

# **La VoIP (Voice over Internet Protocol, « voix sur IP »)**

*Il s'agit en fait de la voix transmise au travers d'un réseau numérique.*

## **→ Introduction et contextualisation**

Avant l'avènement des réseaux numériques, il n'existait que le téléphone classique. Il était utilisé sur un réseau appelé réseau téléphonique commuté (RTC) qui existe depuis qu'un certain Graham Bell a inventé le téléphone. C'est pourquoi le bon vieux téléphone classique perdure dans la plupart des entreprises. Il est basé sur une technique qui a largement fait ses preuves et qui est peu onéreuse : la commutation de circuits.

Dans un réseau d'entreprise à sites multiples, chaque réseau local est connecté à un réseau étendu plus vaste. Si vous vous trouvez au quartier général à New York et désirez contacter un collègue situé dans le bureau de Los Angeles, la Voip va faciliter cette communication longue distance. Au lieu de maintenir des réseaux séparés pour l'informatique et la téléphonie, l'entreprise peut les faire converger en un seul réseau utilisant à la fois la téléphonie sur IP et la VoIP.

## **→ Présentation fonctionnelle**

Contrairement à la téléphonie classique par commutation de circuits, qui repose exclusivement sur un réseau téléphonique commuté (RTC ou PSTN en anglais), la technologie VoIP permet de téléphoner – avec en prime des fonctionnalités nouvelles et innovantes – sur des réseaux spécialisés ou sans fil, y compris les réseaux informatiques. Ces nouveaux types de réseaux utilisent des protocoles « par commutation par paquets ».

En plus des données vocales, un paquet comporte les adresses réseau de l'expéditeur et du destinataire. Celles-ci peuvent être acheminées par des chemins différents, car en cas d'encombrement, l'adresse de destination se trouve dans chaque paquet. Il n'est pas du tout obligatoire que tous les paquets empruntent la même route.

Dans un réseau par commutation par paquets, si l'une des lignes est en panne, un paquet peut commuter d'un emplacement à un autre alors même qu'il est en route afin de préserver l'appel.

Il est important de noter la différence entre la VoIP qui est une technologie, et la ToIP qui est un service de téléphonie utilisant la VoIP.

Les communications sont réalisées via les deux protocoles suivants:

- Protocol SIP (Session Initiation Protocol) permet d'établir une session entre plusieurs participants mais ne transporte aucune voix.
- Protocol RTP (Real time Transport Protocol) permet de transporter les paquets entre les participants d'une session, c'est un flux unidirectionnel.

Pour simplifier l'utilisation des protocoles, prenons une conversation entre deux personnes.

La conversation en elle-même se fera via un Protocole SIP, chaque ligne de dialogues à l'intérieur de cette conversation correspond à un protocole RTP.

## **→ Avantages**

### **Moins de matériel physique à entretenir**

Les systèmes de téléphonie traditionnels nécessitent souvent des équipements coûteux tels que des commutateurs et des lignes téléphoniques fixes.

La VoIP utilise souvent le réseau informatique existant de l'entreprise, ce qui réduit le besoin de maintenir des infrastructures téléphoniques distinctes.

Une grande partie de cette infrastructure matérielle est remplacée par des logiciels et des services basés sur le cloud, ce qui réduit les coûts d'entretien et de remplacement du matériel.

### **Moins de dépannage sur site**

Les problèmes rencontrés avec les systèmes de VoIP peuvent souvent être résolus à distance par le fournisseur de services, ce qui réduit le besoin d'interventions sur site coûteuses.

### **Évolutivité simplifiée**

Il est généralement plus facile d'ajouter de nouvelles lignes ou de mettre à niveau les fonctionnalités avec la VoIP, ce qui réduit les coûts associés à la mise à l'échelle du système.

Les mises à jour logicielles pour les systèmes VoIP sont souvent gérées à distance par le fournisseur de services, ce qui signifie moins de temps et de ressources nécessaires pour les administrateurs informatiques internes.

### **Moins de frais de maintenance périodique**

Les systèmes de téléphonie traditionnels nécessitent souvent des contrats de maintenance coûteux pour assurer leur bon fonctionnement. Avec la VoIP, les frais de maintenance sont souvent inclus dans le coût global du service et peuvent être moins élevés.

