

lundi 3 avril 2006



**Xavier Meunier**

# Installation plate-forme VoIP (serveur Asterisk)

# Table des matières

<b>Introduction</b>	<b>4</b>
<b>Applications</b>	<b>4</b>
<b>Fonctionnement / définitions</b>	<b>4</b>
<i>SIP</i>	<b>4</b>
<i>RTP</i>	<b>5</b>
<i>IAX</i>	<b>5</b>
<b>Installation</b>	<b>6</b>
<b>Téléchargement et compilation</b>	<b>6</b>
<i>Notes:</i>	<b>7</b>
<b>Installer les sons</b>	<b>7</b>
<b>Installation outils d'administration</b>	<b>7</b>
<i>Serveur Apache / MySQL</i>	<b>8</b>
<i>Serveur X11VNC</i>	<b>8</b>
<i>Serveur OpenSSH</i>	<b>8</b>
<i>Agent SNMP / Syslog</i>	<b>8</b>
<b>Lancer Asterisk</b>	<b>9</b>
<i>Notes:</i>	<b>9</b>
<b>Configuration d'Asterisk</b>	<b>10</b>
<b>Configuration des équipements réseau</b>	<b>10</b>
<b>Qualité de service</b>	<b>10</b>
<b>Le NAT</b>	<b>10</b>
<i>SIP et le NAT</i>	<b>11</b>
<i>IAX</i>	<b>11</b>
<i>RTP</i>	<b>11</b>
<b>Configuration par fichiers de configuration</b>	<b>11</b>
<i>asterisk.conf</i>	<b>11</b>
<i>sip.conf</i>	<b>12</b>
<i>iax.conf</i>	<b>12</b>
<i>zapata.conf</i>	<b>12</b>
<i>voicemail.conf</i>	<b>12</b>
<i>musiconhold.conf</i>	<b>12</b>
<b>Configuration en Real Time</b>	<b>13</b>
<b>Configuration à l'aide d'une interface graphique</b>	<b>14</b>

ACTOS	14
<b>Exemples de configurations</b>	<b>16</b>
<b>Les connexions SIP</b>	<b>16</b>
<b>La messagerie unifiée</b>	<b>17</b>
<i>Envoi d'un e-mail d'avertissement</i>	<b>18</b>
Configurer l'envoi d'un e-mail de notification	<b>18</b>
<b>Manager Asterisk</b>	<b>19</b>
<b>Haute disponibilité</b>	<b>19</b>
<i>Haute disponibilité au niveau application</i>	<b>19</b>
<b>Annexes</b>	<b>20</b>
<b>Commandes utiles</b>	<b>20</b>
Fichier /etc/apt/sources.list	<b>20</b>

# Introduction

Asterisk est un PBX-IP open source capable de concurrencer des systèmes commerciaux tels que les Call Manager de Cisco System. Dans ce document je vais tenter d'expliquer ma démarche pour installer Asterisk sur un Linux (distribution Ubuntu 5.04 ou 5.10 en mode normal ou serveur).

Le but de ce papier est de faire une interconnexion entre 2 PABX Asterisk en IAX (dans les réseaux nattés). Le premier sera installé sous Ubuntu et le second sous windows avec AsteriskW32 ([www.asteriskw32.org](http://www.asteriskw32.org)). Celui sous linux sera entièrement OpenSource avec notamment l'utilisation du codec iLBC (voir section qui lui est consacré), des interfaces utilisateurs GNU, etc.

Le but final est de pouvoir proposer un système d'IPBX à moindre coût capable d'être mis en production avec des contraintes fortes (PABX d'entreprise multi-sites, offre de plate-forme d'IP-Centrex, etc.)

## Applications

Le but de ce type d'application est de fournir des services téléphoniques personnalisés à des utilisateur locaux ou distants. De plus le principe de la VoIP permet de réduire les coûts téléphonique de l'entreprise (réseau convergé, capacité liaisons WAN utilisées, équipe technique unique) ou de l'utilisateur (utilisation d'Internet au lieu du réseau RTC ce qui réduit les coûts).

