

# **Installer un serveur VOIP Asterisk et ses clients**

Par LoiselJP  
Le 20/07/2013

# 1 Objectifs

La VOIP (Voice Over Internet Protocol), la voix par IP, presque tout le monde en a entendu parler et encore plus de personnes l'utilisent au travers des abonnements Internet ou en entreprise.

La mettre en œuvre est une autre histoire. Le sujet est assez mal connu et par ce fait paraît compliqué. Loin de là l'idée de dire que la VOIP est une formalité à mettre en œuvre, s'il est facile de créer un serveur de VOIP (c'est la première étape du tutoriel), configurer un serveur en mode avancé demande de bonnes connaissances notamment dans le domaine du réseau.

Ce tutoriel n'a pas pour objectif de faire un grand exposé sur le principe de fonctionnement, les protocoles, l'UDP, le transport de données ou le modèle OSI en général, ce sujet est laissé aux universitaires. L'objectif de ce tutoriel est de mettre en place un serveur VOIP et d'y connecter des clients d'origines différentes.

Dans un second temps, ce serveur sera configuré de manière un peu plus avancée.

C'est le serveur Asterisk qui sera mis en œuvre dans ce document.



## 2 Sommaire

1 Objectifs.....	2
2 Sommaire.....	3
3 Prérequis .....	4
4 Installation du serveur.....	4
Installer les outils d'administration.....	5
Installer et utiliser SSH.....	6
Placer une adresse IP fixe.....	6
Améliorer l'affichage.....	8
5 Installer le serveur Asterisk .....	8
Ajouter des comptes utilisateurs.....	10
6 Installer les clients Linux.....	15
7 Installer les clients Windows.....	18
Configurer Ekiga pour Windows .....	19
Configurer le compte .....	20
8 Tester les clients.....	21
9 Créer un script pour la création de client.....	23
10 Configuration et utilisation avancée d'Asterisk.....	25
Configuration de la messagerie .....	27
11 Conclusion .....	30
©Propriété.....	31
Licence.....	31

### 3 Prérequis

Ce document est principalement destiné à la mise en œuvre d'un serveur Asterisk, il n'est donc pas question ici de voir ou revoir toutes les bases de Linux. Pour appliquer cette opération, on suppose alors que sont acquis :

- L'installation de base d'un poste de travail linux,
- L'installation de logiciel:
  - Principe des paquets et leurs dépendances,
  - Les "dépôts" de paquets,
  - Principe d'installation manuelle,
- Les principales commandes du shell,
- Utilisation du programme man et l'aide des commandes (si vous ne savez pas appliquer ce qui est demandé),
  
- Le lecteur aura également quelques connaissances de base sur le fonctionnement d'une station de travail Windows (rien de plus que de savoir installer et configurer un programme, voir connecter la station de travail sur un réseau).

L'installation du serveur ne sera abordée que de très loin, quant aux postes clients on suppose qu'ils seront déjà installés, il ne restera alors qu'à y installer les clients Voip.

Pour le besoin de ce tutoriel, le serveur utilisé ne demande aucune particularité, il va de soi qu'une connexion réseau correcte, voir bonne, s'impose. Il sera nécessaire de disposer de plusieurs postes, 1 poste serveur, 1 client Windows est le strict minimum, dans le cas d'une utilisation du serveur comme poste client, il sera nécessaire d'y installer l'interface graphique...

Dans le cadre de ce tutoriel, les postes utilisés sont : 1 serveur, 2 postes clients Linux, 1 poste client Windows.

### 4 Installation du serveur

Asterisk est disponible sur un grand nombre de distributions. Chacun choisira la distribution de son choix, dans le cas présent toute l'installation est donnée pour une version à base de Debian (on recherche un serveur et non un poste de travail).

Cette version de Linux est plus destinée à un usage de serveur. D'autant plus qu'aucune interface graphique n'est nécessaire (sauf dans le cas d'une utilisation du serveur comme poste client).

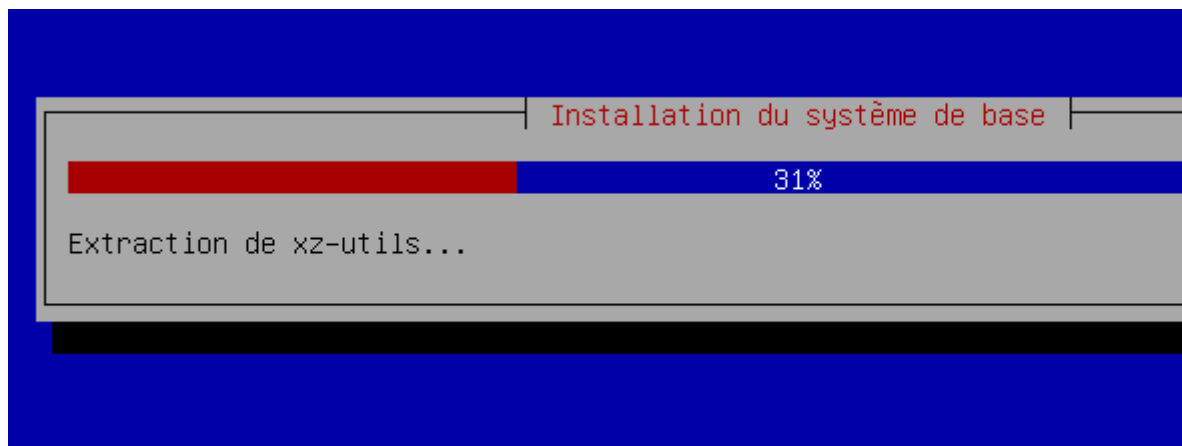
Dans le cadre de ce tutoriel, on effectuera une installation suivant ce modèle:

# Installation à partir d'un CDrom netinstall Debian 7 (la dernière en date au jour de la création de ce tutoriel)

# Avec en sélection des paquets suivants: aucun.



L'installation se fera suivant un modèle classique.



Ce tutoriel ne traite pas de la configuration du serveur, l'administrateur de ce serveur jugera alors à son niveau, l'utilité d'utiliser des partitions séparées, installer un pare-feu, un antivirus... (Un serveur VOIP est souvent connecté au Web...).

## Installer les outils d'administration

Bien que l'ensemble des manipulations peut être effectué directement sur le serveur sans outils d'administration particulier, ce tutoriel, veut présenter un système complet et fonctionnel. SSH est un outil indispensable pour l'administrateur qui n'aura pas toujours l'occasion d'être directement en contact physiquement avec le serveur.

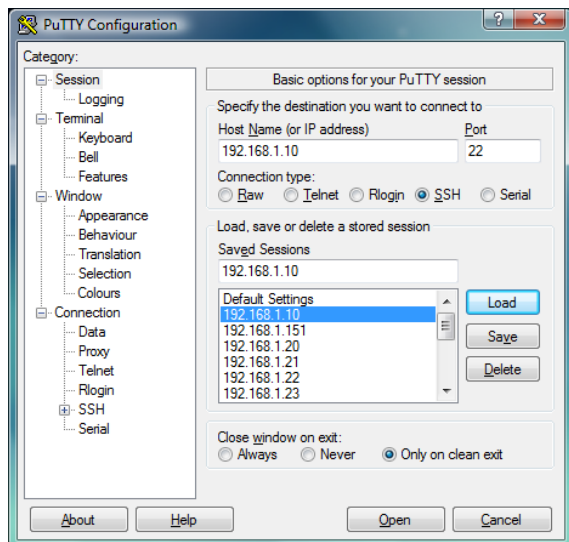
## Installer et utiliser SSH

SSH est une solution de prise en main à distance sécurisée, toutes les commandes pourront être traitées par ce moyen. D'autant plus que le serveur ne dispose pas d'interface graphique...

Pour installer OpenSSH :

```
apt-get install openssh-server
```

Putty qui est un logiciel libre vous permet cela en toute simplicité:



Depuis un autre poste, par une fenêtre de commande, vous pourrez gérer votre serveur sans besoin de travailler directement dessus.

Dès lors que l'on utilise une connexion SSH il sera aisé d'utiliser le copier/coller en prenant directement les commandes de ce tutoriel.

## Placer une adresse IP fixe

En fonction de l'organisation de la structure dans laquelle se trouve le serveur, placer une adresse IP fixe peut être intéressant. D'autant que le serveur derrière un serveur DHCP peut voir son adresse IP régulièrement modifiée, le serveur n'ayant pas d'interface graphique, la plupart des accès se feront par le réseau (mais c'est une question de choix). Le serveur sera également atteint par les clients depuis cette adresse IP ou depuis le nom DNS (s'il est inscrit sur les DNS, cela ne posera aucun souci d'avoir une IP dynamique).

