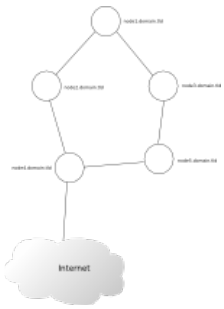
free

Installation d'un cluster ejabberd

Sommaire

Avant-propos

Cet article se veut être un guide pour la mise en place d'un serveur de messagerie instantanée, utilisant le protocole XMPP. L'architecture utilisée est une grappe de serveurs avec passerelle unique vers les protocoles externes, tel MSN. Nous informons d'ailleurs le lecteur que seule la mise en place de cette passerelle MSN est décrite dans la suite de cet article.
Le schéma ci-dessous présente brièvement le type d'architecture auquel s'applique la présente documentation :



Configuration DNS

Afin d'anticiper tout de suite tout problème lié à la résolution DNS, nous vous invitons à paramètrer l'ensemble des machines du noeud de manière à ce qu'elles puissent résoudre les noms suivant :

- `nodeX[.domain.tld]`, où `X` est un numéro de noeud
- `conference.domain.tld`, qui pointera sur le noeud numéro 1 du cluster
- `msn.domain.tld`, qui pointera sur le noeud numéro 1 du cluster (et sera utile pour la mise en place de la passerelle MSN)

Installation

L'installation est ici réalisée sur un *Debian etch*. Cependant, il ne devrait pas être trop difficile de trouver de la documentation relative à votre distribution préférée ; de plus, les paquets utilisés existent surement pour d'autres distributions.
L'installation se déroule en plusieurs étapes :

1. installation des paquets sur le premier noeud
2. paramètrage du premier noeud

3. installation de la passerelle MSN sur le premier noeud
4. installation des paquets sur les autres noeuds
5. paramètrage des noeuds pour les joindre au cluster

Installation sur le premier noeud

Pour installer le serveur `ejabberd`, lancez simplement la « classique »commande :

```
apt-get install ejabberd
```

Le serveur est désormais installé et lancé. Avant de poursuivre plus avant, arrêtons-le :

```
/etc/init.d/ejabberd stop
```

Configuration du noeud

Configuration de base

Les fichiers concernant la configuration du serveur `ejabberd` sont les suivants :

- `/etc/default/ejabberd`, qui servira à définir le nom du noeud
- `/etc/ejabberd/ejabberd.cfg`, qui servira à paramètrer le serveur (méthode d'authentification, ports utilisés, passerelles proposées, modules démarrés ...)
- `/etc/init.d/ejabberd`, le script de lancement, où nous paramètrerons les autres noeuds du cluster

Éditez le fichier `/etc/default/ejabberd` afin d'en modifier la dernière ligne comme suit :

```
export ERL_NODE="ejabberd@`hostname -s`"
```

(Notez que c'est le comportement par défaut, mais que j'ai rencontré quelques problèmes si cette ligne restait en commentaires)
Editez à présent le fichier de configuration principal du serveur `ejabberd.cfg`. Attention, ce fichier est écrit en langage Erlang.
Il est important que vous spécifiez l'administrateur du serveur ; cela se fait au moyen de l'*acl* `admin` définie en début de fichier :

```
%% Admin user
{acl, admin, {user, "admin", "domain.tld"}}.
```

Ceci définit l'utilisateur `admin@domain.tld` comme l'administrateur du serveur. C'est avec ce login (**adresse complète**) que vous pourrez vous connectez à l'interface web de gestion. Vous pouvez indiquer quels utilisateurs peuvent obtenir un accès administrateur en ajoutant par la suite des lignes comme suit :

```
{acl, admin, {user, "toto"}}.
```

où `toto` est un utilisateur enregistré.
Définissons ensuite le nom d'hôte de votre serveur (le domaine pour lequel vous souhaitez mettre en place le serveur) :

```
%% Hostname
{hosts, ["domain.tld"]}.
```

Vous pouvez laisser le reste du fichier tel qu'il est pour tester que tout fonctionne. Lancez le serveur :

```
/etc/init.d/ejabberd start
```

Ouvrez votre client XMPP favori afin de créer le compte administrateur sur le serveur. Vous pouvez alors vous rendre sur l'interface de gestion, accessible depuis votre navigateur sur l'adresse http://domain.tld:5280. Entrez alors les identifiants du compte fraîchement créé.