### **Flexibilité**

Que ce soit dans le cas d'un centre d'appels ou d'un standard téléphonique bien que ce soit deux systèmes de télécommunications différents, conçus pour répondre à des besoins distincts en matière de communication dans une entreprise, la VoIP dispose de nombreuses configurations qui permettront de remplir ces deux fonctions.

En résumé, la configuration VoIP d'un centre d'appels est généralement plus axée sur les fonctionnalités avancées de gestion des appels via des systèmes ACD (Automatic Call Distribution), et le routage intelligent en fonction du service demandé ou de la langue de l'appelant, tandis qu'un standard téléphonique d'entreprise peut se concentrer davantage sur les besoins de communication interne et externe de l'entreprise sans les exigences spécifiques des centres d'appels.

## **→ Inconvénients**

### **Dépendance à l'Internet**

La VoIP nécessite une connexion Internet stable pour fonctionner correctement. En cas de panne de réseau ou de bande passante insuffisante, la qualité des appels peut être affectée, voire interrompue.

### **Qualité de service variable**

Même avec une connexion Internet fiable, la qualité des appels VoIP peut varier en fonction de la quantité de données qui transite sur le réseau, de la qualité des connexions des utilisateurs et d'autres facteurs. Cela peut entraîner des appels de qualité inférieure par rapport aux systèmes téléphoniques traditionnels.

### **Sécurité et confidentialité**

Comme les appels VoIP sont basés sur des données numériques, ils peuvent être plus sujets aux interceptions et aux cyberattaques que les appels téléphoniques traditionnels. Les entreprises doivent prendre des mesures de sécurité appropriées pour protéger la confidentialité des communications VoIP.

### **Voici quelques exemple de méthode de chiffrage les plus utilisées avec la VoIP:**

- **TLS (Transport Layer Security)** : TLS est un protocole de sécurité largement utilisé pour sécuriser les communications sur Internet. Dans le contexte de la VoIP, TLS peut être utilisé pour sécuriser les sessions de signalisation SIP (Session Initiation Protocol) et d'autres protocoles de contrôle de la communication, garantissant ainsi l'authenticité et la confidentialité des échanges.
- **DTLS (Datagram Transport Layer Security)** : DTLS est une version de TLS adaptée pour les communications en mode " datagramme ou communication par paquets". Il fournit une couche de sécurité similaire à TLS mais adaptée aux exigences des communications en temps réel, offrant ainsi une protection contre les attaques telles que la falsification de paquets et l'interception.
- **SRTP (Secure Real-time Transport Protocol)** : protocole de chiffrement conçu spécifiquement pour sécuriser les transmissions en temps réel. Il utilise des algorithmes de chiffrement comme AES (Advanced Encryption Standard) pour garantir la confidentialité des données vocales et des mécanismes d'intégrité pour détecter toute altération des données.
- **ZRTP (Zimmermann Real-time Transport Protocol)** : protocole de chiffrement spécialement conçu pour sécuriser les communications vocales en peer-to-peer. Il permet l'établissement sécurisé de clés de chiffrement sans nécessiter d'infrastructure de clé publique, ce qui en fait une option attrayante pour les applications VoIP axées sur la confidentialité et la simplicité.

### **Alimentation électrique et Compatibilités des équipements :**

Certains systèmes VoIP nécessitent une alimentation électrique pour fonctionner, ce qui signifie qu'en cas de panne de courant, les appels téléphoniques peuvent être interrompus si les dispositifs ne disposent pas de systèmes de secours appropriés. De plus, la VoIP peut nécessiter des équipements spécifiques, tels que des téléphones IP ou des adaptateurs, pour fonctionner correctement. Assurer la compatibilité de ces équipements avec les infrastructures existantes peut être un défi.

## **→ Solutions existantes sur le marché (Open Source/payantes)**

Toutes ces solutions sont appelées PBX,(Private Branch Exchange), qui est un système de télécommunications utilisé dans les entreprises et les organisations pour gérer les appels téléphoniques internes et externes.

Il connecte plusieurs lignes téléphoniques internes à un seul système central, permettant ainsi aux utilisateurs de partager des lignes externes pour passer des appels et communiquer entre eux via des appels internes.

Les PBX peuvent offrir diverses fonctionnalités telles que la messagerie vocale, le transfert d'appels, les conférences téléphoniques, etc. Ils sont souvent utilisés pour optimiser la gestion des appels et réduire les coûts de communication pour les entreprises.

