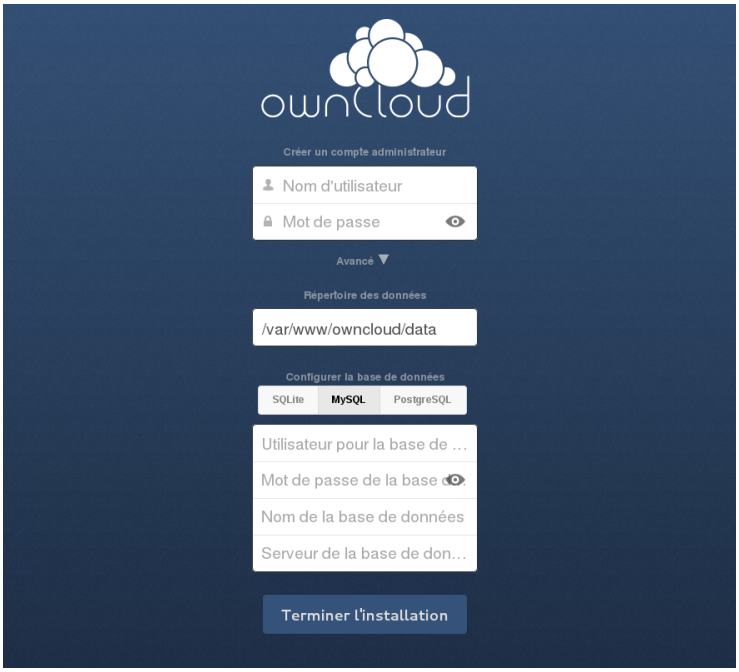
Le serveur web pointe désormais par défaut dans le repertoire de owncloud.

```
nano /etc/apache2/sites-available/default
DocumentRoot /var/www/owncloud/
Directory /var/www/owncloud/
```

Pour terminer l'installation de notre owncloud, nous redémarrons notre serveur Apache2 :

```
/etc/init.d/apache2 restart
```

Le serveur Owncloud est désormais accessible, et fonctionnel via son adresse IP ou sont nom DNS :



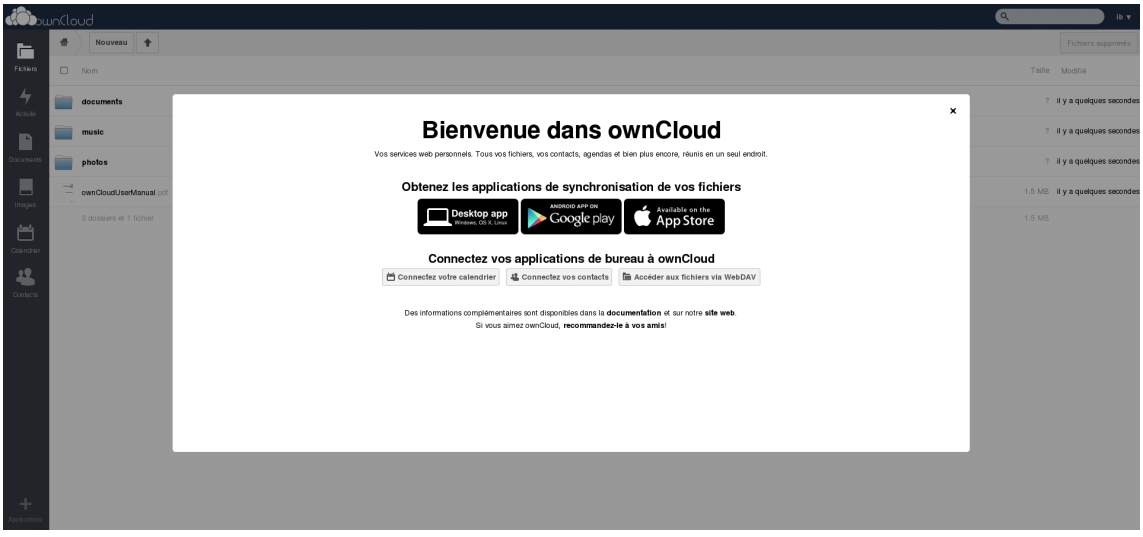
Désormais OwnCloud apparaît avec son Nuage blanc, il nous permet de créer un compte administrateur :

