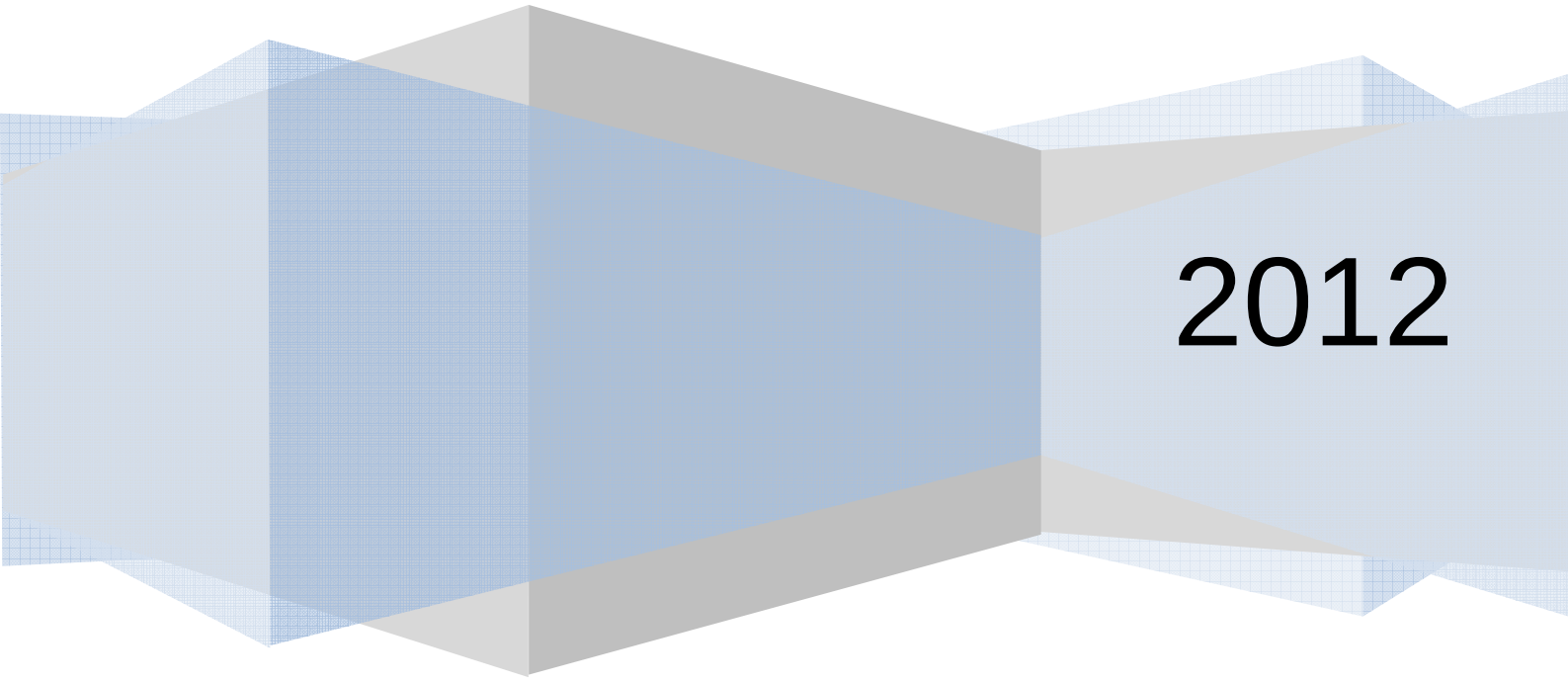
BIND server-install and configure

Réalisée sur CentOS 5.7

Ecrit par Charles-Alban BENEZECH



2012

Sommaire

1. BIND

1.1 Introduction

Bind est un serveur de nom ou DNS (Domain Name Server). Le service DNS permet la communication entre les machines d'un ou plusieurs réseaux interconnectés grâce à son système de résolution de nom permettant de transformer un nom de machine (exemple: W2K8DCT01.domaine.fr) en une adresse IP (exemple: 192.168.30.1).