## Fonctionnement / définitions

Asterisk est donc une solution complète de PABX-IP logiciel fonctionnant sur divers systèmes d'exploitation comme Linux, BSD, Windows et Mac OS X. Asterisk fournit tout les services de base d'un PABX comme la connexion des postes entre eux (qu'ils soient locaux ou distants), messagerie unifiée, services Web intégrés (ex: annuaire, gestion salle de conférence, etc.), service de répondeur interactif (IVR), musique d'attente, interconnexion avec le réseau téléphonique public, etc.

Asterisk est basé comme le Cisco Call Manager sur le principe de canaux (Channels), de plan de numérotation (Dial Plan) et de contextes (search spaces pour le Cisco Call Manager).

Quand un appel arrive sur canal, par exemple le canal SIP, le but du PABX - grâce au plan de numérotation - est de trouver le canal de sortie qui peut être le même canal SIP dans le cas d'un appel en VoIP à l'intérieur du bâtiment, ce canal de sortie peut également être un des autres type de canaux géré par asterisk. Les contextes servent réduire (ou augmenter) les possibilités de sortie d'un appel. Cela peut par exemple servir pour autoriser les appels à l'international pour certains utilisateurs seulement, pour créer des services d'IVR (lors d'un appel sur un numéro, on fait appel à un contexte qui déroule le script du répondeur automatique, etc.).

Asterisk peut être de multiples canaux d'entrée/sortie qui peuvent être de type IP comme MGCP, IAX, SIP, H.323, skinny, etc. ou de type téléphonie classique avec entre autre Zap (FXS et FXO), ISDN (BRI et PRI).

### SIP

Protocole de signalisation de vidéo et voix sur IP qui est basé sur des messages en clair et fonctionnant sur le port 5600 en TCP et UDP.

## **RTP**

Le but de RTP est de fournir un moyen uniforme de transmettre sur IP des données soumises à des contraintes de temps réel (audio, vidéo, ...). Le rôle principal de RTP consiste à mettre en oeuvre des numéros de séquence de paquets IP pour reconstituer les informations de voix ou vidéo même si le réseau sous-jacent change l'ordre des paquets.

Plus généralement, RTP permet :

- d'identifier le type de l'information transportée,
- d'ajouter des marqueurs temporels et des numéros de séquence l'information transportée
- de contrôler l'arrivée à destination des paquets.

## **IAX**

Protocole de signalisation de voix/ vidéo sur IP utilisé par Asterisk (Inter Asterisk eXchange). Ce protocole fonctionne sur le port 4569 en UDP et transporte à la fois les données (voix) et la signalisation. L'intérêt principal de ce protocole est d'être fait pour traverser le NAT et qu'il est possible de créer des trunks IAX entre les serveurs dans lesquels les communications RTP sont multiplexées ainsi on économise les surcharges d'entêtes IP.

# Installation

- Système d'exploitation Linux (ici Ubuntu avec kernel 2.6)
- Asterisk 1.2.4
- Drivers ZapTel 1.2.3 utiles pour la Voicemail, IVR, etc.
- Serveur web Apache (pour interface web)
- Base SQL (MySQL, PostGre pour Asterisk en Real Time)
- PHP pour interfaces web dynamiques
- Serveur x11VNC pour prise à contrôle par interface graphique sans créer de nouvelle session gnome (% CPU en moins)
- serveur SSH (openSSH pour prise de contrôle en ligne de commande)

## Téléchargement et compilation

Il est nécessaire d'installer certains paquets pour pouvoir compiler et lancer asterisk.

- build-essentials pour la commande make
- gcc-4.0 pour compiler asterisk
- termcap-compat et libncurses5-dev pour compiler des programmes non-debian
- cvs, libssl-dev, zlib1g-dev

Ensuite pour télécharger et compiler asterisk:

- Se logger sur le serveur en tant qu'utilisateur normal créé pendant l'installation
- Taper les commandes suivante en tant que Root (avec la commande sudo) afin de télécharger Asterisk:

```
cd /usr/src
export CVSROOT=:pserver:anoncvs@cvs.digium.com:/usr/cvsroot
cvs login (password, use anoncvs)
cvs checkout asterisk
    ou cvs checkout -r v1-2-0 zaptel libpri asterisk
```

