

Dossier documentaire

Document 1 – Installation et prise en main d'Asterisk

D1.1 – Installation d'Asterisk

L'installation peut se faire à partir des sources ou à l'aide du gestionnaire de dépôts. L'avantage d'une installation à partir des sources est qu'elle permet d'obtenir la dernière version du logiciel. Une installation à partir des dépôts reste plus rapide et plus facile à mettre en œuvre.

Pré-requis : La machine sur laquelle sera installé Asterisk doit avoir un accès Internet (mettre temporairement la machine en adressage dynamique, en bridge ou en nat s'il s'agit d'une machine virtuelle).

```
#apt-get install asterisk
```

NB : Une fois les paquets installés, revenir à la configuration prévue sur le schéma, en respectant le plan d'adressage fixe.

Les fichiers de configurations sont localisés dans le répertoire `/etc/asterisk`. Les principaux fichiers que nous serons amenés à configurer sont les suivants :

FICHIERS	DESCRIPTIONS
<code>sip.conf</code>	Configuration des canaux SIP.
<code>users.conf</code>	Configuration des utilisateurs.
<code>voicemail.conf</code>	Configuration de la messagerie vocale.
<code>extensions.conf</code>	Configuration du DialPlan.

L'arrêt et le redémarrage d'Asterisk peuvent se faire à l'aide de la commande `service`.

```
#service asterisk restart
```

Les clients SIP peuvent utiliser les ports TCP et UDP 5060 pour se connecter au serveur. Le port 5061 est utilisé dans le cas d'une utilisation de TLS. La commande `ps` permet de vérifier que notre serveur est prêt. Il est aussi possible d'utiliser la commande `service asterisk status`.

```
#ps -ef | grep asterisk
```

```
root@asterisk:~# ps -ef | grep asterisk
avahi      436      1  0 10:45 ?        00:00:00 avahi-daemon: running [asterisk.
local]
asterisk   2562      1  6 10:54 ?        00:00:00 /usr/sbin/asterisk -p -U asteris
```

Il est intéressant également de vérifier les ports ouverts par Asterisk, dont le port SIP (5060). Utiliser la commande `netstat` pour ce faire.

D1.2 – Interface en ligne de commandes d'Asterisk

La console CLI d'Asterisk offre une large variété de commandes permettant la configuration et la surveillance du serveur. Les principales commandes que nous utiliserons sont les suivantes :

COMMANDES	DESCRIPTIONS
<code>asterisk -rvvvv</code>	Lancement de la console. Le niveau de verbosité est défini par le nombre de lettres v indiqué. - Un seul v affichera les messages d'erreurs ; - Deux v pour les warnings ; - Trois v pour les messages d'informations ; - Quatre v pour tous les messages.

reload	Chargement de la configuration suite à une mise à jour de la configuration du serveur.
sip show users	Affichage de la liste des utilisateurs créés sur le serveur.
sip show peers	Affichage de la liste des téléphones enregistrés auprès du serveur.
dialplan show	Affichage du plan de numérotation.
iax show stats	Affichage des statistiques des liaisons entre serveurs Asterisk.
exit	Quitte le mode CLI.

D1.3 – Protocoles utilisés

Deux types de protocoles sont à distinguer : les protocoles de signalisation et les protocoles de transport de la voix.

Les flux de signalisation VOIP permettent de gérer les appels, c'est à dire tout ce qui concerne l'enregistrement des téléphones auprès du serveur IPBX. Le protocole (standard ouvert) de gestion des sessions que nous utiliserons est SIP (Session Initiation Protocol). Il écoute sur le port UDP 5060.

Le détail des échanges entre un poste et le serveur pourra être observé lors de la partie concernant les captures de trames avec le logiciel Wireshark. Le protocole IAX (Inter eXchange Asterisk) sera utilisé, dans ce côté labo, pour effectuer la liaison entre deux serveurs Asterisk.

Les principaux codes de statut du protocole SIP ressemblent à ceux du protocole HTTP.

CODES	DESCRIPTIONS
1xx	Information (sonnerie, transfert...).
2xx	Succès (200 = OK)
3xx	Redirection.
4xx	Erreur Client
5xx	Erreur serveur

Les requêtes SIP échangées permettent au client de demander une nouvelle session et de s'enregistrer auprès du serveur IPBX. En voici un bref aperçu :

REQUÊTES	DESCRIPTIONS
INVITE	Le client demande une nouvelle session.
ACK	Accusé de réception.
CANCEL	Annulation d'un INVITE en cours.
BYE	Termine une session.
REGISTER	Permet de s'enregistrer auprès de l'IPBX.