Configuration de l'authentification LDAP

Installation de la passerelle MSN

Installation

Nous allons installer la passerelle `pyMSNt` (`py`thon `MSN` `t`ransport) afin de permettre aux utilisateurs de rattacher un compte MSN à leur compte *jabber*. L'installation est là aussi triviale :

```
apt-get install pymsnt
```

Le fichier de configuration est le fichier `/etc/pymsnt.conf.xml`. Les options qui nous intéresse sont :

- `discoName`, pour définir le nom du service tel que présenté aux clients
- `host`, pour définir sur quel machine tourne la passerelle
- `mainServer`, pour définir l'adresse du serveur Jabber auquel se connecter
- `jid`, l'identifiant du service de transport
- `secret`, afin de définir le mot de passe permettant la connexion au serveur XMPP
- `debugLevel`, qui vous sera utile en cas de problème

Vous pouvez définir l'option `discoName` avec la valeur de votre choix. Positionnez l'option `mainServer` avec le nom de la machine hébergeant le serveur `ejabberd` ; dans notre cas, ça sera :

```
<mainServer>node1.domain.tld</mainServer>
```

Pour l'option `host`, c'est la même chose, la passerelle étant installée en local sur le serveur `ejabberd` (en fait, il s'agit de l'adresse publique sur laquelle est publiée la passerelle) :

```
<host>node1.domain.tld</host>
```

Pour le JID, nous allons tout simplement l'appeler :

```
<jid>msn.domain.tld</jid>
```

Reste encore à paramètrer le mot de passe qui sera utilisé pour la connexion sur le serveur `ejabberd`. Ce mot de passe devra également être indiqué dans le fichier de configuration du serveur `ejabberd` :

```
<secret>mon_super_mot_de_passe_incassable_ou_pas</secret>
```

Afin de pouvoir surveiller ce qui se passe, nous allons positionner le niveau de journalisation à 3 :

```
<debugLevel>3</debugLevel>
```

Vous pouvez, à votre convenance, augmenter ce niveau à 4. Attention cependant, tout le traffic (au niveau applicatif tout du moins) sera affiché. De plus, les mots de passe seront affichés en clair. Pensez donc à utiliser cette option avec parcimonie et dicernement ; n'oubliez pas non plus d'éventuellement chiffrer le fichier journal au besoin.
Il ne reste maintenant plus qu'à configurer le serveur `ejabberd` pour pouvoir connecter votre compte *jabber* à votre compte MSN habituel et discuter ainsi avec l'ensemble de vos contacts. Tout ça avec un client unique, cela va sans dire ;)

> ***Nous mettons en garde les utilisateurs du client pidgin. En effet, celui-ci n'offre pas la possibilité à l'heure actuelle de réaliser la découverte des services. Pour les utilisateurs linuxiens, nous vous proposons l'utilisation du client gajim par exemple. Il semble qu'il existe un port de ce logiciel pour les utilisateurs de MacOS. Les utilisateurs windowsiens, le client pandion offre cette fonctionnalité (via « Transports» dans le menu «Outils»)***

Intégration au serveur `ejabberd`

Éditez le fichier de configuration principal de `ejabberd`. Rechercher la section « MSN Transport» et modifiez-la de la sorte :

```
% MSN Transport
 {5557, ejabberd_service, [{access, all},
                           {host, "msn.domain.tld", [{password, "mon_super_mot_de_passe_incassable_ou_pas"}]}]},
```

Veillez bien à ce que les mots de passe soient **rigoureusement identiques** dans ce fichier et celui de `pyMSNt`. C'est en effet ce mot de passe qui permet à la passerelle de s'inscrire auprès du serveur.

> *Dans le cas où vous choisiriez d'installer la passerelle sur une autre machine, notez que vous devrez (éventuellement) paramètrer votre pare-feu de manière à ce que les deux machines puissent communiquer sur le port 5557.*

C'est maintenant l'heure de vérité :

```
/etc/init.d/pymsnt start
```

Surveillez le fichier journal afin de vous assurez qu'il n'y a pas eu d'erreur(s) au lancement du service. Si tel est le cas, connectez-vous avec votre client favori au serveur `ejabberd` puis lancer une découverte des services sur ce dernier. Votre passerelle devrait figurer en bonne place. Il

ne vous reste plus qu'à configurer votre compte MSN pour pouvoir rapatrier vos contacts.

Installation des autres noeuds du cluster

Préparation des serveurs secondaires

Toutes les manipulations décrites dans cette section concerne l'intégralité des noeuds qui composent le *cluster*, exception faite du premier noeud que nous avons paramétré précédemment. Dans un premier temps, lancez un «classqiue» :

```
apt-get install ejabberd
```

sur chacune des machines destinées à faire partie du *cluster*. Le serveur va démarrer ; pensez donc à l'arrêter avant de faire les manipulations qui suivent. La première étape consiste à recopier le *cookie* du serveur primaire. En effet, ce *cookie* est utilisé par les différents noeuds pour l'appartenance ou non au *cluster*.