- ffonaissak
- p4ssw0rd

Laissons le repertoire par défaut pour les données : `/var/www/owncloud/data`
La configuration de la base de données est bien “MySQL”
Les derniers champs ont été précedemment configurés dans notre base de données MySQL :

- owncloduser
- p4ssw0rd
- owncloud
- 127.0.0.1

Après avoir cliqué sur : « Terminer l’installation », nous voici connecté à Owncloud :



Le reverse proxy POUND

Un proxy inverse ou reverse proxy est un type de serveur, placé "frontalement" sur internet.
Il permet à un utilisateur d'Internet d'accéder à des serveurs placés à l'intérieur d'un réseau privé.
POUND permet de faire plusieurs choses telles que :

- Du load balancing avec gestion du fail over ;
- Du reverse proxy ;
- Du SSL wrapping (il s'occupera de l'encapsulation/désencapsulation de HTTPS vers HTTP).

Il est très léger , performant et capable de traiter des requêtes HTTPS, il sait répartir la charge de travail en envoyant des requêtes Web aux machines les moins occupées (suivant le fichier de configuration).
Situé dans une position idéale, il soulage les serveurs web, nous l'installerons sous Debian 7.

Installation de Pound version 2.6

Afin d'installer le reverse proxy pound en version 2.6, il nous faut rajouter le dépôt officiel de Debian France :

```
nano /etc/apt/source.list
deb http://ftp.fr.debian.org/debian wheezy main
```

Synchronisation des miroirs et installation de pound :

```
apt-get update && apt-get upgrade
apt-get install pound
```

Programmer le lancement automatique de pound au démarrage :

```
nano /etc/default/pound
startup=1
```

Le fichier de configuration de Pound est "pound.cfg" et se situe dans le repertoire suivant :

```
nano /etc/pound/pound.cfg
```

Détail du fonctionnement souhaité :

Pound réceptionne les requêtes en HTTP ou HTTPS dans notre cas en HTTP.
Il transfere les requêtes vers le serveur web-1 (défaut) sinon à web-2.
Si web-1 est indisponible alors web-2 desservira nos clients afin que la panne ne soit pas visible.

A savoir : Nous avons décidé de "décharger" le serveur web-1 des extensions suivantes .jpg, .png et .svg.
Procédons à la configuration de notre reverse proxy ci-dessous.

Configuration de Pound version 2.6

La configuration pour que Pound soit en écoute sur notre serveur en http :

```
Alive 10
ListenHTTP
Address 192.168.0.2
Service
BackEnd
Address 172.31.108.2
Port 80
End
```

La configuration pour que Pound n'accepte que l'en-tête défini “owncloud.ksff.me”:

```
Service
HeadRequire "Host: .owncloud.ksff.me.*"
BackEnd
Address 172.31.108.2 #serveur local
Port 80
Priority 5
End
```

La configuration pour que pound envoit les éléments d'affichage suivant au serveur web-2 :

```
Service
URL “.*.(jpg|gif|svg)”
BackEnd
Address 172.31.108.2
Port 80
End
```

La configuration pour que pound prenne la decision de choisir web-1 ou web-2 avec une priorité :

```
Service
HeadRequire "Host: .owncloud.ksff.me.*"
BackEnd
Address 172.31.108.3 #serveur web-1
Port 80
Priority 5
End
BackEnd
Address 172.31.108.4 #serveur web-2
Port 80
Priority 4
End
```

Configuration pour que le cookies de l'utilisateur soit sauvegardé :

```
Session
Type Cookie
ID "sess"
TTL 300
End
```

Configuration initiator ISCSI sur UTIUFA108-srv.web.1

Dans notre réseau nous avons un ISCSI Target (serveur) disponible, nous allons donc l'ajouter dans web.1. Nous avons besoin des paramétres de connexions suivants, car le serveur est à configurer l'ISCSI de cette façon :

Authentification mode : CHAP
Utilisateur : ffoanissak
Password : *******

Installons le logiciel avec le gestionnaire APT de Debian :
```
apt-get install open-iscsi
```

Le répertoire du fichier de configuration se trouve dans :
```
nano /etc/iscsi/iscsid.conf
```

Décommenter les lignes suivantes :
```
node.session.auth.authmethod = CHAP
node.session.auth.username = ffonaissak
node.session.auth.password = ********
```

Nous quittons nano avec : « ctrl +x » et nous validons par « yes »