Le système d'adressage SIP-URI ressemble à une adresse mail et permet d'identifier chaque ressource. Il se présente sous le format suivant : *sip:username@hôte:port*. Le port par défaut étant 5060, il ne sera pas toujours utile de le spécifier.

Quant au transport de la voix, il est assuré par les protocole RTP (Real Time Transfer Protocol) et RTPC (Real Time Transfer Control Protocol). Le protocole RTP fournit un moyen de transmettre des données soumises à des contraintes de temps (flux média, audio, vidéo). Des paquets de contrôle peuvent être périodiquement envoyés par RTPC afin de véhiculer des informations sur les participants d'une session ou dans le cadre de la qualité de service.

Document 2 – Création des utilisateurs

D2.1 – Configuration générale des utilisateurs

La gestion des utilisateurs se fait à l'aide du fichier *users.conf* et peut être centralisée en réalisant une association avec un serveur LDAP. Dans ce côté labo, nous partirons d'un fichier vide que nous alimenterons au fur et à mesure en prenant soin de créer une copie du fichier d'origine riche en exemples de configurations. Cette démarche sera répétée pour la plupart des fichiers que nous manipulerons. Le répertoire contenant les fichiers de configuration est localisé dans */etc/asterisk*.

```
cp users.conf users.conf.sauve ou mv users.conf users.conf.sauve
```

```
echo " " > users.conf ou touch users.conf
```

Des commentaires peuvent être ajoutés dans le fichier de configuration à l'aide du symbole « ; ».

Le fichier *users.conf* est découpé en plusieurs blocs. Le premier concerne la configuration générale qui sera appliquée à l'ensemble des utilisateurs.

```
[general] ;s'applique donc à tous les utilisateurs
hasvoicemail = yes ;l'utilisateur possède une boîte vocale.
hassip = yes ;l'utilisateur possède un compte sip.
```

Beaucoup d'autres options sont possibles comme le filtrage des utilisateurs en fonction de leur adresse IP, la possibilité de transférer des appels ou pour gérer le cas des utilisateurs se trouvant derrière un NAT.

D2.2 – Utilisation d'un modèle de paramètres communs

La configuration d'un *template* (modèle de paramètres) permet de définir un ensemble de paramètres qui seront communs à plusieurs utilisateurs. Le but étant de factoriser ces paramètres afin d'éviter de multiples saisies lors de la création des comptes. Le bloc associé portera le nom du template.

```
[template](!) ;notre template s'appelle template. Le ! Indique qu'il s'agit d'un template.
type = friend ;type d'objet SIP.
host = dynamic ;l'utilisateur n'est pas associé à une IP fixe.
dtmfmode = rfc2833 ;mode DTMF.
disallow = all ;interdit tous les codecs.
allow = ulaw ;autorise le codec ulaw.
allow = alaw ;autorise le codec alaw.
```

Le type d'objet SIP peut prendre 3 valeurs : *peer*, *user* et *friend*.

Le type **peer** correspond à un objet auquel nous pouvons envoyer des objets (cas des *trunk SIP*).

Le type **user** correspond à un objet qui ne peut qu'appeler.

Le type **friend** est à la fois *peer* et *user*. Il peut envoyer et recevoir des appels (cas de nos utilisateurs).

Il convient aussi de donner quelques précisions sur les **codecs** (compresseur/décompresseur). Il s'agit de logiciels qui compressent et décompressent les données. On peut en distinguer trois types : audio, image et vidéo. Les codecs les plus utilisés sont *alaw/ulaw(g711)*, *gsm*, *g726* et *g729*.

Le choix d'un codec s'effectue en fonction des ressources processeur, de la bande passante utilisée et de la qualité de la communication.

D2.3 – Création des blocs utilisateurs

Les utilisateurs sont créés dans des blocs et peuvent faire référence au template de la section précédente.

```
[1101](template) ;1101 est le numéro de téléphone. On précise ensuite le template.
fullname = Utilisateur1 ;remplacer utilisateur1 par votre prénom
username = u1101
```

```
secret = password ;le mot de passe est très simple ici et non sécurisé.  
mailbox = 1101 ;référence vers la boîte vocale (fichiers voicemail.conf).  
context = finance ;l'utilisateur appartient au contexte finance.
```

Il convient de redémarrer Asterisk ou de recharger sa configuration pour prendre en compte les modifications effectuées.

```
asterisk*CLI> reload
```

La configuration n'est pas terminée, des **ERROR** et **WARNINGS** apparaissent.