> *On suppose ici qu'un serveur ssh a été correctement installé et configuré sur le noeud primaire et que* **user** *est un utilisateur ayant accès en lecture au fichier* **.erlang_cookie** *sur le noeud primaire ; on suppose également que l'utilisateur courant sur les noeuds secondaires ont les droits nécessaires pour exécuter les commandes* **sudo** *qui vont suivre.*

Lancez les commandes suivantes sur l'ensemble des serveurs du *cluster* :

```
scp user@noeud.primaire:~ejabberd/.erlang_cookie ~ejabberd
sudo chown ejabberd: ~ejabberd/.erlang_cookie
```

C'est maintenant au tour du fichier de configuration d'être copié sur tous les serveurs :

```
scp user@noeud.primaire:/etc/ejabberd.cfg .
sudo mv /etc/ejabberd/ejabberd.cfg{,.dist}
sudo mv ejabberd.cfg /etc/ejabberd
sudo chown ejabberd: /etc/ejabberd/ejabberd.cfg
```

Par précaution, on réalise une copie du fichier de configuration original. Comme on dit, on ne sait jamais... Il est désormais temps d'aller éditer le fichier **/etc/default/ejabberd**. Modifier la valeur de la variable **ERLANG_NODE** :

```
ERLANG_NODE=ejabberd@`hostname -s`
```

Editez enfin le script d'init de manière à modifier la commande lancée lors de l'appel à la fonction **start** :

```
start()
{
    cd /var/lib/ejabberd
    su $EJABBERDUSER -c "$EJABBERD -noshell -detached extra_db_nodes \"['ejabberd@noeud1', 'ejabberd@noeud2', ...]\"
    ...
}
```

où **noeud1**, **noeud2**, ... sont les **autres** noeuds du *cluster*.
Vos serveurs sont désormais quasiment prêts à être redémarrés. La dernière étape concerne la base de données «internes» du noeud primaire. En effet, **ejabberd** utilise une base de données **mnesia** pour stocker diverses informations relatives à la dernière activité des utilisateurs, aux messages hors ligne, aux historiques des salons, *etc*... Il convient donc de la rappatrier sur les noeuds secondaires avant de les redémarrer.

> ***Assurez-vous que le démon ejabberd est effectivement arrêtez. La commande 'ps -aef | grep ejabberd' vous indiquera quels processus doivent éventuellement être tués (utilisez à cet effet 'kill <pid>'.***

Tout d'abord, nous allons sauvegarder la base courante dans un répertoire **old**, créé à l'occasion dans le répertoire courant de la base de données :

```
sudo mkdir /var/lib/ejabberd/old
sudo mv /var/lib/ejabberd/* /var/lib/ejabberd/old
```

Vous devriez avoir un message «d'erreur» vous indiquant que le répertoire **old** n'a pas été déplacé (normal, on le déplacerait dans lui-même...) Le répertoire **/var/lib/ejabberd** ne doit à présent plus contenir qu'un seul répertoire : **old**. Il est à présent temps de recopier la base de données depuis le serveur primaire :

```
mkdir /tmp/ejabberd
```

```
cd /tmp/ejabberd
scp user@noeud.primaire:/var/lib/ejabberd/* .
sudo cp * /var/lib/ejabberd/
sudo chown ejabberd: /var/lib/ejabberd/*
```

La base est maintenant recopiée. Les serveurs secondaires sont désormais prêts. Cependant, il reste encore une petite manipulation à réaliser sur le serveur principal.

Fin du paramètrage du serveur principal

Connectez-vous de nouveau sur le serveur principal. Éditez-le script d'*init* du server `ejabberd` de manière à prendre en compte les serveurs secondaires dans le *cluster* :

```
start()
{
    cd /var/lib/ejabberd
    su $EJABBERDUSER -c "$EJABBERD -noshell -detached extra_db_nodes \"['ejabberd@noeud1', 'ejabberd@noeud2', ...]\"
    ...
}
```

Liens

- Site officiel **ejabberd**
- Site officiel **pyMSNt**
- Site officiel du protocole **XMPP**
- Site officiel **Jabber**
- Site d'entraide francophone sur **Jabber**
- Le client libre **pidgin**

Contacts

N'hésitez pas à me contacter sur cette adresse **wblitz_at_free_dot_fr** pour me faire part de vos remarques, suggestions ou améliorations à apporter à cet article.