Un DNS est la pierre angulaire des réseaux, de sa configuration dépendra l'efficacité du système d'information.

1.2 Fonctionnement

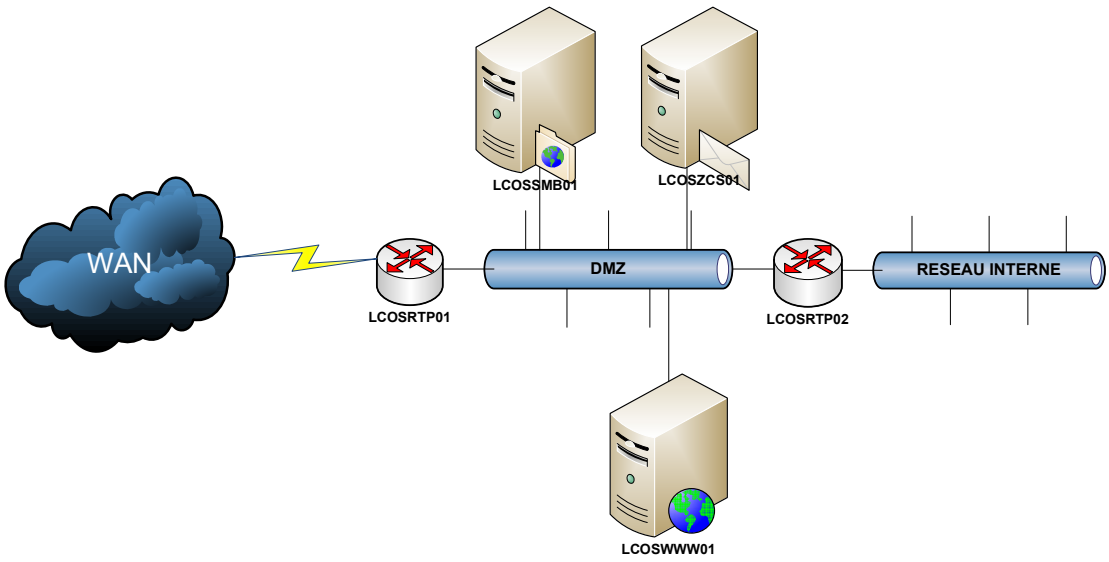
Un DNS fonctionne comme un annuaire où un nom de machine correspond à une adresse et inversement (si la fonction reverse est activée). En effet lorsque qu'une requête est transmise sur le réseau vers une machine le plus souvent les applications ou les utilisateurs ou bien encore les administrateurs vont utiliser le nom de la machine afin de simplifier la tâche; dans ce cas on fera appel à notre DNS afin qu'il nous donne l'adresse IP de la machine visée, ensuite le paquet sera routé jusqu'au destinataire.

Un DNS fonctionne très simplement de la manière suivante: il reçoit une requête DNS, tente de la résoudre, s'il y parvient il retourne le résultat attendu sinon il demande au serveur au dessus de lui (celui de son fournisseur d'accès (FAI) par exemple) et ainsi de suite. Les requêtes ne peuvent aller que vers le haut (entendez dans la hiérarchie des DNS) sauf dans certains cas particuliers.

Nous pouvons isoler trois type de configuration: autonome ou standalone, réplication, délégation. Ces topologies ont toutes trois leurs utilités et leurs défauts et sont utilisées chacune dans des cas différents.