Ce côté labo ne met pas en œuvre la liaison avec un serveur LDAP.

Par précaution, il peut être utile de vérifier également que les utilisateurs sont bien listés :

```
asterisk*CLI> sip show users
```

Document 3 - Création des boîtes vocales

D3.1 – Configuration générale des boîtes vocales

Les boîtes vocales permettent de laisser un message vocal aux utilisateurs. Le premier bloc comprend des paramètres qui s'appliqueront à l'ensemble des boîtes vocales. Le fichier concerné est *voicemail.conf*. Comme pour *users.conf*, nous partons d'un fichier vierge.

```
[general]  
maxmsg = 100 ;nombre maximum de messages de la boîte vocale.  
maxsecs = 0 ;durée maximum d'un message. Le 0 indique l'absence de limite.  
minsecs = 0 ;durée minimum d'un message.  
maxlogins = 3 ;nombre maximum d'erreur de login.  
review = no ;permet à l'appelant de réécouter son message avant de le laisser sur la boîte  
vocale.  
saycid = no ;dicte le numéro de l'appelant avant l'écoute du message.
```

D3.2 – Création des blocs de boîtes vocales

La suite du fichier permet de créer les boîtes vocales en fonction du contexte de l'utilisateur (voir la directive *context* dans le fichier *users.conf*).

```
[finance]  
1101 => 1234, Utilisateur1  
1102 => 1234, Utilisateur2  
  
[compta]  
1201 => , Utilisateur5
```

Dans cet exemple, les utilisateurs devront saisir le mot de passe 1234 pour accéder à leur boîte vocale. Les autres informations données doivent être cohérentes avec les directives *fullname* et *mailbox* du fichier *users.conf*. L'utilisateur *Utilisateur5* n'a pas de mot de passe affecté à sa boîte vocale.

À ce stade nos utilisateurs ne peuvent pas encore passer des appels. Il reste encore à définir le DialPlan.

Document 4 – Création du DialPlan

D4.1 – Présentation du DialPlan

Cette présentation est à lire avant d'effectuer la configuration du sous-chapitre suivant.

Le plan d'appels (DialPlan) permet le routage des appels à travers le serveur. Le fichier concerné est *extensions.conf*. Il s'agit de déterminer le comportement du serveur en cas d'appels reçus et émis. De même en cas de consultation de la boîte vocale. Plusieurs notions doivent être présentées afin de comprendre l'intégralité du plan d'appel : les extensions, les contextes, les numéros, les applications ainsi que les variables.

Voici un premier exemple pour illustrer ces notions :

N° DE LIGNE	EXTRAIT DU FICHIER EXTENSIONS.CONF
1	<i>[finance]</i> ;Attention, ici XX est un caractère générique à ne pas remplacer, mais
2	; 11 est à adapter à votre équipement éventuellement.
3	
4	;plan de numérotation du contexte finance
5	<i>exten => _11XX,1,DIAL(SIP/\$_{EXTEN},20)</i>
6	<i>exten => _11XX,2,Voicemail(\$_{EXTEN}@finance)</i>
7	
8	;consultation des boîtes vocales du contexte finance.
9	<i>exten => 1199,1,Answer()</i> <i>exten => 1199,2, VoiceMailMain(\$_{CALLERID(num)}@finance)</i>

La ligne n°1 correspond au nom du contexte tel que défini dans le fichier *users.conf*.

A partir de la ligne n°4, le mot clé *exten* indique le début d'une extension, c'est à dire un ensemble d'actions correspondant à un appel émis ou reçu.

Le format général d'une extension est :

```
exten => numéro de téléphone, priorité, application()
```

La priorité définit l'ordre des actions de l'extension car il y a généralement plusieurs actions définies.

Examinons en détail la ligne n°5 afin de comprendre le contenu d'une action.

→Les numéros

Il est possible de spécifier un numéro précis ou d'utiliser un *pattern* afin de prendre en compte plusieurs numéros. Un *pattern* doit commencer par le symbole « *_* ». Les caractères génériques utilisables sont les suivants :

CARACTÈRES	SIGNIFICATIONS
X	Correspond à n'importe quel chiffre entre 0 et 9.
W	Correspond à n'importe quel chiffre entre 1 et 9,.
N	Correspond à n'importe quel chiffre entre 2 et 9.
.	Correspond à un ou plusieurs chiffres.
!	Correspond à zéro ou plusieurs chiffres.

L'utilisation des crochets permet de spécifier des intervalles ou des séquences de chiffres.