Ces commandes permettent de télécharger la toute dernière version d'Asterisk (version CVS de développement). A éviter pour la mise en production. La dernière commande permet de récupérer la version stable 1.2.0 avec les drivers zaptel et libpri. On peut également récupérer des sons et add-on en ajoutant "asterisk-addons" et "asterisk-sounds"

Il est bien sûr possible de télécharger les fichiers par HTTP ou FTP sur le serveur de Digium à l'adresse <ftp.digium.com/pub/telephony/asterisk/>

Ensuite pour compiler Asterisk, entrer les commandes suivantes:

```
cd /usr/src/asterisk
sudo make clean
sudo make
sudo make install
sudo make samples
```

la commande `make samples` permet de créer des fichiers de configuration de base. il faut exécuter cette commande avec `sudo` car il est nécessaire de créer le répertoire `/etc/asterisk`.

#### Notes:

- ▶ Chemin des fichiers de configuration: `/etc/asterisk/`
- ▶ `/usr/sbin/asterisk/` contient le démon asterisk, s'il n'y a rien c'est que la compilation c'est mal passée.
- ▶ Si la compilation est mal faite c'est que des paquets ne sont pas installés
- ▶ Asterisk peut également être installé à partir des dépôts universe d'ubuntu breezy avec la commande `apt-get install`. Sur ces dépôts sont également disponibles d'autres packages tel que les fichiers sons français, etc. La version est moins récente (1.0.9 actuellement).

### Installer les sons

Les sons se trouvent dans le répertoire `/var/lib/asterisk/sounds`. Par défaut seuls des fichiers de démonstration sont disponibles. Ils sont utilisés pour les services IVR et VoiceMail.

Pour récupérer les sons en français, télécharger le package à l'adresse <http://mxs.free.fr/asterisk/FrenchPrompts.tar> (`cd /var/lib/asterisk/` puis rapatrier le fichier avec `sudo wget`).

Il est nécessaire d'installer les sons par défaut en anglais et ceux en français qui seront placés dans les répertoires `/var/lib/asterisk/sounds/fr` et `/var/lib/asterisk/digits/fr/`.

```
Pour configurer les sons en français, il faut le faire dans le fichier
extensions.conf avec soit language=fr ou la commande SetLanguage().
```

- ▶ Les fichiers sont au format GSM par défaut, ce qui consomme plus de bande passante (64kb/s) que les formats G.729a (soumis à licence) ou iLBC (OpenSource).

### Installation outils d'administration

Ces outils bien que non sensibles sont indispensables à la bonne gestion d'un Serveur Asterisk en production.

Le serveur Apache permet des modifications simples des fichiers de configuration comme l'ajout d'un utilisateur.

Le serveur OpenSSH permet une administration à distance en ligne de commande. Cela permet également une sécurité accrue avec notamment une authentification par clé publique/clé privée. De plus il peut être envisager de faire du VNC sur du SSH. Sur une Ubuntu serveur, seul SSH peut être utilisé (sauf à installer x11) pour administrer le serveur.

L'agent SNMP permet d'envoyer des informations essentielles à un serveur SNMP telles que la consommation mémoire, CPU, réseau, les coupures réseau, etc.) qui peut à son tour les retransmettre à l'opérateur (par alarmes visuelles, e-mail, SMS, etc.).

### **Serveur Apache / MySQL**

Le serveur MySQL est nécessaire pour se servir d'Asterisk en mode Real Time (fichiers de configuration remplacés par des bases de données). Il peut bien sûr être installé sur la même machine ou sur une machine distante.

Le serveur s'installe avec la commande `apt-get install mysql-server`.

Pour plus de simplicité il est possible d'installer PHPMyAdmin (le package PHPMyAdmin intègre le serveur Apache 2, PHP et modules d'apache pour le PHP). Il suffit ensuite de se connecter sur `http://@IP_serveur/phpmyadmin` et de rentrer le mot de passe root. Pour asterisk en mode Real Time, créer un utilisateur asterisk (aucun droit d'administration) et une base du même nom.

Pour les informations à mettre dans cette base, voir le chapitre "Configuration en Real Time".