1.2.1 Autonome

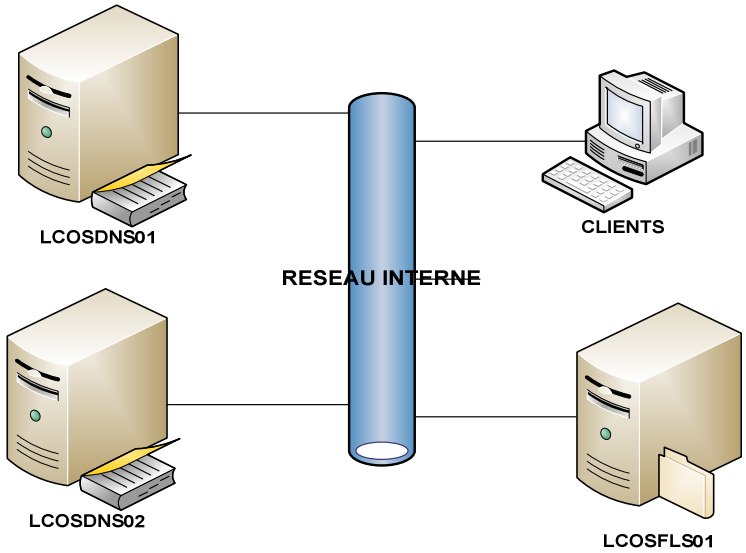
Un serveur DNS autonome, appelé aussi "standalone server", est un serveur qui se suffit à lui même. En effet, il fonctionnera seul et pour de petites tâches, couplé avec un service particulier par exemple.



Dans cet exemple, notre service DNS est hébergé par le serveur de mail LCOSZCS01 qui permettra de résoudre les requêtes à l'intérieur de la zone démilitarisée (DMZ). Ce serveur n'aura que cette tâche plus ou moins légère ce qui permet de se protéger d'éventuelles attaques ou failles de sécurité qu'auraient provoqué la mise en place d'un DNS contenant tous l'adressage interne dans la DMZ ou la redirection de toutes les requête vers un DNS vers l'intérieur. Dans ce cas seul le DNS de la DMZ est capable d'envoyer des requêtes DNS vers le réseau interne.

1.2.2 Réplication

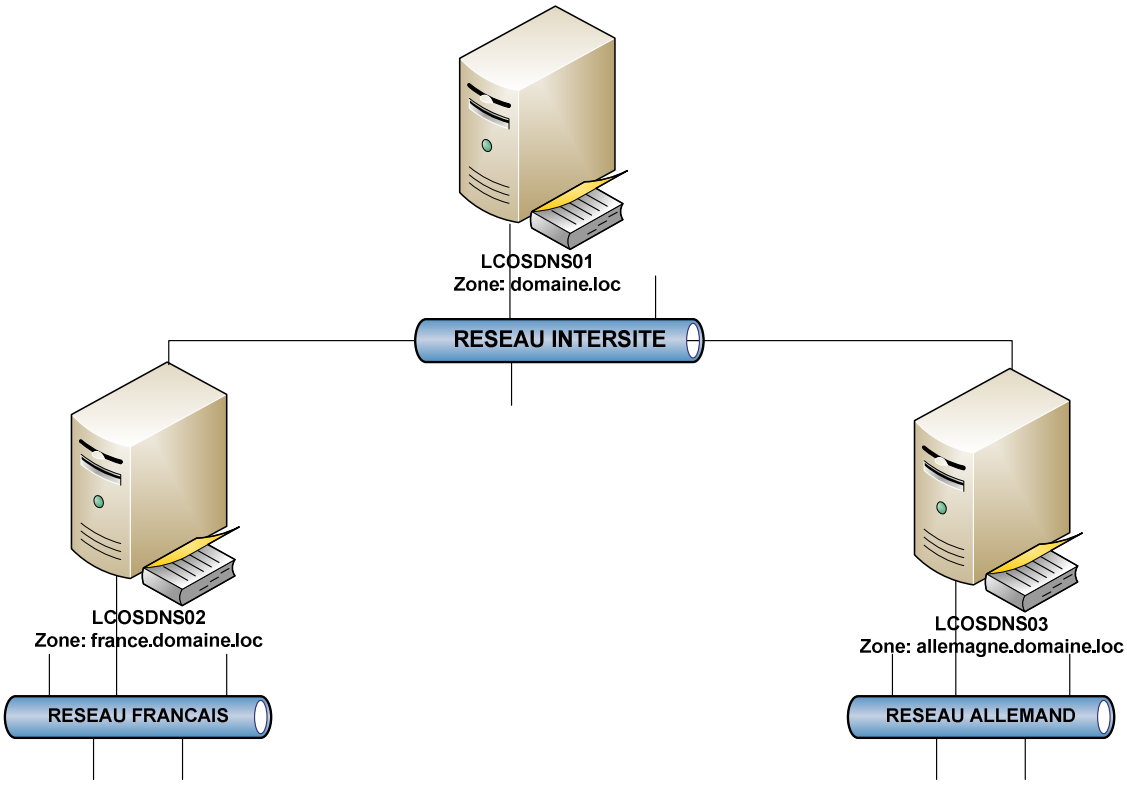
La réplication d'un serveur permet de répondre à plusieurs contrainte du système d'information comme la continuité de service ou encore la tolérance aux pannes. Cette architecture, simple à comprendre repose sur la copie de la zone d'autorité du DNS dit primaire sur le DNS dit secondaire.