Pour placer une adresse IP fixe :

```
nano /etc/network/interfaces
```

Placer l'ip fixe dans le fichier de configuration

```
# Interface reseau de bouclage
```

```
auto lo

iface lo inet loopback

address 127.0.0.1

netmask 255.0.0.0


# carte reseau en ip statique (ou auto)

allow-hotplug eth0

iface eth0 inet static

# <- a adapter suivant le besoin

address 192.168.1.10

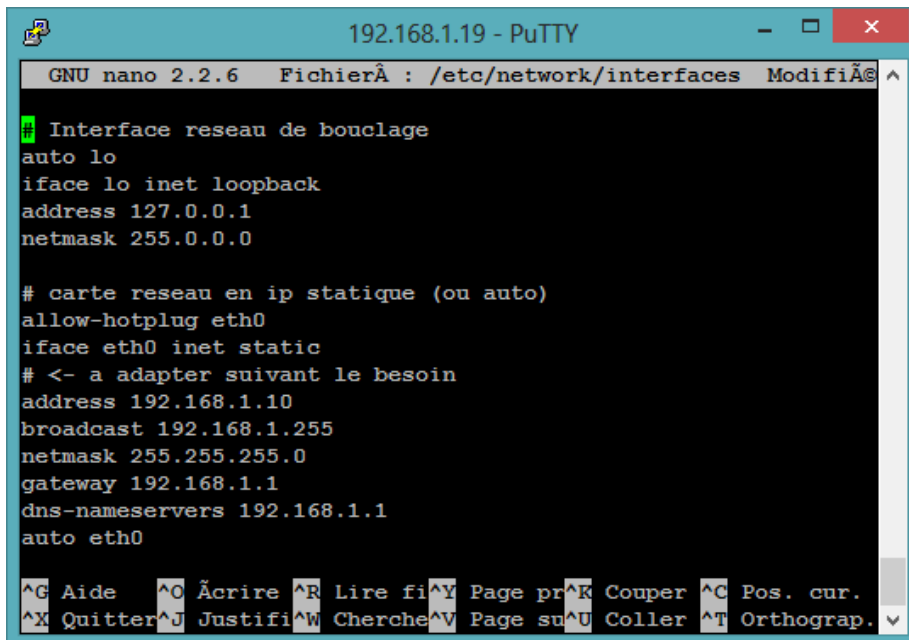
broadcast 192.168.1.255

netmask 255.255.255.0

gateway 192.168.1.1

dns-nameservers 192.168.1.1

auto eth0
```



```
192.168.1.19 - PuTTY
GNU nano 2.2.6 Fichier : /etc/network/interfaces Modification
Interface reseau de bouclage
auto lo
iface lo inet loopback
address 127.0.0.1
netmask 255.0.0.0

# carte reseau en ip statique (ou auto)
allow-hotplug eth0
iface eth0 inet static
# <- a adapter suivant le besoin
address 192.168.1.10
broadcast 192.168.1.255
netmask 255.255.255.0
gateway 192.168.1.1
dns-nameservers 192.168.1.1
auto eth0

^G Aide ^O Ãcrire ^R Lire fi ^Y Page pr ^K Couper ^C Pos. cur.
^X Quitter ^J Justifi ^W Cherche ^V Page su ^U Coller ^T Orthograp.
```

Puis redémarrer l'interface réseau

```
/etc/init.d/networking restart
```

Ou

```
service networking restart
```

L'IP a été placée comme on le voulait.

## Améliorer l'affichage

Vous l'aurez peut-être rencontré certaines versions de Linux proposent les listes en couleur et quelques commandes comme « ll » qui remplace aisément « ls -l ».

Ces fonctions sont comprises dans le système mais non activées par défaut « ll » est un alias de « ls -l ». Ces fonctions pourront être activées à partir du fichier « .bashrc » du dossier de l'utilisateur (ici, c'est le compte root qui nous intéresse) :

```
nano /root/.bashrc
```

```
GNU nano 2.2.6      Fichier : .bashrc

# ~/.bashrc: executed by bash(1) for non-login shells.

# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
# PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
# umask 022

# You may uncomment the following lines if you want 'ls' to be colorized:
# export LS_OPTIONS='--color=auto'
# eval "`dircolors`"
# alias ls='ls $LS_OPTIONS'
# alias ll='ls $LS_OPTIONS -l'
# alias l='ls $LS_OPTIONS -lA'
#
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'
```

Il suffira de décommenter les lignes voulues...

```
GNU nano 2.2.6      Fichier : .bashrc      Modifié

# ~/.bashrc: executed by bash(1) for non-login shells.

# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
umask 022

# You may uncomment the following lines if you want 'ls' to be colorized:
export LS_OPTIONS='--color=auto'
eval "`dircolors`"
alias ls='ls $LS_OPTIONS'
alias ll='ls $LS_OPTIONS -l'
alias l='ls $LS_OPTIONS -lA'
#
# Some more alias to avoid making mistakes:
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
```

Les couleurs et commandes seront disponibles à la prochaine connexion...

```
debian64-Asterisk:~# cd /
debian64-Asterisk:/# ls
bin      etc      lib      media    proc     sbin     sys      var
boot     home     lib64    mnt      root     selinux  tmp      vmlinuz
dev      initrd.img  lost+found  opt      run      srv      usr
debian64-Asterisk:/# _
```

## 5 Installer le serveur Asterisk



Asterisk est un serveur de VOIP simple et complet.

Chaque poste client s'y connecte au travers d'un logiciel comme « Ekiga » sous Linux. Sous Windows Ekiga est également disponible, alors qu'on pourra lire sur l'Internet qu'il est extrêmement difficile de configurer un client Voip sur Windows en comparaison avec d'Ekiga sous Linux. Le logiciel « 3CX VOIP Phone for Windows » est également disponible pour les stations de travail Windows.

Dans un prochain chapitre (Configuration et utilisation avancée d'Asterisk) on verra qu'il comment gérer un répondeur, une file d'attente, de la musique en attente de prise de ligne...

Après mise à jour du serveur l'installation d'Asterisk pourra être effectuée:

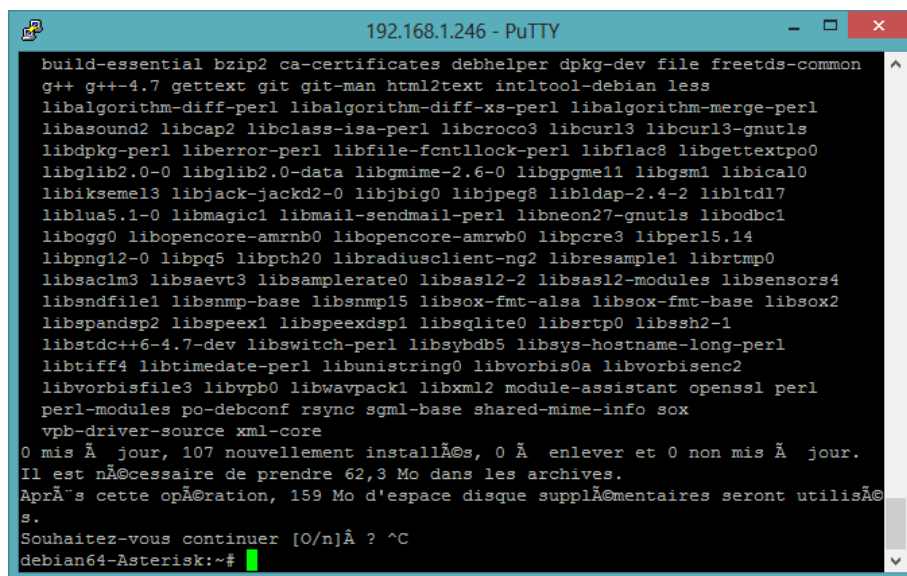
```
apt-get update
apt-get upgrade
```

Suivant les cas (versions, méthode d'installation du serveur, installation et méthode choisie pour installer Asterisk..) on pourra y ajouter quelques dépendances :

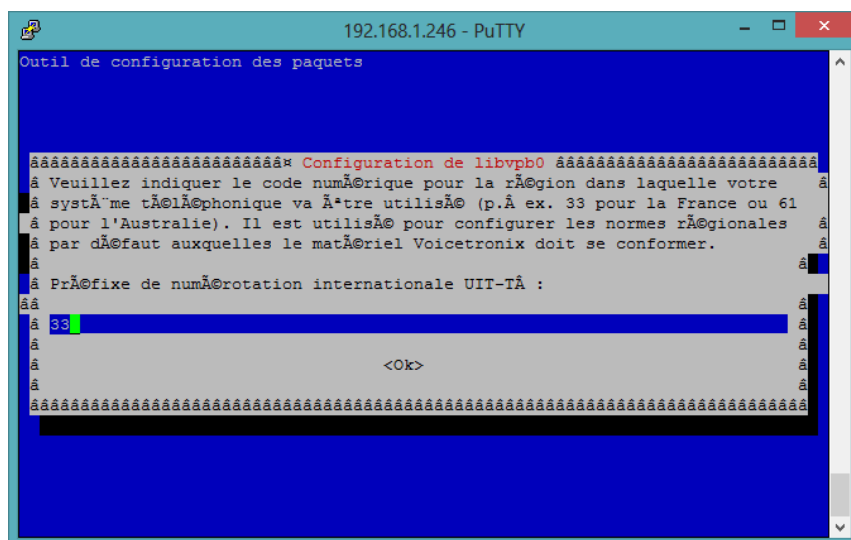
```
apt-get install build-essential libxml2-dev libncurses5-dev linux-headers-`uname -r` libsqlite3-dev libssl-dev make
cpp
```