### **Solutions payantes :**

#### **1. Cisco Unified Communications Manager (CUCM) :**

C'est l'une des solutions VoIP les plus populaires sur le marché. Elle offre une large gamme de fonctionnalités pour la voix, la vidéo, la messagerie instantanée et la présence.

#### **2. Avaya Aura Communication Manager :**

Avaya propose une suite de solutions de communication unifiée qui inclut le gestionnaire de communication Aura. Cela offre des fonctionnalités avancées de VoIP, de messagerie instantanée, de vidéo et de collaboration.

#### **3. Microsoft Teams :**

Bien que principalement connu pour sa fonction de messagerie instantanée et de collaboration, Microsoft Teams intègre également des fonctionnalités de VoIP via des abonnements Office 365 Entreprise. Attention cependant dans certains forfaits proposer par la suite 365 microsoft teams n'est plus compris.

#### **4. 3CX :**

Une solution VoIP populaire qui offre une variété de fonctionnalités, y compris la visioconférence, la messagerie instantanée et la collaboration.

### **Solutions open source :**

#### **1. Asterisk :**

C'est l'une des solutions VoIP open source les plus populaires et les plus anciennes. Asterisk offre une grande flexibilité et une large gamme de fonctionnalités, et il est très personnalisable.

#### **2. FreeSWITCH :**

Une autre alternative open source à Asterisk, FreeSWITCH est une plateforme de communication en temps réel qui peut gérer la voix, la vidéo, le chat et d'autres types de médias.

#### **3. Kamailio :**

Conçu principalement comme un serveur SIP, Kamailio est souvent utilisé pour construire des solutions de téléphonie VoIP évolutives et robustes.

#### **4. OpenSIPS :**

Similaire à Kamailio, OpenSIPS est un serveur SIP open source qui peut être utilisé pour construire des systèmes de téléphonie VoIP à grande échelle.

## **→ Exemples d'implémentation**

Après quelques recherches il m'a sauté aux yeux que la quasi totalité des sites marchand / service client (Ebay, Rueducommerce, Amazone, LDLC, etc...) on recours à une architecture VoIP pour gérer les communications téléphoniques de leur entreprise.

Les détails spécifiques propres à chaque entreprise étant logiquement considérés comme des informations sensibles ne sont pas rendus publics par souci de sécurité et de confidentialité. Il m'a fallu donc imaginer à quoi pourrait ressembler leur infrastructure.

### **1. Serveur VoIP**

Un serveur VoIP constitue le cœur de l'architecture. Il est responsable de la gestion des appels entrants et sortants, à la fois pour les communications avec les clients et en interne entre les employés. Ce serveur peut être hébergé en interne ou dans le cloud, en fonction des besoins et des ressources de l'entreprise. Il gère la création des canaux SIP chargés de supporter les communications.

### **2. Infrastructure réseau**

Une infrastructure réseau robuste est essentielle pour assurer la qualité des appels VoIP, à la fois pour les communications externes avec les clients et internes entre les employés. Cela comprend des commutateurs Ethernet, des routeurs, des pare-feu et des passerelles VoIP pour faciliter la connectivité avec le réseau téléphonique public commuté (PSTN) si nécessaire.

### **3. Téléphones IP ou Softphones**

Les employés utilisent des téléphones IP physiques ou des softphones installés sur leurs ordinateurs ou appareils mobiles pour passer et recevoir des appels VoIP en interne. Les téléphones IP sont connectés au réseau Ethernet, tandis que les softphones utilisent le réseau de données de l'entreprise ou d'Internet pour établir une connexion VoIP.

### **4. Logiciel de centre d'appels**

Les agents du service client utilisent des logiciels de centre d'appels dotés de fonctionnalités VoIP, pour répondre aux appels des clients. Ces logiciels offrent des fonctionnalités avancées telles que le routage intelligent des appels, l'accès aux informations client et la gestion des files d'attente pour améliorer l'efficacité opérationnelle.

Par exemple, une fonctionnalité de gestion des tickets du service client pour enregistrer les interactions des clients. Les appels entrants sont automatiquement enregistrés en tant que tickets dans le système, ce qui permet aux agents du service client de les traiter efficacement et de suivre l'historique des communications.