Dans cet exemple les serveurs DNS permettent la résolution d'adresse entre les clients et le serveur de fichiers. Ainsi, si le serveur LCOSDNS01 vient à devenir indisponible les clients redirigeront leurs requêtes vers le second serveur DNS et permettront aux clients de pouvoir continuer à interagir avec le serveur de fichiers W2K8FLS01.
Ainsi, lorsque le premier tombe en panne ou devient injoignable le second résoudra les requêtes à sa place. Cette configuration est simple à mettre en place chez les clients. Sous Windows, il suffit simplement de remplir avec l'adresse du DNS secondaire le champs correspondant dans la configuration IP de la carte réseau et sous linux de rajouter la ligne suivante: "nameserver <adresseIpDuDnsSecondaire>". Ou alors de configurer le DHCP afin de fournir les deux adresses directement aux clients.

1.2.3 La délégation

Cette configuration, plus ou moins complexe à mettre en œuvre se base sur la division du réseau en plusieurs réseaux plus petits, chacun ayant son propre serveur DNS (pouvant être répliqué au besoin).



Dans ce cas LCOSDNS01 délègue la résolution des requêtes allemandes et françaises au DNS correspondant. Ainsi seules les requêtes intersites atteindront LCOSDNS01 et pourront être redirigées vers la zone correspondante car seul le DNS ayant autorité sur la zone peut résoudre les adresses de cette même zone. LCOSDNS01 n'a pas le contenu des zones de ses délégations et inversement. La délégation peut être assimilée à une redirection vers le bas.

1.3 Adressage

Afin de bien comprendre le fonctionnement d'un DNS nous allons clarifier certains points sur le fonctionnement de l'adressage. Pour commencer nous pouvons dissocier deux catégories d'adresses, les adresses publiques et les adresses privées. Dans notre cas seules les adresses privées nous intéressent.

1.3.1 Adresses Privées

Les adresses privées sont utilisées afin d'adresser les machines faisant partie d'un même réseau. Chaque réseau dispose de son propre adressage et de ses propres paramètres.

Nous pouvons dissocier plusieurs classes (A, B, C, D et E), chacun ayant ses propres caractéristiques. Nous ne nous intéresserons qu'aux classes A, B et C.

Une adresse IPv4 est adressée sur 32bits. Une partie définira l'adresse du réseau, appelée netID, et l'autre définira la machine, appelée hostID, le tout (adresse IP) devant définir de manière unique une machine sur le réseau. La taille du hostID et du netID est définie par le masque de sous-réseau. Il existe plusieurs types notations dont la notation cidr qui permet en un coup d'œil de connaitre le masque de sous-réseau d'une adresse et donc de son hostID et son netID.