Installer ensuite Asterisk (attention la commande est compliquée 🤖) :

```
apt-get install asterisk
```



```
192.168.1.246 - PuTTY
build-essential bzip2 ca-certificates debhelper dpkg-dev file freetds-common
g++ g++-4.7 gettext git git-man html2text intltool-debian less
libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
libasound2 libcap2 libclass-isa-perl libcroco3 libcurl3 libcurl3-gnutls
libdpkg-perl liberror-perl libfile-fcntllock-perl libflac8 libgettextpo0
libglib2.0-0 libglib2.0-data libglib2.0-0 libgpgme1 libgsm1 libical0
libiksemel3 libjack-jackd2-0 libjbig0 libjpeg8 libldap-2.4-2 libltdl7
liblua5.1-0 libmagic1 libmail-sendmail-perl libneon27-gnutls libodbc1
libogg0 libopencore-amrnb0 libopencore-amrwb0 libpcre3 libperl5.14
libpng12-0 libpq5 libpth20 libradiusclient-ng2 libresample1 librtmp0
libsac1m3 libsaevt3 libsamplerate0 libsasl2-2 libsasl2-modules libsensors4
libsndfile1 libsnmp-base libsnmp15 libsox-fmt-alsa libsox-fmt-base libsox2
libspandsp2 libspeex1 libspeexdsp1 libsqlite0 libstrtp0 libssh2-1
libstdc++6-4.7-dev libswitch-perl libsybdb5 libsys-hostname-long-perl
libtiff4 libtimedate-perl libunistring0 libvorbis0a libvorbisenc2
libvorbisfile3 libvpb0 libwavpack1 libxml2 module-assistant openssl perl
perl-modules po-debconf rsync sgml-base shared-mime-info sox
vpb-driver-source xml-core
0 mis Ã jour, 107 nouvellement installÃs, 0 Ã enlever et 0 non mis Ã jour.
Il est nÃcessaire de prendre 62,3 Mo dans les archives.
AprÃs cette opÃration, 159 Mo d'espace disque supplÃmentaires seront utilisÃs.
Souhaitez-vous continuer [O/n]Ã ? ^C
debian64-Asterisk:~#
```



```
192.168.1.246 - PuTTY
Outil de configuration des paquets
Configuration de libvpb0
Veillez indiquer le code numÃrique pour la rÃgion dans laquelle votre
systÃme tÃlÃphonique va Ãtre utilisÃ (p.Ã ex. 33 pour la France ou 61
pour l'Australie). Il est utilisÃ pour configurer les normes rÃgionales
par dÃfaut auxquelles le matÃriel Voicetronix doit se conformer.
PrÃfixe de numÃrotation internationale UIT-TÃ :
33
<Ok>
```

Sauf pour le cas où le serveur est hébergé en dehors de la France, aucune information particulière n'est nécessaire...

Et c'est fait !

Certains sites préconiseront de passer par le téléchargement des paquets puis de procéder à une installation manuelle :

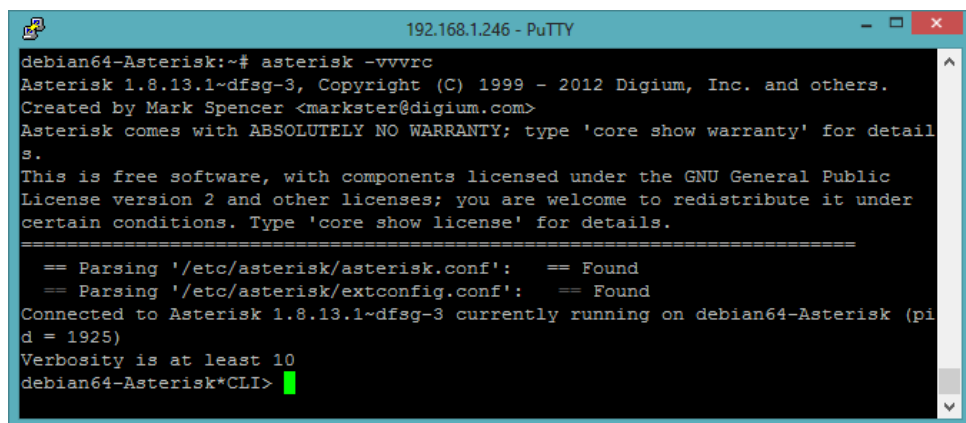
```
cd /tmp
wget http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-10.7.0.tar.gz
tar xvfz asterisk-10.7.0.tar.gz
cd asterisk-10.7.0
./configure
make menuselect
```

Après sélection des modules (« menuselect ») procéder à l'installation :

```
make
make install
make samples
make config
```

A ce stade, Asterisk doit être démarré, on pourra vérifier son fonctionnement avec la commande suivante qui donnera accès à la console. (CTRL+c pour quitter)

```
asterisk -vrc
```

A screenshot of a terminal window titled '192.168.1.246 - PuTTY'. The terminal shows the output of the 'asterisk -vvvrc' command. It displays the Asterisk version (1.8.13.1-dfsg-3), copyright information (1999-2012 Digium, Inc. and others), and the creator (Mark Spencer). It also shows the license (GNU General Public License version 2) and the location of the configuration files. The terminal ends with the prompt 'debian64-Asterisk\*CLI>' and a green cursor.

```
debian64-Asterisk:~# asterisk -vvvrc
Asterisk 1.8.13.1-dfsg-3, Copyright (C) 1999 - 2012 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
== Parsing '/etc/asterisk/asterisk.conf': == Found
== Parsing '/etc/asterisk/extconfig.conf': == Found
Connected to Asterisk 1.8.13.1-dfsg-3 currently running on debian64-Asterisk (pid = 1925)
Verbosity is at least 10
debian64-Asterisk*CLI>
```

## Ajouter des comptes utilisateurs

Pour ajouter un nouveau compte utilisateur, ouvrir le fichier suivant :

```
nano /etc/asterisk/sip.conf
```

Le fichier de configuration (donné pour exemple) compte pas moins de 1300 lignes, chaque paramètre et chaque ligne sont très largement commentées, un peu de lecture permettra de comprendre rapidement les différentes possibilités offertes par le serveur.

Dans cette première phase de configuration, le serveur sera utilisé dans sa plus simple configuration : permettre la communication entre les clients.

Commencer par mettre les sons par défauts en Français. Pour cela, remplacer « ;language=en » par

language=fr

(Penser à bien retirer le point-virgule « ; »)

Jusque-là il n'y aura rien de compliqué : ajouter en fin de fichier de configuration la définition des clients

```
[general]
hasvoicemail = yes
hassip = yes
hasiax = yes
callwaiting = yes
threewaycalling = yes
callwaitingcallerid = yes
transfer = yes
canpark = yes
cancallforward = yes
callreturn = yes
callgroup = 1
pickupgroup = 1
nat = yes

[template](!)
type=friend
host=dynamic
dtmfmode=rfc2833
disallow=all
allow=ulaw
context = travail

[001](template)
username=client1
secret=mot_de_passe_client1

[002](template)
username=client2
secret=mot_de_passe_client2

[003](template)
username=client3
secret=mot_de_passe_client3
```

```
192.168.1.246 - PuTTY
GNU nano 2.8.1 Fichier : ...sip.conf Modifié
threewaycalling = yes
callwaitingcallerid = yes
transfer = yes
canpark = yes
cancallforward = yes
callreturn = yes
callgroup = 1
pickupgroup = 1
nat = yes

[template] (!)
type=friend
host=dynamic
dtmfmode=rfc2833
disallow=all
allow=ulaw
context = travail

[001] (template)
username=client1
secret=client1

[002] (template)
username=client2
secret=client2

[003] (template)
username=client3
secret=client3
```

**Note :** On veillera à ne pas laisser d'espace entre le numéro sip et le modèle... dans le cas contraire le client ne serait pas reconnu !

```
allow=ulaw
context = travail

[001] (template)
username=client1
secret=client1

[002] (template)
username=client2
secret=client2

[003] (template)
username=client3
secret=client3
```

Dans l'exemple ci-dessus, nous utilisons un modèle général (qui contient les caractéristiques générales des utilisateurs), puis un modèle (template) indiquant les caractéristiques de ce modèle avec un nom « travail ».

Dans ce modèle « friend » indique que le client pourra recevoir et appeler, le nom d'utilisateur correspondra au login, l'hôte « dynamic » indiquera que l'utilisateur peut se connecter depuis n'importe quelle adresse IP.

Puis chaque client commencera par son numéro « sip » (numéro d'appel) et sera rattaché au modèle « template ».

Bien que la confidentialité du mot de passe sur une connexion téléphonique soit d'un intérêt secondaire, il conviendra de protéger le fichier de configuration en conséquence (il est en accès root uniquement par défaut), comme on peut le voir dans cet exemple ci-dessus, les mots de passe sont en clair dans le fichier configuration d'Asterisk (on pourra ensuite envisager une configuration sur partition cryptée...)