Pour relancer le service iscsi : `/etc/init.d/open-iscsi restart`

Pour faire une découverte des targets disponibles sur notre clible :
```
iscsiadm -m discovery -t sendtargets -p 172.31.108.10
```

Pour regarder le nœud et son détail :
```
iscsiadm -m node -o show
```

Identifions nous à notre target :
```
iscsiadm -m node –login
Logging in to [iface: default, target: iqn.2013-05.world.server:target00, portal:
172.31.108.10,3260] (multiple)
Login to [iface: default, target: iqn.2013-05.world.server:target00, portal:
172.31.108.10,3260] successful.
```

Vérifions notre connexion en TCP au Target ISCSI :
```
iscsiadm -m session -o show
tcp: [1] 172.31.108.10:3260,1 iqn.2013-05.world.server:target00
```

Vérifions les nouvelles partions présentes sur notre système :
```
cat /proc/partitions
7176192 sdb #voici notre nouvelle partitions de 7Go environ#
```

Montage de l’initiator ISCSI sur UTIUFA108-srv.web.1

Nous allons utiliser fdisk qui est un outil de base pour réaliser des opérations sur les tables de partitions.

```
fdisk /dev/sdb
new/primary
7348,42
write
```

Pour lister les disques présents dans notre système linux :

```
fdisk –l

Disk /dev/sdb: 7348 MB, 7348420608 bytes

178 heads, 4 sectors/track, 20157 cylinders, total 14352384 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
Device Boot Start End Blocks Id System

/dev/sdb1 62 14352383 7176161 83 Linux #ici notre partition sdb1 de 7Go#
```

Formatons notre système en EXT4, nous ne souhaitons pas que Windows puisse lire la table de partition :

```
mkfs.ext4 /dev/sdb1
```

Montage de la parition ISCSI sur UTIUFA108-srv.web.1

Nous sommes maintenant disposés à monter notre partition dans un répertoire de notre système Linux.
Pour pouvoir faire une modification du stockage à la volée, nous souhaitons le monter dans owncloud :

```
mkdir /var/www/owncloud/owncloud-data
mount /dev/sdb1 /var/www/owncloud/owncloud-data/
```

Le groupe définit à l’utilisateur www-data et le groupe root

```
Chown www-data:owncloud /var/www/owncloud/owncloud-data/
```

Le groupe et l’utilisateur auront le droit à : rwx-rwx

```
chmod 770 /var/www/owncloud/owncloud-data/
```

Modification de owncloud sur UTIUFA108-srv.web.1

Le fichier de « données » de owncloud est présent dans le répertoire :

```
/var/www/owncloud/data/
```

Nous souhaitons le modifier, nous devons donc modifier le fichier de configuration de owncloud :

```
nano /var/www/owncloud/config/config.php
```

La valeur actuelle de «datadirectory » devra être remplacée par : `‘/var/www/owncloud/owncloud-data/’,`
Afin de ne pas compromettre les données présentes, nous les copions de l’ancien répertoire au nouveau :

```
mv /var/www/owncloud/data/* /var/www/owncloud/owncloud-data/
```

La modification du répetoire des données de owncloud est terminée, et cela fonctionne très bien !

Connexion AD avec owncloud sur UTIUFA108-srv.web.1

Nous avons la possibilité de nous connecter à un annuaire LDAP avec owncloud.
Nous utiliserons l'active directory de Windows serveur 2008R2 disponible dans le réseau de notre entreprise.
Voici comment est configuré notre serveur Active Directory nommé « windows2008 »

1. Créer un domaine dans une nouvelle forêt avec le nom de domaine complet : KSFF.ME
2. Le niveau fonctionnel de la forêt sous : Windows Server 2008R2
3. Voici les options supplémentaires sélectionnées, pour le contrôleur de domaine : DNS et Catalogue Global
4. L'emplacement de la base de données, des journaux et de SYSVOL sont par défaut
5. Le mot de passe administrateur de restauration AD est configuré.

Notre annuaire active directory contient :

- Deux unité d'organisation : « entreprise » et « owncloud-ldap ».
- Deux utilisateurs : « jk » et « owncloudaduser ».

L'unité d'organisation « owncloud-ldap » contient :

- Le groupe « owncloud-ldap »
- L'utilisateur « owncloudaduser »

[La mise en place de la solution de liaison AD sur la ferme à était suspendu par manque de temps]
[Je serai cependant capable de la mettre en place lors de l'examen final]