### **Serveur X11VNC**

Le serveur X11VNC permet de rediriger la session X11 en cours d'utilisation vers un hôte distant en VNC cela à l'avantage de réduire la consommation CPU lors des phases d'administrations.

### **Serveur OpenSSH**

Le serveur SSH s'installe par la commande `apt-get install openssh-server`.

- ▶ Pour lancer ou relancer le serveur: `/etc/init.d/ssh start`

Le fichier de configuration du serveur SSH: `/etc/ssh/sshd_config`

Paramètres importants:

```
PermitEmptyPasswords no
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile      %h/.ssh/authorized_keys2
AllowUsers <nom_utilisateur>
```

il faut placer les clés publiques RSA des hôtes autorisés dans le répertoire `~/.ssh/authorized_keys2`. La commande `AllowUsers` est très importante car elle permet de limiter l'accès à la machine.

Pour créer un couple de clés privée/publique, il faut utiliser la commande `ssh-keygen -t rsa`. Ensuite copier la clé publique à la suite des autres clés dans le fichier (voir ci-dessus) et vérifier que la variable `PubkeyAuthentication` est à "yes".

### **Agent SNMP / Syslog**



# Lancer Asterisk

Paramètres	Effets
-c	Mode console, reste en avant-plan (implique un -f)
-C <nomfichier>	Démarre asterisk avec un fichier de configuration spécifique
-d	Mode debug
-f	Reste en avant plan
-h	Aide, affiche toutes les options possibles
-p	Démarre asterisk en tant que processus pseudo temps-réel
-q	Mode silencieux
-v	Mode verbeux (messages consoles plus explicites en fonction du nombre de v)
-r	Pour se connecter à un serveur (local) en arrière-plan et affiche une invite de commande. Pour savoir si un serveur est lancé, utiliser la commande <code>sudo ps -A   grep asterisk</code>
-x	Avec -r permet d'exécuter une commande sur le serveur asterisk

Les 2 dernières commandes n'ont d'effets que sur un serveur déjà démarré.

## Notes:

- ▶ `asterisk -vvvvvc`: Lance un serveur Asterisk en avant-plan
- ▶ `asterisk -rx reload`: Pour relancer un serveur Asterisk en arrière-plan

# Configuration d'Asterisk

Le but de cette section est de montrer le fonctionnement général d'Asterisk et les différentes manières de le paramétrer. Pour une configuration plus avancée, se reporter au chapitre sur les exemples de configurations.

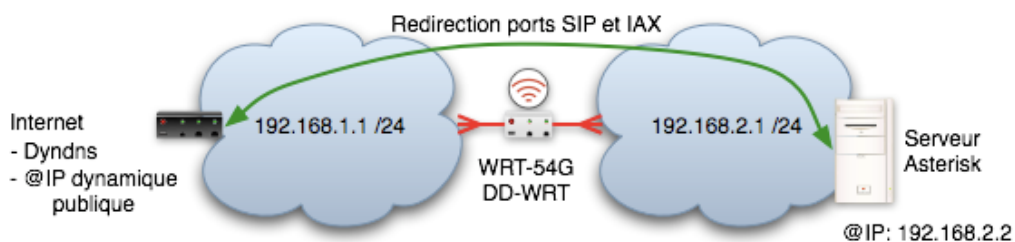
## Configuration des équipements réseau

Comme nous l'avons vu dans le fonctionnement de VoIP, la VoIP nécessite une qualité de service sur le réseau.