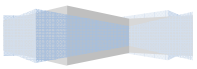
Classe	NetID | hostID	Début d'adresse	Broadcast
A	**1111.0000.0000.0000**	10.0.0.0 /8	10.255.255.255
B	**1111.1111.0000.0000**	172.(16 à 32).0.0 /16	172.(16 à 32).255.255
C	**1111.1111.1111.0000**	192.168.X.0 /24	192.168.X.255

La classe sera choisie en fonction du nombre de machine à adresser qui est donné par la formule 2^x-2 où x=le nombre de bit du hostID disponible et moins deux car nous devons retirer l'adresse du réseau et l'adresse de broadcast 255.

exemple:
Classe A:
adresse de réseau: 10.0.0.0 /8
adresse de broadcast: 10.255.255.255
nombre d'adresses disponibles: 2^24 = 16 777 214
nombre de réseaux disponibles: 2^8 = 126

1.3.2 Adresse spéciale

Il existe aussi plusieurs adresses dites spéciales, chacune ayant un rôle en particulier. La plus utilisée étant l'adresse de bouclage (loopback) 127.0.0.1.



2. Les dépendances

Avant d'installer bind nous allons devoir installer les dépendances dont il a besoin pour fonctionner.

2.1 Téléchargement

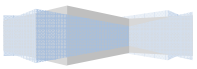
Afin de pouvoir fonctionner correctement bind aura besoin de openssl, que nous allons installer de la manière suivante:

```
$ su –
  Password: *****
# mkdir /root/SETUP
# cd /root/SETUP
# wget http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el5.rf.i386.rpm
# rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt
# wget http://ftp.gnu.org/pub/gnu/gettext/gettext-0.18.1.1.tar.gz
# rpm -K rpmforge-release-0.5.2-2.el5.rf.*.rpm
# rpm -i rpmforge-release-0.5.2-2.el5.rf.*.rpm
# yum upgrade
# yum update
# yum install openssl
```

2.2 Installation

Nous allons maintenant pouvoir installer gettext.

```
# cd /root/SETUP
# tar -xvzf gettext-0.18.1.1.tar.gz
# cd /root/SETUP/gettext-0.18.1.1
# ./configure && make && make install
```