[1-6] correspond aux chiffres de 1 à 6.

[1234] correspond aux chiffres 1,2,3 ou 4.

L'extension de la ligne n°5 comprend donc tous les numéros de 1100 à 1199.

→Les applications et les variables

Les applications définissent les actions que le serveur doit réaliser. Elles peuvent prendre des variables en paramètres. Voici celles que nous utiliserons dans notre côté labo.

APPLICATIONS	SIGNIFICATIONS
Answer()	Permet de décrocher l'appel.
HangUp()	Permet de raccrocher l'appel.
Dial(type/id,timeout)	Permet de composer un numéro. Le type peut valoir <i>SIP</i> et le <i>timeout</i> correspond au nombre de secondes à écouler avant de passer à l'action suivante si l'utilisateur ne décroche pas.
VoiceMail(user@contexte)	Permet de joindre la messagerie de l'utilisateur indiqué.
VoiceMailMain(user@contexte)	Permet de consulter la messagerie de l'utilisateur indiqué.
Goto(contexte,extension, priorité)	Permet de se rendre à un contexte associé à une extension et à une priorité (voir trunk IAX).

Concernant la ligne n°5, l'application Voicemail() est associée à l'extension du numéro appelé par l'intermédiaire de la variable *EXTEN*.

D'autres variables sont utilisables dans différents contextes. Le tableau ci-dessous donne quelques exemples.

VARIABLES	SIGNIFICATIONS
\${EXTEN}	Fait référence à l'extension appelée.
\${CONTEXT}	Fait référence au contexte actuel.
\${CALLERID(name)}	Renvoie le nom de la personne qui appelle.
\${CALLERID(num)}	Renvoie le numéro de la personne qui appelle.

Les lignes n°4 et 5 permettent donc de composer un appel et de basculer vers la boîte vocale si le destinataire ne décroche pas son combiné dans les 20 secondes.

→Les contextes

Les contextes sont les blocs définis entre crochets. Ils serviront à faire référence aux services *compta* et *finance* de notre cas.

Dans la ligne n°8, les utilisateurs du contexte *finance* doivent composer le 1199 pour accéder à leur boîte vocale.

D4.2 – Configuration du DialPlan

Après avoir sauvegardé le fichier *extensions.conf* d'origine, la première étape consiste à définir un bloc de configuration nommé *general*.

→ Le bloc de configuration générale

```
[general]
static = yes           ;le DialPlan est statique.
writeprotect = yes     ;On ne peut pas le modifier depuis le CLI.
clearglobalvars = yes  ;les variables sont effacées et recalculées à chaque redémarrage d'Asterisk.
```

Puis, les contextes sont mis en place dans des blocs contenant les extensions.

→ Création des contextes

```
[finance]
;plan de numérotation du service finance
;voir l'extrait du fichier de configuration fourni en amont.

[compta]
;plan de numérotation du service compta
...
```

Les utilisateurs d'un même contexte pourront se joindre lorsque leur téléphone sera lié au serveur.

Après redémarrage du service Asterisk, on peut vérifier la prise en compte du « dial plan » :

```
asterisk*CLI> dialplan show

A ce niveau, recharger la configuration d'Asterisk pour prendre en compte les modifications effectuées. Il ne doit plus y avoir d'erreurs ou de warnings.
```

Document 5 – Utilisation d'un softphone

D5.1 – Installation d'un softphone

Afin de commencer nos premiers tests, il peut être intéressant d'utiliser un logiciel émulant le fonctionnement d'un téléphone IP (softphone). Plusieurs solutions sont disponibles, libres ou propriétaires (Ekiga, Blink, TeamSpeak...).

Ce côté labo met en avant les softphones Ekiga et Blink disponibles sur Windows ou Linux. Pour le téléchargement, il faut aller sur le site officiel : www.ekiga.org et icanblink.com. Pour les machines Linux de type Ubuntu ou Debian, Ekiga peut s'installer à partir des dépôts.

```
#apt-get install ekiga
```

D5.2 – Création d'un compte sur le softphone

La création d'un compte consiste à renseigner les champs permettant d'enregistrer le téléphone en l'associant à un utilisateur déjà créé sur le serveur. Voici un exemple avec *Ekiga* qu'il faut adapter selon le softphone utilisé. Il s'agit de créer un compte de type SIP.

Dans le champ *Registar*, il faut indiquer l'adresse IP du serveur Asterisk.

Le champ *User* fait référence au numéro de téléphone de l'utilisateur créé (fichier *users.conf*).

Le mot de passe est celui indiqué dans le fichier *users.conf* pour l'utilisateur concerné.