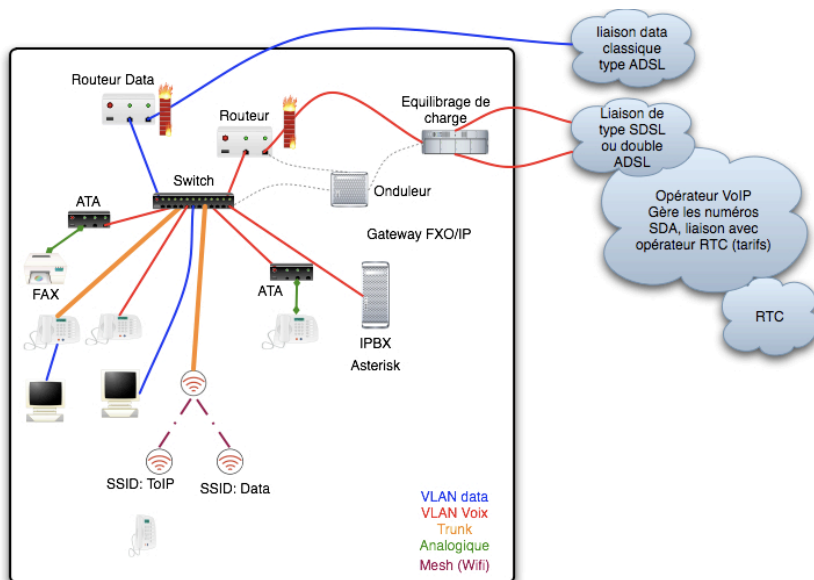
### Qualité de service

### Le NAT

Dans notre cas, le serveur Asterisk sous Ubuntu est dans un réseau privé comme le montre le schéma suivant:



Sur le routeur chargé du NAT, il faut rediriger les ports SIP (port 5060 TCP et UDP) et IAX (4569 en UDP) vers le serveur afin de pouvoir utiliser des postes distants (utile pour faire un serveur de IPBX décentralisé (IP-Centrex)).



Voici un exemple de schéma simple de réseau local informatique avec séparation des réseaux Voix et Data par des VLAN avec priorisation des flux du VLAN Voix sur le switch (champs 802.1p) et sur l'Internet avec le champs ToS des entêtes IP.

L'intérêt de rediriger les ports SIP et IAX est de permettre à des postes IP reliés à Internet de s'enregistrer sur l'IPBX. le protocole IAX est plus simple a configurer pour traverser le NAT car il peut transporter à la fois la voix et la signalisation. IAX est également très intéressant pour relié 2 IPBX asterisk.

## SIP et le NAT

## IAX

## RTP

## Configuration par fichiers de configuration

En configuration classique, les fichiers de configuration (fichiers .conf) se trouvent dans le répertoire `/etc/asterisk/`

Tout les fichiers sont du même format:

```
[class]
variable=valeur
objet => paramètres et/ou fonctions
```

### asterisk.conf

Fichier de configuration principal, contient la liste des répertoires utilisé par le processus asterisk

```
[directories]
astetcdir => /etc/asterisk
astmoddir => /usr/lib/asterisk/modules
```

```
astvarlibdir => /var/lib/asterisk
astagidir => /var/lib/asterisk/agi-bin
astspooldir => /var/spool/asterisk
astrundir => /var/run
astlogdir => /var/log/asterisk
```

**Note:** En cas de messages d'erreurs de type droits d'écriture dans des répertoire, vérifier la configuration de ces variables

### **sip.conf**

Fichier de configuration des paramètres globaux et des comptes SIP.

Ce fichier permet également de configurer les canaux SIP vers des fournisseurs (asterisk en tant que client SIP).

Le fichier comprend une partie de paramètres globaux indiqués dans la partie [general] et de comptes SIP, chaque compte étant une classe:

Pour configurer Asterisk en tant que client SIP:

```
[general]
register => user[:secret[:authuser]]@host[:port][/extension]
```

Il faut ensuite crée un compte de type peer, en effet l'enregistrement permet seulement d'être appelé, pour transmettre des appels vers l'extérieur, il faut créer un compte SIP de type peer et créer une extension (en fait une route) dans le fichier extensions.conf de type

```
exten => _0.,1,Dial(SIP/${EXTEN:1}@nom_domaine_provider,30,r)
```

Pour configurer un compte SIP (a chaque compte SIP doit correspondre une extension ou route dans le fichier extensions.conf):

```
[compte_utilisateur]
variable=valeur
```

et dans le fichier extensions.conf

```
exten => actions
```

### **iax.conf**

Comme pour le fichier sip.conf, permet de configurer des clients ou peers avec le protocole IAX qui permet une meilleure gestion de la bande passante ainsi qu'une facilité de traversée des réseaux NAT.

### **zapata.conf**

Permet de configurer les cartes d'accès FXO, BRI et PRI de digium (ou assimilé) qui utilisent les drivers zapata.