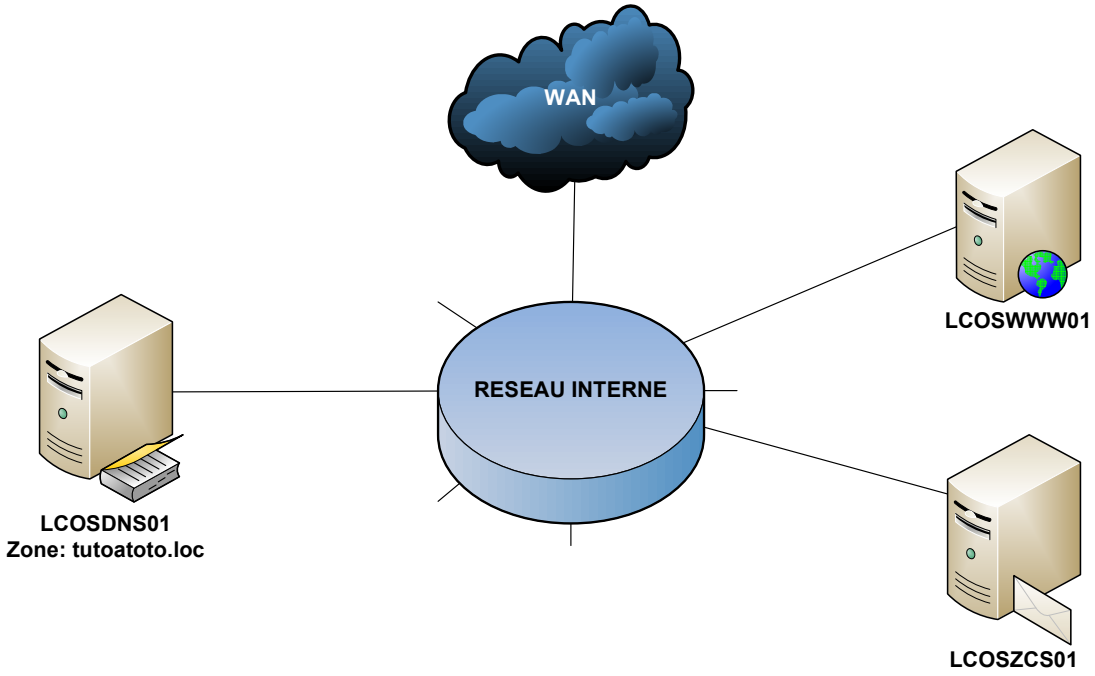
3. Implémentation

Nous allons maintenant pouvoir installer bind. Nous allons le configurer en tant que serveur standalone et en utilisant les packages disponibles dans les dépôts.

Notre serveur sera configuré de la manière suivante pour le tutoriel:

```
nom FQDN: LCOSDNS01.tutoatoto.loc
IP: 192.168.13.100 /24
zone: tutoatoto.loc
répertoire d'installation: /var/named/chroot/
serveur DNS suivant: 192.168.1.1
LCOSWWW01 (serveur web): 192.168.13.20
LCOSZCS01 (serveur de mail): 192.168.13.7
```

Voici un schéma de notre réseau:



3.1 Installation

Nous allons donc utiliser yum afin d'installer les paquets bind disponibles dans les dépôts de centos.

```
# yum install bind bind-chroot* bind-libs*
```

Nous venons d'installer les paquets concernant bind, plusieurs binaires afin de compléter notre librairie et chroot afin de pouvoir mettre en place notre structure de base chroot. bind-chroot aura pour effet de déplacer la racine au répertoire de l'application, ainsi l'arborescence interne du serveur est cachée.

Une fois l'opération terminée nous allons devoir éditer les fichiers de configuration présents sur la machine afin de lui permettre de fonctionner correctement.

4. configuration

La phase d'installation, simplifiée par l'utilisation de packages précompilés va nous obliger à configurer manuellement tous les paramètres de notre serveur afin qu'il puisse fonctionner correctement.

4.1 hosts

Le fichier hosts permet de résoudre des adresses localement. Il devra être édité et modifié de la manière suivante afin qu'il puisse résoudre son propre nom et dans certains cas d'autres machines comme un autre serveur DNS; mais avant toute modification nous allons d'abord sauvegarder le fichier que nous allons éditer afin de pouvoir revenir en arrière en cas de problème.

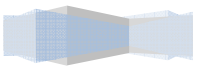
```
# mv /etc/hosts /etc/hosts.bk_dateDeLaSauvegarde
# mv /etc/hosts /var/named/chroot/
# ln -s /var/named/chroot/hosts /etc/
# vi /etc/hosts
192.168.13.100        LCOSDNS01.tutoatoto.loc          LCOSDNS01
```

Une fois le fichier édité, enregistrez le, et quitter en tapant ":wq" dans vi. Bien sûr, rappelez-vous que nous avons dû modifier l'emplacement de nos fichiers car nous utilisons la structure chroot.

4.2 resolv.conf

Maintenant afin de pouvoir communiquer sur le réseau nous allons devoir insérer dans le fichier resolv.conf le nom de domaine ainsi que le nom des serveurs de nom nécessaire à la résolution des noms.

```
# mv /etc/resolv.conf /etc/resolv.conf.bk_dateDeLaSauvegarde
# mv /etc/resolv.conf /var/named/chroot/
# ln -s /var/named/chroot/resolv.conf /etc/
# vi /etc/resolv.conf
search tutoatoto.loc
nameserver 192.168.13.100
```