De même, les employés peuvent utiliser des logiciels de communication interne pour faciliter la collaboration et les échanges au sein de l'entreprise.

### **5. Sécurité**

La sécurité des communications VoIP est primordiale, aussi bien pour les communications externes avec les clients que pour les communications internes entre les employés. Des mesures de sécurité telles que le chiffrement des appels, l'authentification des utilisateurs, la détection des fraudes et la protection contre les attaques par déni de service (DoS) sont mises en place pour garantir la confidentialité et l'intégrité des communications, qu'elles soient externes ou internes.

## **→ Conclusion**

En conclusion, la VoIP a révolutionné la manière dont les entreprises gèrent leurs communications téléphoniques. En permettant de transmettre la voix sur des réseaux numériques, la VoIP offre une flexibilité, une évolutivité et des fonctionnalités avancées qui dépassent de loin les capacités des systèmes téléphoniques traditionnels.

Les avantages de la VoIP, tels que la réduction des coûts, la facilité de maintenance et la flexibilité, en font un choix attrayant pour de nombreuses entreprises, qu'il s'agisse de petites startups ou de grandes entreprises multinationales.

Cependant, la VoIP n'est pas sans défis. La migration complexe d'un système traditionnel vers celle-ci, la dépendance à Internet, la qualité de service variable et les préoccupations en matière de sécurité et de confidentialité sont autant de considérations importantes à prendre en compte lors de l'adoption de cette technologie.

Malgré ces défis, la VoIP continue de gagner en popularité et reste un pilier essentiel de la communication d'entreprise moderne. Avec des solutions innovantes et une évolution constante de la technologie, la VoIP continuera probablement de jouer un rôle central dans le paysage des communications pour les années à venir.

## → Mise en place de mon serveur VoIP

Tuto : [Asterisk\\_config/Voip.pdf at main · thierry-rami/Asterisk\\_config · GitHub](https://thierry-rami.github.io/Asterisk_config/Voip.pdf)

Lors du démarrage de la VM rien ne s'affiche avant la demande de login.

Afin d'éviter d'ouvrir une session pour taper ip a, l'utilisation de ces deux commandes permet de voir les adresses ip de la vm.

Ces commandes sont tapées en root.

```
root@debian:~# echo "IPv4\4" >> /etc/issue
root@debian:~# echo "IPv6\6" >> /etc/issue
root@debian:~#
```

```
Debian GNU/Linux 12 debian tty1

IPv4192.168.198.135
IPv6fe80::20c:29ff:fe34:6f0c
debian login:
```

Par défaut il faut se connecter a un utilisateur, puis tapez su - pour passer en root, par défaut et surtout par mesure de sécurité la connexion à distance au compte root en SSH n'est pas autorisée.

Via cette commande j'autorise la connexion en ssh au compte root de ma vm avec putty :

```
echo "PermitRootLogin Yes" > /etc/ssh/sshd_config.d/ssh.conf
```

```
login as: root
root@192.168.198.135's password:
Linux debian 6.1.0-18-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.76-1 (2024-02-01)
x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Apr 25 11:21:50 2024 from 192.168.198.1
root@debian:~#
```

On installe les paquets nécessaires pour compiler Asterisk :

```
apt install git curl build-essential libedit-dev git curl wget libnewt-dev
libssl-dev libncurses5-dev subversion libsqlite3-dev build-essential
libjansson-dev libxml2-dev uuid-dev avahi-daemon libsrtplib2-dev
```

Nous allons télécharger la dernière version d'Asterisk dans le répertoire /usr/src. L'utilisation de ce répertoire est nécessaire pour compiler le logiciel, de plus il s'agit d'une bonne pratique linux.

```
apt install wget
wget
http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-21-current.tar.gz
tar xfv asterisk-21-current.tar.gz
rm asterisk-21-current.tar.gz
cd asterisk-21*
```

Ces commandes sont utilisées lors de la configuration préliminaire d'Asterisk pour s'assurer que les dépendances nécessaires sont satisfaites et que les fonctionnalités souhaitées sont incluses dans la compilation du logiciel.

```
./configure --with-srtp
```