Dans l'ajout des comptes utilisateurs, il va falloir maintenant affecter un numéro d'appel (et autoriser les utilisateurs à appeler un autre utilisateur), c'est le fichier « extensions.conf » qui se chargera de répertorier les clients :

```
nano /etc/asterisk/extensions.conf
```

Au bout du fichier, ajouter les lignes suivantes:

```
[travail]
exten => _0XX,1,Dial(SIP/${EXTEN},20)
exten => _0XX,2,Hangup()
```

[travail] indique le conteneur dans lequel les utilisateurs pourrons communiquer entre eux, dans le même ordre qu'un groupe de travail dans lequel les utilisateurs peuvent discuter.

Lors de la création de nos deux utilisateurs nous avons spécifié le contexte work.

**exten =>** : déclare l'extension (on peut aussi simplement dire numéros)

**\_0XX** : Prend les extensions (ou numéros) de 000 à 099 le « \_ » permet d'utiliser des regex

**1** : Ordre de l'extension

**Dial** : application qui va être utilisé

**SIP**: Protocol qui va être utilisé

**\${EXTEN}** : variable de l'extension composé, si on appelle le 001 la variable \${EXTEN} prendra comme valeur 001

**20**: temps d'attente avant de passer à l'étape suivante.

Lorsque l'on compose le numéro 001 (par exemple), le numéro 001 est appelé. Après 20 secondes sans réponses la ligne 2 est exécutée. « exten => \_0XX,2,Hangup() » permet de raccrocher en cas de non réponse.

Plusieurs méthodes sont possibles pour remplir le fichier extension, la méthode ci-dessous est plus compréhensible que celle utilisée dans notre configuration. Mais deviendra très compliquée à gérer lorsque notre serveur gèrera plusieurs dizaines de client. On préférera donc celle proposée précédemment.

```
exten => 001,1,Dial(SIP/001,20)
exten => 002,1,Dial(SIP/002,20)
exten => 003,1,Dial(SIP/003,20)
```

...

```
GNU nano 2.2.6      Fichier : /etc/asterisk/extensions.conf      Modifi  
; For more information on applications, just type "core show applications" at y$
; friendly Asterisk CLI prompt.
;
; "core show application <command>" will show details of how you
; use that particular application in this file, the dial plan.
; "core show functions" will list all dialplan functions
; "core show function <COMMAND>" will show you more information about
; one function. Remember that function names are UPPER CASE.

exten => 111,1,Dial(SIP/client1)
exten => 222,1,Dial(SIP/client2)
exten => 333,1,Dial(SIP/client3)

^G Aide      ^O   crire  ^R Lire fich.^Y Page pr  c.^K Couper  ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher  ^V Page suiv.^U Coller    ^T Orthograp.
```

Terminer l'ajout de client en red  marrant le serveur :

service asterisk restart

Lors d'ajout de nouveaux utilisateurs, il sera possible d'utiliser la commande reload au lieu de restart !

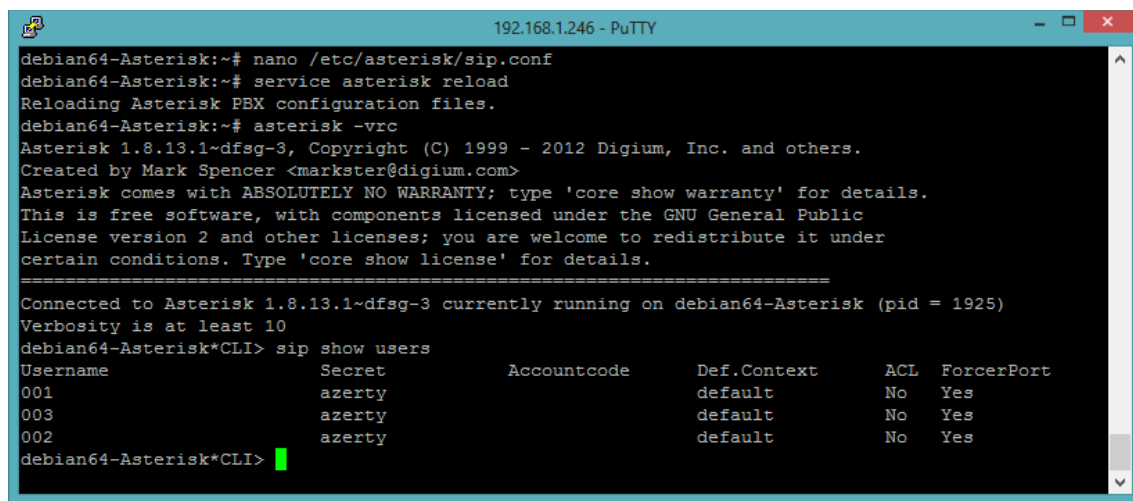
service asterisk reload

Dans la console Asterisk, il sera possible de voir les utilisateurs saisis avec la commande suivante :

sip show users

```
192.168.1.246 - PuTTY
debian64-Asterisk:~# asterisk -vrc
Asterisk 1.8.13.1~dfsg-3, Copyright (C) 1999 - 2012 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.8.13.1~dfsg-3 currently running on debian64-Asterisk (pid = 1925)
Verbosity is at least 10
debian64-Asterisk*CLI> sip show users
Username      Secret      Accountcode  Def.Context  ACL  ForcerPort
001           client1     travail      No           Yes
003           client3     travail      No           Yes
002           client2     travail      No           Yes
debian64-Asterisk*CLI>
```

Suivante le contexte utilis  e ou non....



```
192.168.1.246 - PuTTY
debian64-Asterisk:~# nano /etc/asterisk/sip.conf
debian64-Asterisk:~# service asterisk reload
Reloading Asterisk PBX configuration files.
debian64-Asterisk:~# asterisk -vrc
Asterisk 1.8.13.1~dfsg-3, Copyright (C) 1999 - 2012 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.8.13.1~dfsg-3 currently running on debian64-Asterisk (pid = 1925)
Verbosity is at least 10
debian64-Asterisk*CLI> sip show users
Username      Secret      Accountcode  Def.Context  ACL  ForcerPort
001           azerty      001          default      No   Yes
003           azerty      003          default      No   Yes
002           azerty      002          default      No   Yes
debian64-Asterisk*CLI> █
```

## 6 Installer les clients Linux

La très grande majorité des postes de travail restent sous Windows, cependant, on trouvera de manière courante des stations Linux pour l'utilisation que l'on peut faire d'Asterisk. On peut citer parmi les applications possibles, les laboratoires de langue, les salles de conférence, les hotlines... les postes de travail ont alors une destination spécifique et dans ce domaine Linux est en mesure de proposer des systèmes très modulables et adaptable facilement à toute situation.

C'est pourquoi ce tutoriel présente l'utilisation du client linux. Il n'est pas question ici de voir l'installation et la configuration de postes client.

Ce chapitre montre comment installer et configurer le « client » « Ekiga ». Le reste de la configuration (salle de conférence, traduction, dial...) ne seront qu'une adaptation.

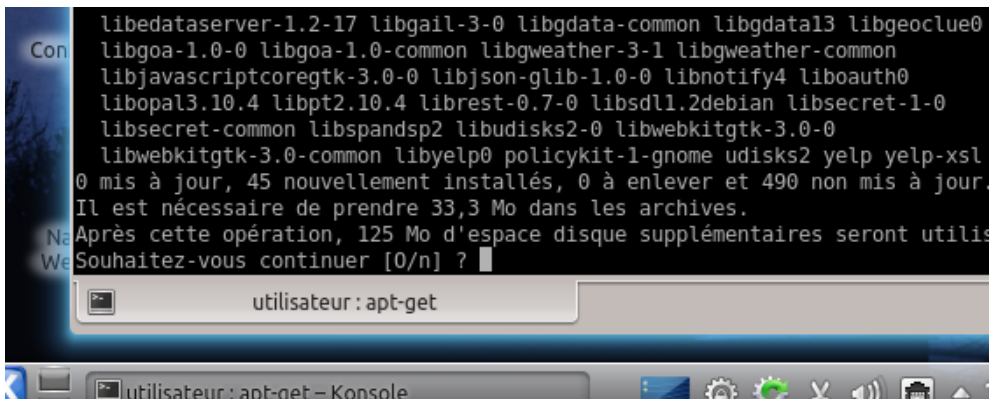
Pour cette action, vous devez disposer d'un poste de travail sous Linux. La plupart des distributions proposent le paquetage Ekiga. L'exemple ci-dessous prend une station de travail sous Ubuntu (station classique avec une connexion réseau).



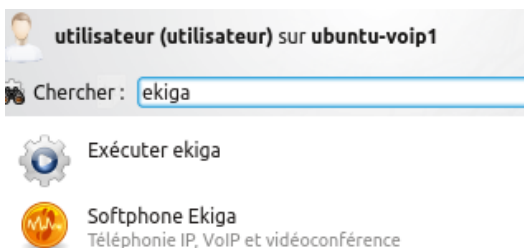
Installer Ekiga (attention la commande est encore compliquée 🤖) :

```
apt-get install ekiga
```

Il y aura un peu plus de 100mo à installer...



Après installation lancer le client :



Lors du premier lancement un assistant d'installation va être exécuté, passer cet assistant (à moins que vous désiriez remplir toutes les rubriques, surtout les différentes inscriptions...)