Il convient d'être attentif au message renvoyé par Ekiga lors de l'enregistrement du compte.

 **Utilisateur3Finance Inscrit**

Le positionnement du serveur en mode CLI permet d'observer les traces de l'enregistrement du compte.

```
-- Registered SIP '1103' at 192.168.1.11:5060
> Saved useragent "Ekiga/4.0.1" for peer 1103
```

En cas de problème, l'activation du mode *debug* permet d'obtenir les traces des requêtes échangées.

```
asterisk*CLI> sip set debug on
```

Enfin, une commande permet de lister les comptes de téléphones associés au serveur (peers) : *sip show peers*. La capture suivante confirme l'enregistrement d'un premier compte.

```
asterisk*CLI> sip show peers
Name/username      Host              Dyn Forcerport
Comedia    ACL Port      Status      Description
1101/u1                (Unspecified)      D Auto (No)
No                0      Unmonitored
1102/u2                (Unspecified)      D Auto (No)
No                0      Unmonitored
1103/u3      192.168.1.11      D Auto (No)
No                5060      Unmonitored
1104/u4      192.168.1.12      D Auto (No)
No                5060      Unmonitored
```

Cette opération est à répéter pour chacun des comptes à enregistrer. Il est possible de retirer un softphone associé au serveur avec la commande *sip unregister*.

```
asterisk*CLI> sip unregister 1101
```

D5.3 – Premiers appels

Lorsque le serveur est en mode CLI, il permet de voir en direct la trace des échanges avec le serveur quand un appel est passé.

Dans cette capture d'écran l'utilisateur *Utilisateur3* de numéro 1103 appelle l'utilisateur *Utilisateur4* de numéro 1104. Ces deux utilisateurs appartiennent au contexte **finance**.

```
6 sip peers [Monitored: 0 online, 0 offline Unmonitored: 2 online, 4 offline]
== Using SIP RTP CoS mark 5
-- Executing [1104@finance:1] Dial("SIP/1103-00000000", "SIP/1104,20") in new
stack
== Using SIP RTP CoS mark 5
-- Called SIP/1104
-- SIP/1104-00000001 is ringing
> 0x7fe88c0485a0 -- Probation passed - setting RTP source address to 192.
168.1.12:5062
-- SIP/1104-00000001 answered SIP/1103-00000000
-- Remotely bridging SIP/1103-00000000 and SIP/1104-00000001
== Spawn extension (finance, 1104, 1) exited non-zero on 'SIP/1103-00000000'
```

Dans cette autre capture d'écran, l'utilisateur *Utilisateur5* de numéro 1201 appelle l'utilisateur *Utilisateur6* de numéro 1202. Ces deux utilisateurs appartiennent au contexte **compta**.

```
== Using SIP RTP CoS mark 5
-- Executing [1202@compta:1] Dial("SIP/1201-00000000", "SIP/1202,20") in new
stack
== Using SIP RTP CoS mark 5
-- Called SIP/1202
-- SIP/1202-00000001 is ringing
-- SIP/1202-00000001 answered SIP/1201-00000000
-- Remotely bridging SIP/1201-00000000 and SIP/1202-00000001
> 0x7f74fc005fb0 -- Probation passed - setting RTP source address to 192.
168.1.11:5066
== Spawn extension (compta, 1202, 1) exited non-zero on 'SIP/1201-00000000'
```

Pour ce qui est de la messagerie vocale, les traces sont visibles sur la console du serveur.


```
-- SIP/1202-00000005 is ringing
-- Nobody picked up in 20000 ms
-- Executing [1202@compta:2] VoiceMail("SIP/1201-00000004", "1202@compta") i
n new stack
-- <SIP/1201-00000004> Playing 'vm-intro.gsm' (language 'en')
  > 0x7f74fc051a20 -- Probation passed - setting RTP source address to 192.
168.1.11:5078
-- <SIP/1201-00000004> Playing 'beep.gsm' (language 'en')
-- Recording the message
-- x=0, open writing: /var/spool/asterisk/voicemail/compta/1202/tmp/AM6kWX
format: wav, 0x7f74fc034b88
-- User hung up
== Spawn extension (compta, 1202, 2) exited non-zero on 'SIP/1201-00000004'
```

Les messages vocaux sont stockés sur le serveur.

```
root@asterisk:/var/spool/asterisk/voicemail/compta/1202/INBOX# ls
msg0000.txt  msg0000.wav
```

Après une première écoute, les messages sont transférés dans un autre sous-répertoire « old ».