1. **./configure** : Cette commande lance le script de configuration qui prépare le processus de compilation pour Asterisk. Ce script vérifie les dépendances système, configure les options de compilation et génère les fichiers nécessaires pour le processus de compilation.
2. **--with-srtp** : C'est une option passée au script de configuration. Cette option indique à Asterisk d'inclure le support pour SRTP

```
configure: Menuselect build configuration successfully completed
```

```

      .$$$$$$$$$$$$$$$$$=..
      .7$77..      .7$77:..
      .$$:..      ,77.7
      .77.      7$$$$$      .7$77
      ..$$      $$$$$$      .7$77
      ..7$ .? . $$$$$$ .? . 7$$$$.
      $.$.      .7$77. 7$77. 7$77. 7$77.
      .777.      .7$77. 7$77. 7$77. 7$77.
      $$$~      .7$77. 7$77. 7$77. 7$77.
      .7$77      .7$77. 7$77. 7$77. 7$77.
      $$$      ?7$77. 7$77. 7$77. 7$77.
      $$$      .7$77. 7$77. 7$77. 7$77.
      $$$      $$$ 7$77. 7$77. 7$77.
      $$$      $$$ 7$77. 7$77. 7$77.
      7$77      7$77. 7$77. 7$77. 7$77.
      $$$7.      $$$ (TM)
      $$$$$$.      .7$77. 7$77. 7$77.
      $$$$$$7$77. 7$77. 7$77. 7$77.
      $$$$$$7$77. 7$77. 7$77. 7$77.

```

```
configure: Package configured for:
configure: OS type : linux-gnu
configure: Host CPU : x86_64
configure: build-cpu:vendor:os: x86_64 : pc : linux-gnu :
configure: host-cpu:vendor:os: x86_64 : pc : linux-gnu :
root@debian:/usr/src/asterisk-21.2.0#
```

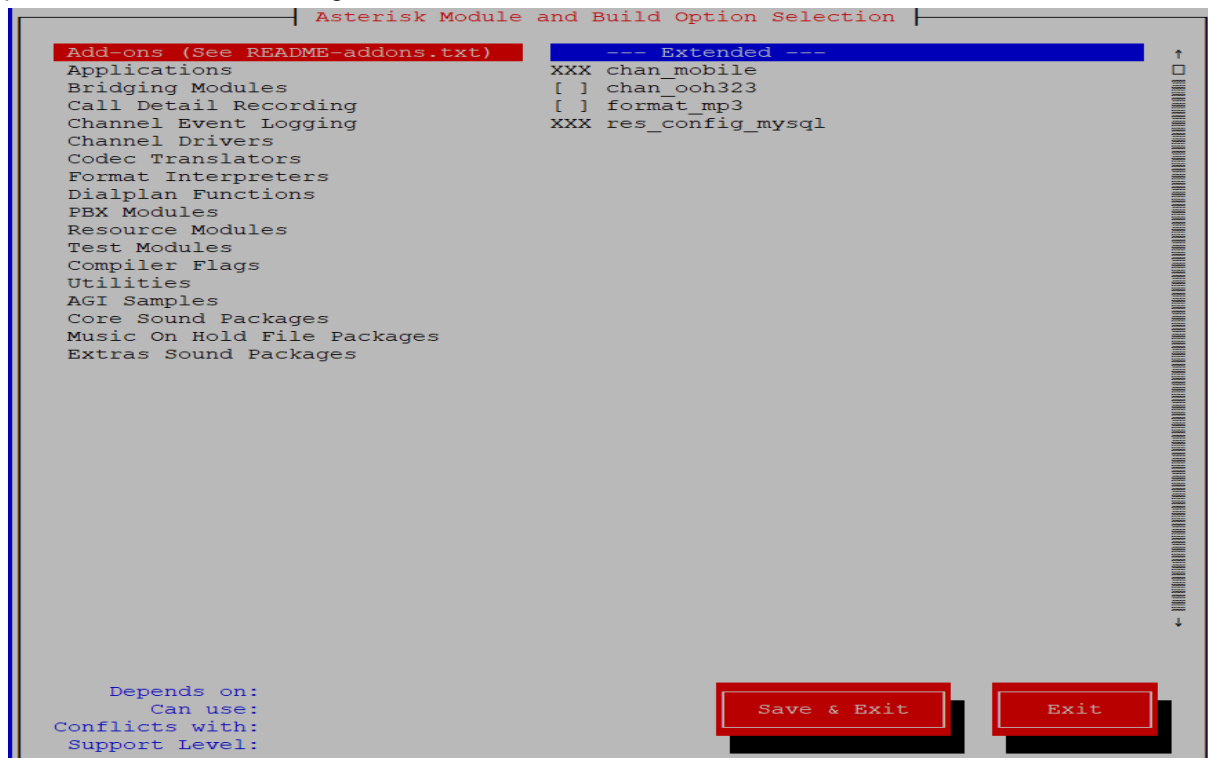
```
contrib/scripts/get_mp3_source.sh
```