### **voicemail.conf**

Ce fichier permet de configurer les paramètres généraux de la voicemail (configuration e-mail d'avertissement), les zones de temps (pour l'heure de réception du messages) et enfin les boîtes vocales.

```
[<context_voicemail>]
n°extension => mot_de_passe,nom_utilisateur,adresse_email
```

### **musiconhold.conf**

Permet de configurer les répertoires et les classes de musique d'attente.

Par défaut asterisk utilise le player mpg123 (pas libre de droit pour utilisation commerciale) mais dans ce fichier il est possible d'en changer.

- ▶ Pour que la musique d'attente fonctionne, il faut installer mpg123 (`sudo apt-get install mpg123`)

extensions.conf

Il s'agit du fichier le plus important.

C'est dans ce fichier que l'on crée le Dial Plan, les extensions de postes, les macros, les contextes (search spaces). Il se présente sous la forme suivante:

```
[macro-<nom_macro>]
objet => parametres

[<nom_contexte>]
exten => <numero_ou_route_pattern>,<n°_seq>,parametres/fonctions
```

Plusieurs fonctions peuvent être utilisées dans ce fichier que nous verrons dans le chapitre sur les exemples de configuration.

## Configuration en Real Time

Le Real Time permet de stocker les informations de configuration (comptes SIP, etc.) dans des tables SQL (serveur MySQL, PostGre via ODBC) au lieu des fichiers de configurations standards. Cela permet des mises à jour en temps réel et la possibilité de créer des interfaces WEB personnelles.

Pour activer le mode Real Time, il faut avoir téléchargé les fichiers add-ons sur le serveur de digium (voir chapitre téléchargement).

Ensuite, lancer les commandes suivantes pour compiler et installer les fichiers nécessaires au fonctionnement en Real Time avec MySQL:

```
cd /usr/asterisk/asterisk-addons
sudo make
sudo make install
sudo cp ./configs/res_mysql.conf.sample /etc/asterisk/res_mysql.conf
```

Le fichier `/etc/asterisk/res_mysql.conf` contient les informations nécessaires pour se connecter au serveur MySQL:

```
dbhost = 127.0.0.1
dbname = asterisk
dbuser = nom_utilisateur
dbpass = mot_de_passe
dbport = 3306
dbsock = /tmp/mysql.sock
```

Le reste de la configuration se fait dans le fichier `/etc/asterisk/Extconfig.conf`.

- Le problème de la configuration en Real Time c'est qu'elle ne permet pas l'utilisation de toutes les fonctionnalités comme la traversée du NAT.

## Configuration à l'aide d'une interface graphique

Il peut être intéressant d'avoir une interface graphique pour manager un serveur asterisk, bien sûr une interface graphique permet moins de souplesse d'une configuration directe des fichiers mais avec une interface Web par exemple, une programmation personnelle peut être faite pour des besoins spécifiques.

Il existe 5 types d'interfaces pour manager entièrement asterisk:

- Interface Utilisateur
- Interface Configuration
- Interface Administrateur
- Interface Statut
- Interface facturation (important surtout dans le cadre d'IP-centrex)

Dans ce chapitre nous allons voir comment s'installe et se configure plusieurs interfaces graphique avec les points forts et faibles de chacune.

### ACTOS

Actos (Asterisk Configuration Tool Open Source) est une application qui fonctionne en client/serveur. Un serveur s'installe sur le serveur asterisk et envoie les fichiers de configuration vers un client qui se connecte à ce serveur via une socket TCP. Le client peut être soit sur Windows soit sur Linux.

La première étape consiste à installer le serveur Actos pour cela seul le package pour le langage de programmation Python est nécessaire.

```
sudo apt-get install python2.4-gtk2
sudo apt-get install python2.4-dev
```

Pour installer le serveur Actos, il faut d'abord récupérer le fichier `actos_server-2.23.tar`, le décompresser dans le répertoire personnel et l'installer:

```
tar -xvf actos_server-2.23.tar.gz
sudo ./launch_me_to_install
```

Cet exécutable placera le démon `actos_server.py` dans le répertoire `/usr/bin/` et les fichiers dans `/usr/share/actos/`. Il y aura également un nouvel utilisateur qui aura les droits sur les fichiers de configuration d'asterisk.