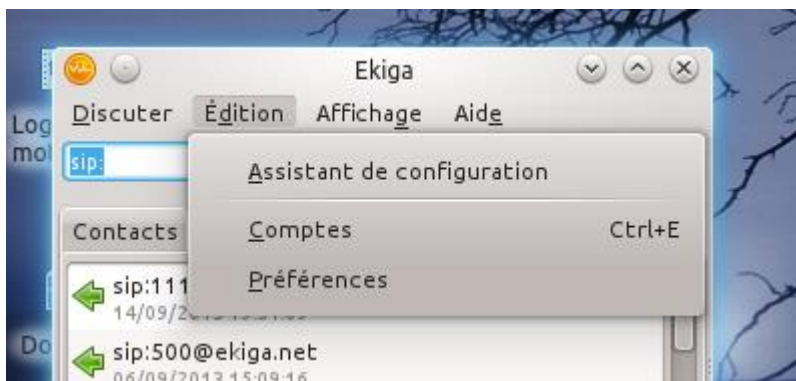




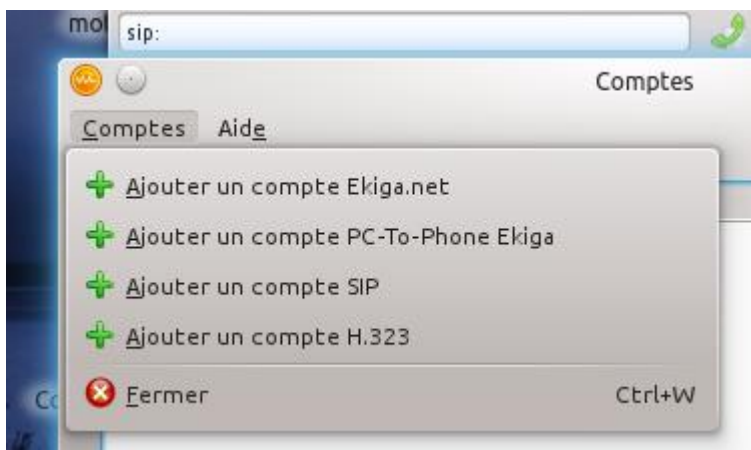
Après lancement de l'interface vous pourrez toujours revenir aux préférences (menu « Edition »... )

Une simple configuration du compte suffira...

Se rendre dans « Edition » puis « Comptes » :



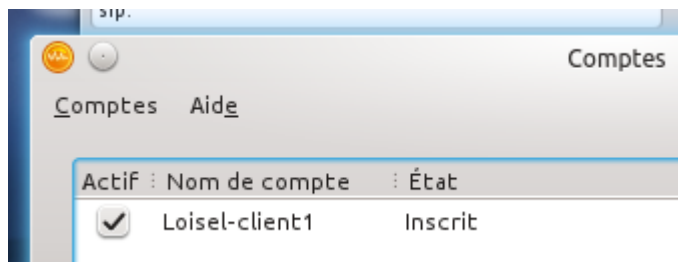
Dans la nouvelle fenêtre aller dans « Comptes » puis « Ajouter un compte SIP »



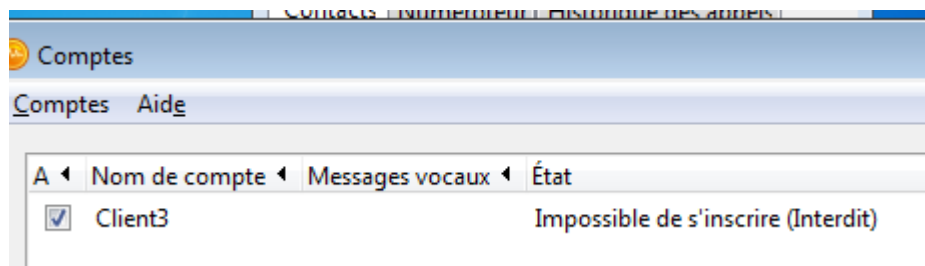
Ajouter alors les paramètres correspondant au serveur (registaire est l'adresse du serveur)...



En validant le compte, celui-ci doit être indiqué comme valide :



Sinon (mais cela risque de ne pas fonctionner 🙄) :



## 7 Installer les clients Windows

Comme indiqué dans le chapitre d'installation du serveur, on pourra lire souvent sur l'Internet qu'il n'existe pas de client, ou que les clients sur les produits Microsoft sont compliqués à installer et configurer.

C'est pour cette raison que le client Linux a été installé en premier.

Dans ce chapitre, vous allez constater qu'effectivement, il est bien compliqué d'installer et configurer le client VOIP sur des solutions Microsoft.

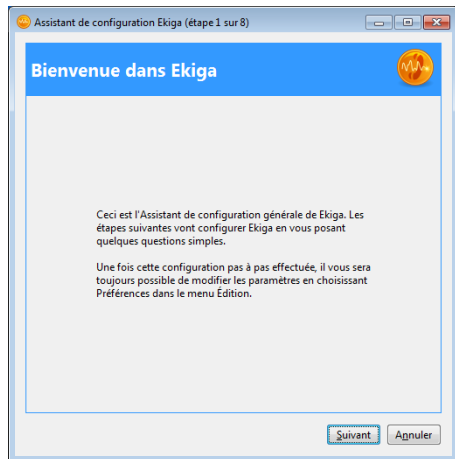
Fort heureusement, pour simplifier la tâche, il existe un client Ekiga disponible pour les produits Microsoft. Il s'agit d'une portabilité vers cette plateforme du logiciel rencontré sous Linux.

Libre et gratuit (libre ne veut pas pour autant dire gratuit !), il sera possible de le télécharger sur les sites classiques de téléchargement. D'une taille d'environ 20Mo il sera très vite rapatrié sur le poste client.

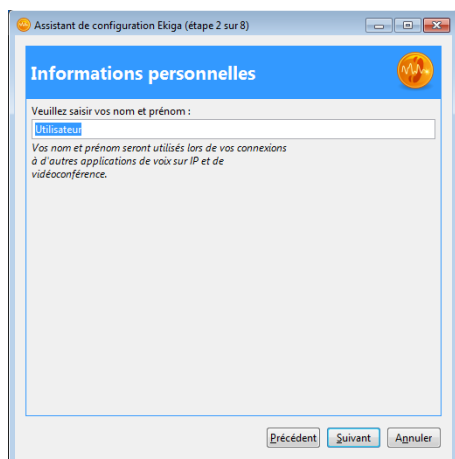
Ne restera qu'à l'installer. Il ne sera pas fait ici de grande démonstrations (certains diront il suffit de faire « suivant » « suivant » « suivant »... et c'est le cas).

En fin d'installation il sera demandé si l'on veut lancer le programme... pourquoi pas !

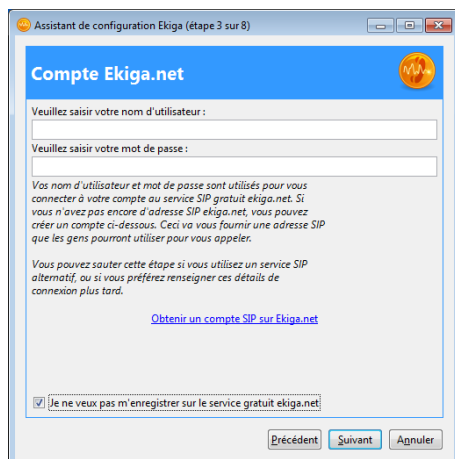
## Configurer Ekiga pour Windows



Il suffira ensuite de suivre les étapes de configuration.



Suivante le choix, s'il s'agit d'une utilisation interne, sur le serveur précédemment installé, il ne sera pas nécessaire de s'inscrire au programme de connexion gratuit Ekiga (mais c'est gratuit...) ce service permettra de se connecter à distance vers d'autres utilisateur inscrit eux également sur le site.



Il en est de même pour le compte Ekiga PC-To-Phone (les conditions sont quelque peu différentes...)

Assistant de configuration Ekiga (étape 4 sur 8)

### Compte PC-To-Phone Ekiga

Veuillez saisir votre identifiant de compte :

Veuillez saisir votre code PIN :

Vous pouvez effectuer des appels vers des téléphones fixes et mobiles dans le monde entier en utilisant Ekiga.

Pour activer ceci, vous devez :

- créer tout d'abord un compte en utilisant l'URL ci dessous.
- saisir ensuite votre identifiant de compte et le code PIN.

Le service ne fonctionnera que si vous utilisez l'URL de cette fenêtre pour créer le compte.

[Obtenir un compte Ekiga PC-To-Phone](#)

[Recharger le compte](#)

[Afficher l'historique de la balance](#)

[Consulter l'historique des appels](#)

☒ Je ne veux pas m'inscrire au service PC-To-Phone Ekiga

Fin Précédent Suivant Annuler

Les réglages suivants concerneront la connexion et l'utilisation courante du logiciel (sonnerie, périphériques d'emploi...)

Assistant de configuration Ekiga (étape 5 sur 8)

### Type de connexion

Veuillez choisir le type de votre connexion :

Conserver les réglages actuels

Le type de connexion permettra de déterminer les meilleurs paramètres que Ekiga utilisera durant les appels. Vous pouvez changer ces paramètres plus tard de façon individuelle dans la fenêtre des préférences.