4.3 named.conf

Nous allons maintenant passer à la configuration proprement dite du serveur DNS en créant et en éditant le fichier named.conf qui est le fichier de configuration du service named qui assure la résolution des noms.

```
# touch /var/named/chroot/etc/named.conf
# vi /var/named/chroot/etc/named.conf
/*********************************************************************************
*
*          Fichier de configuration du serveur DNS BIND
*
* options:
* Permet la définition des variables systèmes de l'application comme l'emplacement
* du fichier de sauvegarde (dump-file),  du fichier statistiques (statistics-file)
* des redirections éventuelles vers un serveur plus haut placé.
*
* controls:
* configure diverses contraintes de sécurité nécessaires â l'utilisation de la
* commande rndc pour administrer le service named.
*
* zone:
* permet de publier une zone, doit y figurer le rôle du serveur sur la zone, le nom
* de la zone, l'emplacement du fichier de configuration et si oui ou non on peut
* autoriser la mise à jour et si oui comment (rndc ou openssl par exemple).
*********************************************************************************/
// définition des paramètres du serveur
options {
           directory "/var/named";
           dump-file "/var/name/data/cache_dump.db";
           statistics-file "/var/named/data/named_stats.txt";
           forwarders { 192.168.1.1; };
};

// définition des paramètres de sécurité de manière à ce que rndc puisse connecter a
// partir de l'hôte local
controls {
           inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};

// définition des clés
include "/etc/rndc.key";

// définition des zones, le serveur étant maître de ces zones et permettant la mise à
// jour des zones via rndc
zone "tutoatoto.loc" IN {
           type master;
           file "/etc/zones/tutoatoto.loc.zone";
           allow-update { key "rndc-key"; };
};

// zone de reverse dns
zone "13.168.192.in-addr.arpa" IN {
           type master;
           file "/etc/zones/13.168.192.rev";
           allow-update { key "rndc-key"; };
}
```

4.4 RNDC

RNDC (Remote Nameserver Daemon Controller) est un outil qui permet de réaliser des transactions sécurisées (TSIG) avec un serveur de nom. Le mode de fonctionnement est dit à « clé partagée », c'est à dire que le client rndc et le serveur de noms doivent avoir la même clé.
rndc.key est le fichier qui nous permet d'administrer notre serveur DNS à distance via un client. Cela peut être utile par exemple lorsqu'un DHCP installé sur un autre serveur offre une adresse, il mettra automatiquement à jour le DNS si l'update est autorisé, dans notre cas oui.

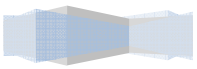
```
# vi /var/named/chroot/etc/rndc.key
/*****************************************************************************
*
*          Fichier de configuration du serveur DNS BIND
*
* key:
* permet de définir certains paramètres liés à la clé. on y trouve l'algorithme de
* chiffrement, seuls deux types existent (dsa ou hmac-md5), ainsi que secret, qui
* contient la clé cryptée.
*
* options:
* définit les paramètres de sécurité de base du serveur de nom
*****************************************************************************/
key "rndckey" {
           algorithm hmac-md5;
           secret "+Mm3VyKPub6HiitK4fCpPg==";
};

options {
           default-key "rndckey";
           default-server 127.0.0.1;
           default-port 953;
}
```

Afin de créer la clé de sécurité (secret) nous allons utiliser la commande dnssec-keygen qui est basée sur le script rndc-key_generator.sh.

```
# dnssec-keygen -a hmac-md5 -b 1024 -n 127.0.0.1 keyHmacMd5.key
```

ATTENTION: pour des raisons de sécurité, ne peut être relancé qu'avec un intervalle d'une minute environ !