Pour le premier lancement, il faut démarrer le serveur avec la commande:

```
actos_server.py -manage
```

Il faudra créer une autorisation (un mot de passe) afin de sécuriser l'échange entre le serveur et le client.

Ensuite lors des lancements suivants il faudra se logger en utilisateur `actos` et lancer le serveur:

```
actos_server.py -run [numero_port]
```

- ▶ Par défaut le port utilisé est le 8007

Pour installer un client Actos sur Windows, il faut installer gtk pour windows et décompresser (zip) l'archive astos pour Windows. Ensuite lancer le fichier `wininstall.bat`.

Lors du premier lancement, le programme essayera de récupérer des fichiers en local, il y aura donc des erreurs. Pour récupérer les fichiers sur le serveur asterisk, il faut aller dans le menu "*Import Asterisk configuration*" et choisir "*Import from a remote machine running Actos-server*". Il faut alors rentrer l'adresse IP du serveur asterisk, le port (8007 par défaut), le mot de passe (celui de l'autorité ajoutée lors du premier lancement du serveur) et laisser Remote directory par défaut.

# Exemples de configurations

Le but de cette section est de montrer comment utiliser les fonctionnalités d'asterisk pour en faire un IPBX répondant aux demandes des clients.

## Les connexions SIP

Que ce soit pour des connexions avec asterisk en tant que serveur (pour compte poste téléphoniques locaux par exemple) ou en tant que client (pour s'interconnecter avec un provider ou un second PABX sur un autre site), c'est dans le fichier sip.conf que la configuration se fait:

```
***** Protocol definitions *****
[general]
;----- general setup
bindaddr = 0.0.0.0      [1]
tos = lowdelay
externhost = homemxs.dyndns.org  [2]
externip = homemxs.dyndns.org
localnet = 192.168.2.0/255.255.255.0

;----- codecs setup
allow = all
;----- other options
srvlookup = yes
context = default [3]
bindport = 5060
;----- register to peers

***** Users *****
[xavier]
accountcode = xavier
callerid = "Xavier Meunier" <1001>
language = fr
nat = no
dmthmode = rfc2833
host = dynamic
mailbox = 1001@default
mailbox = 1001@default
secret = counter
context = default
qualify = yes
type = friend
disallow = all
allow = ilbc
allow = G729
allow = GSM

[marielle]
accountcode = marielle
callerid = "Marielle Kermarrec" <1002>
```



```
dmtfmode = rfc2833
qualify = yes
nat = no
mailbox = 1002@default
secret = vertigo
context = default
host = dynamic
type = friend
language = fr
```

1. bindaddress: adresse IP sur laquelle le serveur écoute les requêtes qui lui sont envoyées. Si le routeur n'a qu'une seule interface réseau, il faut laisser par défaut (0.0.0.0).
- 2.
3. context de recherche pour les appels entrants, il faut donc s'assurer que les extensions qui doivent être joignables sont bien dans ce context.

## La messagerie unifiée

Pour créer des boîtes de messagerie vocale aux utilisateurs, il faut indiquer un identifiant de messagerie dans les comptes (sip,iax,zaptel) avec la commande `mailbox=<identifiant>[@context_voicemail]`

Ensuite il faut créer un compte dans le fichier `voicemail.conf`:

```
[<context_voicemail>]
n°extension => mot_de_passe,nom_utilisateur,adresse_email
```

Dans ce fichier on peut également créer des zones ce qui permet par exemple de régler les fuseaux horaires, le format de l'heure, etc. Ces zones se créent dans la classe

```
[zonemessages]
```

Exemple simple, numéroté le 123 pour accéder au menu principal de la messagerie unifiée. L'utilisateur doit ensuite rentrer son login et mot de passe. Dans `extensions.conf`:

```
exten => 123,1,VoiceMailMain()
```

Un exemple plus poussé et plus pratique pour l'utilisateur serait:

```
exten => ****99,1,VoiceMailMain(${CALLERID})
```

Ainsi pour l'utilisateur 1000, en numérotant le 100099, l'utilisateur accède directement à la demande de mot de passe. Par contre il faut que dans le compte de l'utilisateur la variable `callerid` soit indiquée (ex: `callerid = "xavier meunier" <1000>`).