Fin Précédent Suivant Annuler

Assistant de configuration Ekiga (étape 6 sur 8)

### Périphériques audio

Veuillez choisir le périphérique de sonnerie :

Default (PTLIB/WindowsMultimedia)

Le périphérique audio de sonnerie est le périphérique qui sera utilisé pour jouer le son de la sonnerie lors des appels entrants.

Veuillez choisir le périphérique de sortie audio :

Default (PTLIB/WindowsMultimedia)

Le périphérique de sortie audio est le périphérique qui sera utilisé pour produire le son au cours des appels.

Veuillez choisir le périphérique d'entrée audio :

Default (PTLIB/WindowsMultimedia)

Le périphérique d'entrée audio est le périphérique qui sera utilisé pour enregistrer votre voix au cours des appels.

Fin Précédent Suivant Annuler

Assistant de configuration Ekiga (étape 7 sur 8)

### Périphérique d'entrée vidéo

Veuillez choisir le périphérique d'entrée vidéo :

Moving Logo (Moving Logo/Moving Lo...)

Le périphérique d'entrée vidéo est le périphérique qui sera utilisé pour enregistrer la vidéo au cours des appels.

Précédent Suivant Annuler

Appliquer la configuration en fin d'assistant.

Assistant de configuration Ekiga (étape 8 sur 8)

### Configuration terminée

Vous avez maintenant terminé la configuration d'Ekiga. Tous les réglages peuvent être modifiés dans les préférences d'Ekiga. Amusez vous bien !

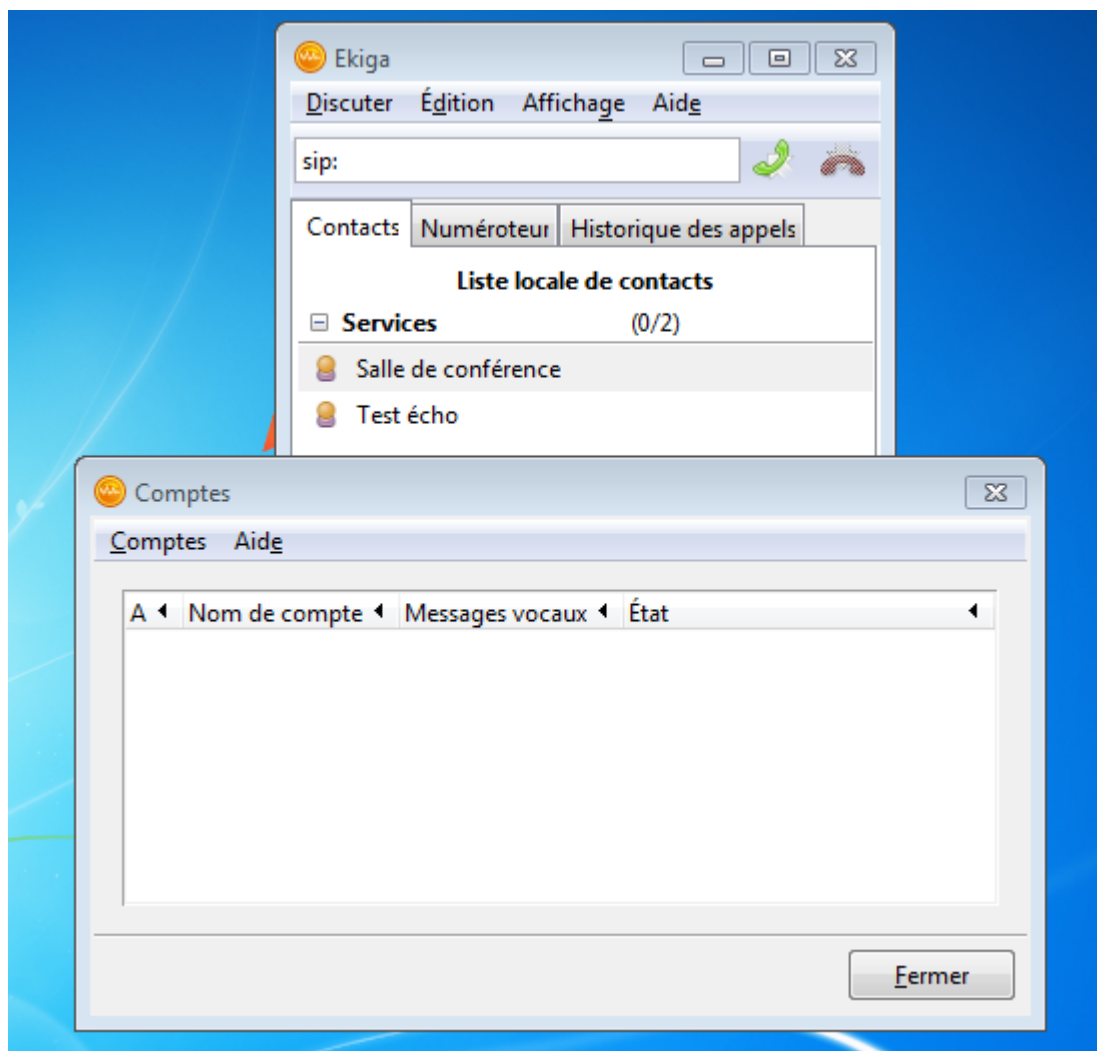
Résumé de la configuration :

Option	Value
Nom complet	Utilisateur
Type de connexion	Conserver les réglages actuels
Périphérique audio de sonnerie	Default (PTLIB/WindowsMultimedia)
Périphérique de sortie audio	Default (PTLIB/WindowsMultimedia)
Périphérique d'entrée audio	Default (PTLIB/WindowsMultimedia)
Périphérique d'entrée vidéo	Moving Logo (Moving Logo/Moving Logo)
URI SIP	None
PC-To-Phone Ekiga	None

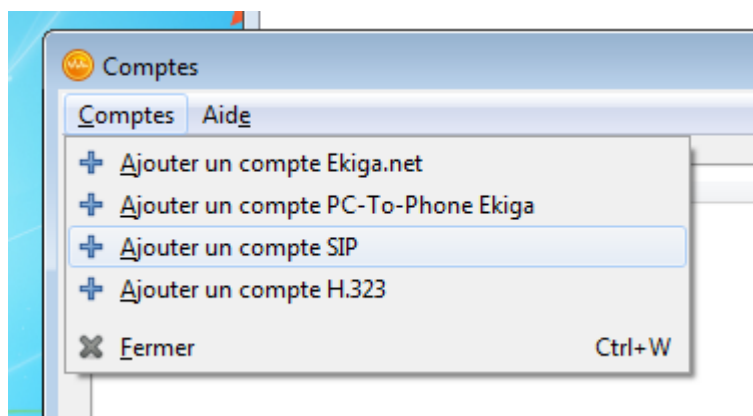
Précédent Appliquer Annuler

## Configurer le compte

Après lancement du logiciel, on pourra remarquer une grande similitude avec la version Linux et, tout comme pour Linux, il suffira de se rendre dans « Edition », « Compte » pour y ajouter et configurer un compte.



Il faudra alors ajouter un compte « Sip »



Chacun aura remarqué cette difficulté tant indiquée sur l'Internet quant à l'installation et la configuration du client. Preuve encore une fois que ce support est à manipuler et à suivre avec précaution et recul.

## 8 Tester les clients

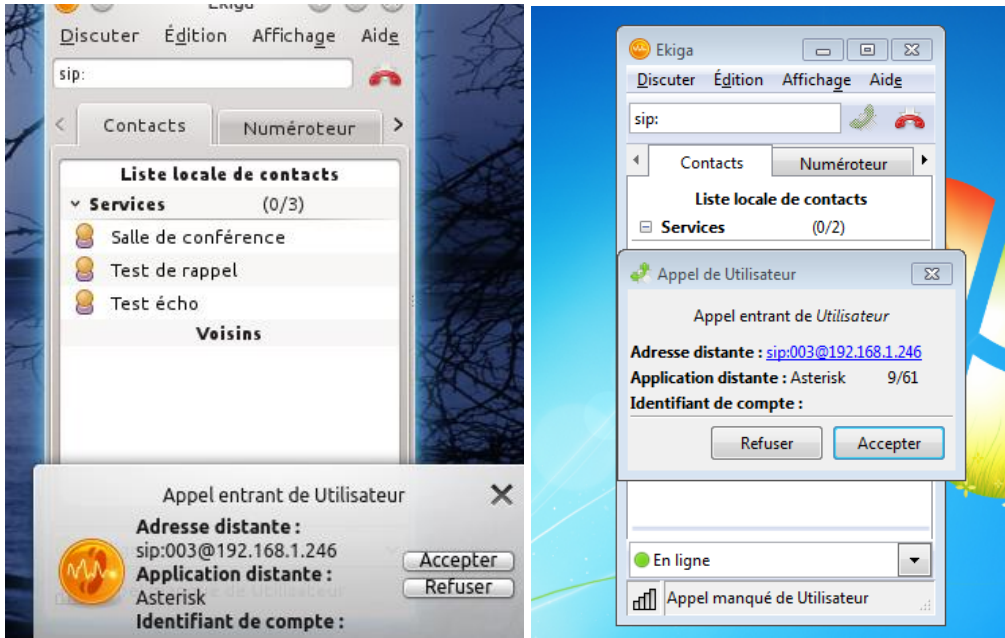
Pour tester le client, il conviendra d'avoir au minimum 2 clients de configurés...