4.5 Fichier de zone

Nous allons enfin pouvoir configurer nos fichiers de zone afin que notre machine puisse résoudre les adresses de notre zone.

```
# mkdir /var/named/chroot/etc/zones
# touch /var/named/chroot/etc/zones/tutoatoto.loc.zone
# touch /var/named/chroot/etc/zones/13.168.192.rev
# vi /var/named/chroot/etc/zones/tutoatoto.loc.zone
/*******************************************************************************
*
*                     Fichier de configuration d'une zone
*
* TTL:
* Time To Live, permet de définir (en seconde) le délai maximum pendant lequel un
* enregistrement pourra être gardé en cache, ce délai passé le cache sera vidé et
* les fichiers relus.
*
* SOA:
* Notre serveur de nom étant de type master sur cette zone, la première ligne doit
* être SOA, Start Of Autority, qui définira le nom du serveur, l'adresse électronique
* de l'administrateur et d'autres paramètres. Refresh, Retry, et Expire sont des
* délais, exprimés en secondes, qui vont piloter le comportement des serveurs
* esclaves. A l'expiration du délai refresh, l'esclave va entrer en contact avec le
* maître ; s'il ne le trouve pas, il essaiera de nouveau à la fin du délai retry.
* Et si, au bout du délai expiré, il n'est pas parvenu à ses fins, il considérera que
* le serveur maître n'existe plus. minimum, enfin, détermine toujours, en
* secondes, la durée de vie minimum du cache. On remarque la disposition des
* parenthèses obligatoires pour disposer les informations sur plusieurs lignes.
*
* @:
* Manière simplifiée de définir la zone, ici tutoatoto.loc.
*
* serial:
* numéro de série que l'on ne doit pas oublier d'incrémenter à chaque modification du
* fichier. Ce numéro permet aux serveurs esclaves de savoir s'il y a du nouveau, et
* de modifier le contenu de leurs bases en conséquence.
*
* NS:
* Désigne un serveur de nom (Name Server).
*
* A:
* Désigne une adresse IP classique.
*
* CNAME :
* Définit un alias qui pointera vers un serveur réel.
*
* MX:
* Donne l'adresse d'un ou plusieurs relais pour l'acheminement du courrier
* électronique. Si plusieurs serveurs sont disponibles, il faut leur affecter un
* numéro de priorité. Le plus petit numéro a la priorité la plus forte.
*******************************************************************************/
$TTL       86400
@                     IN SOA     LCOSDNS01.tutoatoto.loc.         OVampO.tutoatoto.loc. {
                                           2012020701 ; serial
                                           1H         ; refresh
                                           1M         ; retry
                                           1W         ; expiry
```

```
                                        1D )       ; minimum

@                   IN NS      LCOSDNS01.tutoatoto.loc.
LCOSDNS01  IN A     192.168.13.100
LCOSWWW01  IN A     192.168.13.20
LCOSZCS01  IN A     192.168.13.7
WWW                 IN CNAME   LCOSWWW01

@                   IN MX 0    LCOSDNS01.tutoatoto.loc.
mail                IN A       LCOSZCS01
```

Maintenant que ce fichier est configuré nous allons pouvoir mettre en place la configuration de notre zone de résolution réversive.

```
vi /var/named/chroot/etc/zones/13.168.192.rev
/*******************************************************************************
*
*          Fichier de configuration d'une zone de résolution inverse
*
* NOTES:
* La déclaration SOA est exactement la même que pour la zone dont elle est la
* reverse. @ renvoie maintenant vers 13.168.192.in-addr.arpa.
*
* PTR:
* les derniers octets des adresses des machines. Remarquons que rien n'interdit
* d'entrer des adresses complètes, 20.200.168.192.in-addr.arpa., de la même manière
* que l'on peut éventuellement entrer des noms pleinement qualifiés dans les
* enregistrements de type A.
*******************************************************************************/
$TTL       86400
@                   IN SOA     LCOSDNS01.tutoatoto.loc.         OVampO.tutoatoto.loc. {
                                          2012020701 ; serial
                                          1H         ; refresh
                                          1M         ; retry
                                          1W         ; expiry
                                          1D )       ; minimum

@                   IN NS      LCOSDNS01.tutoatoto.loc.
100                 IN PTR     LCOSDNS01.tutoatoto.loc.
20                  IN PTR     LCOSWWW01.tutoatoto.loc.
7                   IN PTR     LCOSZCS01.tutoatoto.loc.
```

Une fois toute ces configurations terminées vous n'aurez plus qu'à exécuter ces dernières lignes afin de finaliser l'installation de votre machine.

```
# chkconfig --add named
# chkconfig named on
# reboot
```

Faites bien attention car dans une structure chroot la racine du DNS est déplacée de / à /var/named/.