- Les messages sont stockés dans le répertoire `/var/spool/asterisk/voicemail/<context>/<n°extension>`

### **Envoi d'un e-mail d'avertissement**

La messagerie intégrée à asterisk permet l'envoi un e-mail lorsqu'un message vocal arrive. Il est possible de mettre le message en pièce jointe au mail (par exemple au format wave). La configuration se fait dans le fichier voicemail.conf

### **Configurer l'envoi d'un e-mail de notification**

Le but de la messagerie unifiée est d'informer par e-mail l'utilisateur de la réception d'un message vocal. Asterisk permet de le faire et la configuration se trouve dans le fichier voicemail.conf.

# Manager Asterisk

Le but de ce chapitre est d'étudier la possibilité de mise en production et donc la facilité et les possibilités de management et d'exploitation de ce logiciel.

## Haute disponibilité

Pour qu'un serveur asterisk puisse être utilisé en production, il doit être dans un environnement dit de haute disponibilité c'est-à-dire avec une redondance à tous les niveaux intervenant dans le bon fonctionnement du service ToIP fournit par Asterisk.

- Electricité
- Réseau
- Serveur

Pour l'électricité, un onduleur en cas de panne peut être suffisant.

En ce qui concerne le réseau, plusieurs facteurs doivent être pris en compte

### Haute disponibilité au niveau application

Elle peut se faire par l'utilisation de 2 applications: heartbeat et cron.

La 1ère permet de tester la disponibilité du serveur maître et dans le cas d'une panne de celui de relayer avec le second serveur (ou plus). Il existe d'autres solutions comme le logiciel ultramonkey qui permet lui aussi de créer un cluster.

Les fichiers de configurations peuvent ainsi être échangés entre les 2 serveurs par tes scripts cron.

# Annexes

## Commandes utiles

voici des commandes pouvant aider pour l'administration du serveur:

Commande	
top	Permet d'afficher la charge CPU (touche q pour quitter)
free -m	Permet d'afficher les informations RAM et Swap
df -h	Permet d'afficher les informations sur le disque dur
apt-cache search <mot>	Permet de chercher un package Debian contenant le mot <mot>
who	Pour connaître les utilisateurs connectés sur le serveur
sudo -s	Sous Ubuntu, permet d'avoir accès à un terminal root
find / -name <mot>	Permet de chercher un fichier dont le nom contient <mot>
more .*   grep <mot>	Exemple de tunnel redirigeant la sortie du 1er vers l'entrée du second.  Ici on demande d'afficher le contenu de tout les fichiers du répertoire courant et de filtrer avec grep en affichant que les lignes avec le mot <mot>.

## Fichier /etc/apt/sources.list

Ce fichier contient la liste des serveurs de dépôts sur lesquels se trouvent les packages debian (.deb) nécessaire au fonctionnement d'asterisk. Pour trouver tout les packages nécessaires à la compilation de voici la liste des dépôts nécessaires:

```
deb http://fr.archive.ubuntu.com/ubuntu breezy main restricted
# deb-src http://fr.archive.ubuntu.com/ubuntu breezy main restricted

deb http://fr.archive.ubuntu.com/ubuntu breezy-updates main restricted
# deb-src http://fr.archive.ubuntu.com/ubuntu breezy-updates main restricted

deb http://fr.archive.ubuntu.com/ubuntu breezy universe
deb http://fr.archive.ubuntu.com/ubuntu breezy multiverse
# deb-src http://fr.archive.ubuntu.com/ubuntu breezy universe
deb http://fr.archive.ubuntu.com/ubuntu breezy-backports main restricted
universe multiverse
# deb-src http://fr.archive.ubuntu.com/ubuntu breezy-backports main
restricted universe multiverse

deb http://security.ubuntu.com/ubuntu breezy-security main restricted
# deb-src http://security.ubuntu.com/ubuntu breezy-security main restricted

deb http://security.ubuntu.com/ubuntu breezy-security universe
# deb-src http://security.ubuntu.com/ubuntu breezy-security universe
```