Dès lors que la configuration des compte est faite, que le compte vous indique « Inscrit », le test est encore une opération bien compliquée : dans la zone de texte « sip », taper le numéro du correspondant :

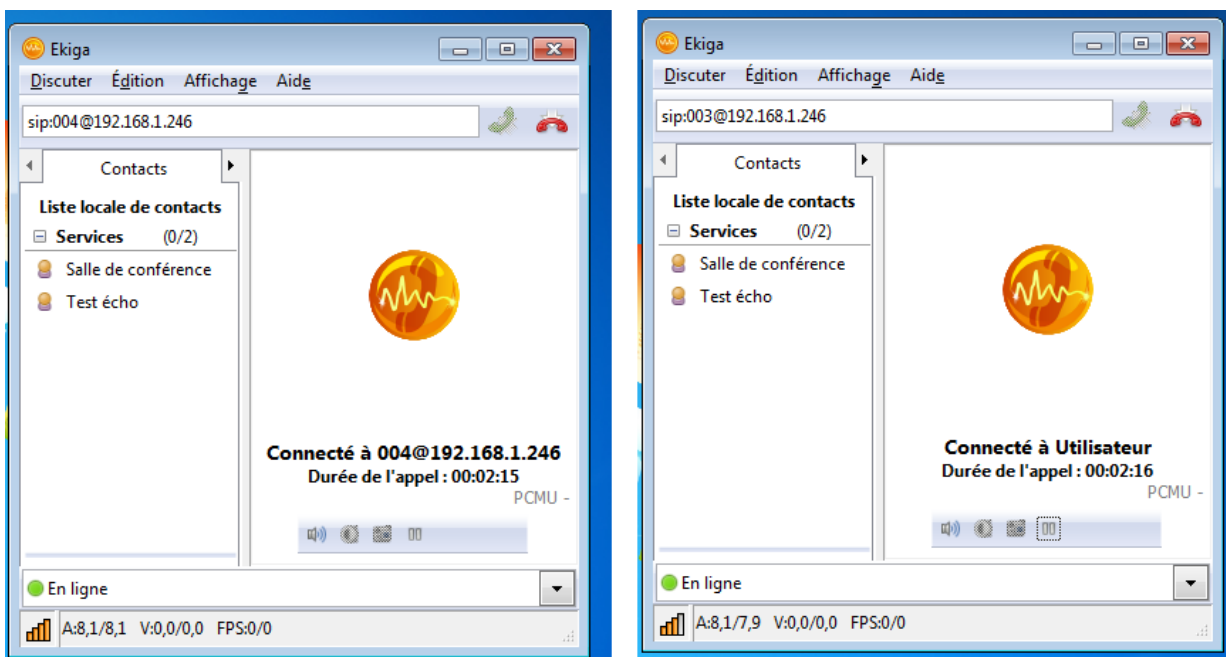


Puis décrocher.

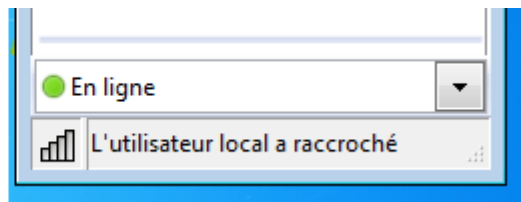
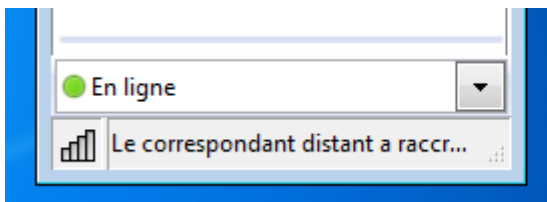
A la réception de l'appel, le client distant pourra décrocher...



En cours de connexion, les clients afficheront l'état...



...



## 9 Créer un script pour la création de client

A cette étape il va s'agit de créer un nouvel utilisateur (ou client) depuis un script en passant simplement quelques paramètres : le numéro sip / Le nom d'utilisateur / Le mot de passe. A ce stade de la configuration du serveur, il s'agit de créer un script simple qui facilitera la tâche de l'administrateur. Il n'est pas question dans ce tutoriel d'expliquer toutes les finesses de la création d'un script. Les commandes utilisées étant très simples, on suppose que le lecteur (vous) avez quelques connaissances dans ce domaine. De plus, on notera que le script est directement utilisable dans l'état ou il est présenté dans ce document.

Le script sera directement intégré dans « /usr/bin » ce qui permettra un emploi direct du fichier.

Le script sera utilisé en administrateur suivant la stratégie appliquée au fichier « /etc/asterisk/sip.conf »

Créer le fichier « créer\_sip » :

```
touch /usr/bin/creer_sip
```

Rendre le fichier exécutable avec commande :

```
chmod +x /usr/bin/creer_sip
```

Ouvrir et éditer le fichier « créer\_sip » :

```
nano /usr/bin/creer_sip
```

Pour obtenir le script qui intégrera un nouveau client dans le fichier « sip.conf », ajouter au fichier le code suivant :

```
#!/bin/sh

# Script de creation d un utilisateur sip
# #####

# Retourne l'usage du script au besoin
usage()
{
    echo ""
    echo "#### Creation utilisateur sip par LoiselJP ####"
    echo "#####"
    echo "Utilisation du script: "
    echo "$0 numero_sip nom_utilisateur mot_de_passe"
```

```

    echo "Le numero doit repondre a la plage du serveur"

    echo "#####"

    echo ""

}

# Teste l'entree passee en parametre => si pas d'entree retourne l'usage
if [ -z $3 ]; then
    usage
    exit 0
fi

# On ajoute le code dans le fichier /etc/asterisk/sip.conf
echo "" >> /etc/asterisk/sip.conf
echo "[$1](template)" >> /etc/asterisk/sip.conf
echo "username=$2" >> /etc/asterisk/sip.conf
echo "secret=$3" >> /etc/asterisk/sip.conf

# Recharger le service
service asterisk reload

```

Vérifier le fonctionnement du script : en tapant le nom du script sans utilisateur, puis avec un utilisateur.

```

192.168.1.246 - PuTTY
debian64-Asterisk:~# nano /etc/asterisk/sip.conf
debian64-Asterisk:~# nano /etc/asterisk/sip.conf
debian64-Asterisk:~# touch /usr/bin/creer_sip
debian64-Asterisk:~# chmod +x /usr/bin/creer_sip
debian64-Asterisk:~# nano /usr/bin/creer_sip
debian64-Asterisk:~# creer_sip

#### Creation utilisateur sip par LoiselJP ####
#####
Utilisation du script:
/usr/bin/creer_sip numero_sip nom_utilisateur mot_de_passe
Le numero doit repondre a la plage du serveur
#####
debian64-Asterisk:~# creer_sip 008 client8 client8

```

Vérifier la création de cet utilisateur :



```
192.168.1.246 - PuTTY
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.8.13.1~dfsg-3 currently running on debian64-Asterisk (pid = 1925)
Verbosity is at least 10
debian64-Asterisk*CLI> sip show users
Username      Secret      Accountcode  Def.Context  ACL  ForcerPort
005           client5     00000000     travail      No   Yes
004           client4     00000000     travail      No   Yes
007           client7     00000000     travail      No   Yes
006           client6     00000000     travail      No   Yes
001           client1     00000000     travail      No   Yes
003           client3     00000000     travail      No   Yes
002           client2     00000000     travail      No   Yes
008           client8     00000000     travail      No   Yes
debian64-Asterisk*CLI>
```

Note : Le script ne teste pas l'existence d'un utilisateur.

Pour les plus aisés, il sera possible de tester la présence de la ligne « [\$1](template) » dans le fichier sip.conf (en compte la nombre de ligne contenant cette donnée), puis en renvoyant vers une fonction indiquant cette existence. Les scripts ne sont pas à sujet de ce tutoriel...

## 10 Configuration et utilisation avancée d'Asterisk

Le serveur Asterisk est désormais configuré pour un fonctionnement standard. Asterisk propose toutes les subtilités qui peuvent être demandées à un serveur moderne.

Il serait bien difficile de faire un tour complet des possibilités offertes par ce serveur, le fichier de configuration en donne une vague idée...

Pour n'y prendre qu'un petit exemple, les options de configuration disponibles en disent long à elles seules :

1. context
2. callingpres
3. permit
4. deny
5. secret
6. md5secret
7. remotesecret
8. transport
9. dtmfmode
10. directmedia
11. nat
12. callgroup
13. pickupgroup
14. language
15. allow
16. disallow
17. insecure
18. trustpid
19. progressinband
20. promiscreadir
21. useclientcode

- 22. accountcode
- 23. setvar
- 24. callerid
- 25. amaflags
- 26. callcounter
- 27. busylevel
- 28. allowoverlap
- 29. allowsubscribe
- 30. allowtransfer
- 31. ignoresdpversion
- 32. subscribecontext
- 33. template
- 34. videosupport
- 35. maxcallbitrate
- 36. rfc2833compensate
- 37. mailbox
- 38. session-timers
- 39. session-expires
- 40. session-minse
- 41. session-refresher
- 42. t38pt\_usertpsource
- 43. regexten
- 44. fromdomain
- 45. fromuser
- 46. host
- 47. port
- 48. qualify
- 49. defaultip
- 50. defaultuser
- 51. rtptimeout
- 52. rtpholdtimeout
- 53. sendrpid
- 54. outboundproxy
- 55. rfc2833compensate
- 56. callbackextension
- 57. registertrying
- 58. timert1
- 59. timerb
- 60. qualifyfreq
- 61. t38pt\_usertpsource
- 62. contactpermit
- 63. contactdeny
- 64. directmediapermit
- 65. directmediadeny
- 66. unsolicited\_mailbox
- 67. use\_q850\_reason
- 68. maxforwards
- 69. encryption

Pour l'exemple de ce chapitre, la messagerie va être configurée.

Par défaut, les voix proposée par Asterisk sont en anglais.

Les voix françaises vont donc être installées :

```
apt-get install asterisk-prompt-fr-armelle
```

(le paquet « asterisk-prompt-fr-proformatique » était également disponible)

Télécharger et installer le paquet « asterisk-sounds-extra » :

```
cd /tmp  
wget http://archive.debian.org/debian/pool/main/a/asterisk-sounds-extra/asterisk-sounds-extra_1.4.7.orig.tar.gz  
tar -xvf asterisk-sounds-extra_1.4.7.orig.tar.gz
```

Puis déplacer les fichiers complémentaires dans « /usr/share/asterisk/sounds/ »

```
mv *.gsm /usr/share/asterisk/sounds/
```

Le chapitre suivant (Configuration de la messagerie) est un exemple de configuration possible du serveur. A l'écoute du message d'accueil de cette messagerie il est possible d'imaginer l'étendue du système, il sera alors facile de comprendre les possibilités offertes par le serveur.

Les possibilités offertes par le serveur étant très grandes (passerelle vocale, centre d'appel, centre de conférence, serveur d'appel, serveur vocal, serveur de mail vocal...), pour une utilisation spécifique, il sera possible de se rendre sur le site du système : <http://www.asterisk.org>

## Configuration de la messagerie

Dans notre exemple, nous allons créer une boîte vocale pour «001» («client1 ») .

Modifier le fichier de configuration de mail « voicemail.conf »:

```
nano /etc/asterisk/voicemail.conf
```

On pourra y voir les paramètres possibles :

Nomdre de message maxi : Maxmsg  
Longueur maxi d'un message : Maxsecs  
Longueur mini d'un message : Minsecs  
...

En fin de fichier ajouter les numéros d'appel de messagerie:

```
[default]
```

091 => 091, client1, mail@client1.com

092 => 092, client2, mail@client2.com

093 => 093, client3, mail@client3.com

094 => 094, client4, mail@client4.com

...

Le premier numéro est le numéro de boîte vocale, Le second chiffre est le mot de passe pour le boîte vocale, viennent ensuite le nom du propriétaire puis son adresse mail.

Éditer le fichier sip.conf pour y ajouter l'adresse de messagerie correspondante:

nano /etc/asterisk/sip.conf

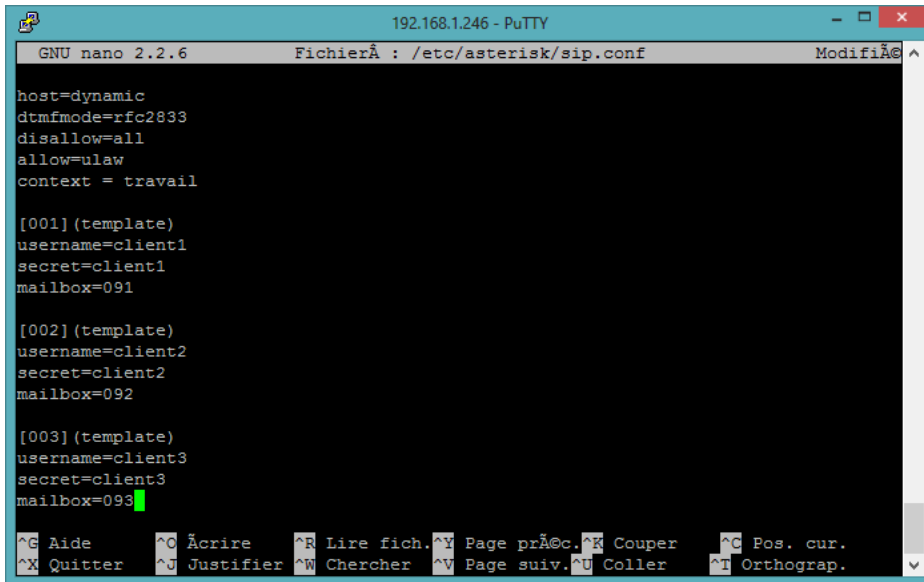
Au client 1 ajouter la ligne :

mailbox=091

Au client2 ajouter la ligne:

mailbox=092

...



```
GNU nano 2.2.6 Fichier : /etc/asterisk/sip.conf
host=dynamic
dtmfmode=rfc2833
disallow=all
allow=ulaw
context = travail

[001] (template)
username=client1
secret=client1
mailbox=091

[002] (template)
username=client2
secret=client2
mailbox=092

[003] (template)
username=client3
secret=client3
mailbox=093
```

Terminer la configuration en ajoutant dans le fichier extensions.conf le(s) numéro(s) d'appel de messagerie :

nano /etc/asterisk/extensions.conf

Pour configurer un numéro par boîte :

exten => 091,1,VoiceMailMain(091)

exten => 092,1,VoiceMailMain(092)

Ou configurer un numéro commun (090) :

```
exten => 090,1,VoiceMailMain
```


Il est possible de mettre 1 numéro commun d'appel de messagerie et un numéro par boîte, c'est ainsi que pour l'exemple on peut placer le numéro 090 commun et de 091 à 098 le numéro de boîte correspondant au client 1 à 8...

Comme indiqué lors des chapitres précédents, il est nécessaire de recharger la configuration pour qu'elle soit prise en charge :

```
service asterisk reload
```

Suivant la méthode choisie (1 numéro commun ou numéro de boîte mail, voir les 2 méthodes...), appeler la boîte de messagerie.

Après avoir tapé votre « login/mot de passe » ou « mot de passe » (cas d'un numéro par boîte), vous pourrez alors découvrir une boîte de messagerie très complète dans laquelle vous pourrez créer un nouveau message, changer l'accueil, écouter, effacer, gérer vos messages...

Pour le besoin du tutoriel, il était difficile de placer une impression d'écran pour illustrer la fonction répondeur, le message a été enregistré.... Tendez bien l'oreille pour l'écouter  !

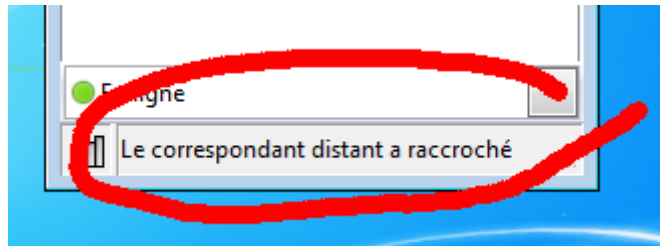
## 11 Conclusion

Ce document, relativement assez court exprime bien cette difficulté à créer un serveur VOIP.

Bien que la gestion de salle de conférence, la configuration d'un répondeur avec la gestion d'un central puisse amener quelques difficultés supplémentaires, comme expliqué dans l'objectif de ce document, il s'agissait de démystifier cette idée sur les serveurs « voip ».

Vous pouvez classer aujourd'hui les serveurs VOIP au même niveau que tout serveur : il ne s'agit de rien d'autre qu'un service (programme) destiné à transférer des données (le flux audio n'est rien d'autre qu'une donnée) d'un client vers un autre client.

Ce serveur devra être configuré suivant des paramètres définis tout comme le serait un serveur Web ou un serveur de fichier avec ses difficultés suivant l'objectif visé.



# # init 0

# ©Propriété

## Installer un serveur VOIP Asterisk et ses clients.

Jean Paul Loisel  
56 Rue Philippe de Girard  
59160 Lomme

✉loiseljp@club-internet.fr

☎ 07 70 32 35 67

## Licence

Ce document est distribué en "Public Documentation License".

The contents of this Documentation are subject to the Public Documentation License. You may only use this Documentation if you comply with the terms of this License.

A copy of the License is available at this mail [loiseljp@club-internet.fr](mailto:loiseljp@club-internet.fr).

The Original Documentation is "**Installer un serveur VOIP Asterisk et ses clients**".

The Initial Writer of the Original Documentation is Jean Paul LOISEL © 2013. All Rights Reserved.

Contributor(s): \_\_\_\_\_.

Portions created by \_\_\_\_\_ are Copyright © \_\_\_\_\_ [Insert year(s)]. All Rights Reserved.

(Contributor contact(s): \_\_\_\_\_ [Insert hyperlink/alias]).

📖 The text of this chapter may differ slightly from the text of the notices in the files of the Original Documentation.

You should use the text of this chapter rather than the text found in the Original Documentation for Your Modifications.