Elle est utilisée pour obtenir le code source de l'extension MP3 nécessaire pour lire les fichiers audio au format MP3 dans Asterisk. Cette commande télécharge le code source depuis un référentiel en ligne et le place dans le répertoire approprié pour être utilisé lors de la compilation d'Asterisk.

```
root@debian:/usr/src/asterisk-21.2.0# contrib/scripts/get_mp3_source.sh
A   addons/mp3
A   addons/mp3/layer3.c
A   addons/mp3/interface.c
A   addons/mp3/MPGLIB_TODO
A   addons/mp3/mpg123.h
A   addons/mp3/mpglib.h
A   addons/mp3/decode_ntom.c
A   addons/mp3/MPGLIB_README
A   addons/mp3/common.c
A   addons/mp3/huffman.h
A   addons/mp3/tabinit.c
A   addons/mp3/Makefile
A   addons/mp3/README
A   addons/mp3/decode_i386.c
A   addons/mp3/dct64_i386.c
Exporté à la révision 204.
root@debian:/usr/src/asterisk-21.2.0#
```



Via la commande `make menuselect` nous accédons au menu de configuration d'asterisk.  
 Cette étape est nécessaire pour configurer notre serveur en français.  
 Afin de choisir ou non des options dans ce menu il suffit de faire entrer la présence ou non d'une petite étoile entre crochet signifie si le service est activé ou désactivé.



Dans "Core Sound Packages" :

```
[*] CORE-SOUNDS-FR-WAV
[*] CORE-SOUNDS-FR-ULAW
[*] CORE-SOUNDS-FR-ALAW
[*] CORE-SOUNDS-FR-GSM
[*] CORE-SOUNDS-FR-G729
[*] CORE-SOUNDS-FR-G722
[*] CORE-SOUNDS-FR-SLN16
[*] CORE-SOUNDS-FR-SIREN7
```

Dans "Extra Sound Packages" :

```
[*] EXTRA-SOUNDS-FR-WAV
[*] EXTRA-SOUNDS-FR-ULAW
[*] EXTRA-SOUNDS-FR-ALAW
[*] EXTRA-SOUNDS-FR-GSM
[*] EXTRA-SOUNDS-FR-G729
[*] EXTRA-SOUNDS-FR-G722
[*] EXTRA-SOUNDS-FR-SLN16
[*] EXTRA-SOUNDS-FR-SIREN7
[*] EXTRA-SOUNDS-FR-SIREN14
```

On sauvegarde les configurations via Save & Exit.

A chaque compilation, lorsqu'il s'agit d'un programme en "C" 2 ou 3 étapes sont nécessaires :

- `./configure` qu'on a réalisée précédemment pour installer le serveur.
- `make`

```
+----- Asterisk Build Complete -----+  
+ Asterisk has successfully been built, and +  
+ can be installed by running:           +  
+                                         +  
+             make install                +  
+-----+
```

- ```
- make install
```

```
+----- Asterisk Installation Complete -----+
+
+   YOU MUST READ THE SECURITY DOCUMENT      +
+
+ Asterisk has successfully been installed.  +
+ If you would like to install the sample   +
+ configuration files (overwriting any      +
+ existing config files), run:              +
+
+ For generic reference documentation:      +
+     make samples                          +
+
+ For a sample basic PBX:                   +
+     make basic-pbx                       +
+
+----- or -----+
+
+ You can go ahead and install the asterisk +
+ program documentation now or later run:   +
+
+     make progdocs                         +
+
+ **Note** This requires that you have     +
+ doxygen installed on your local system   +
+-----+

```

La compilation étant terminée on va ajouter quelques commandes afin d'ajouter des fonctionnalités à notre serveur.

- `make samples` simplifie le processus de configuration d'Asterisk en fournissant des fichiers de configuration d'exemple commentés et prêts à être utilisés comme point de départ pour personnaliser la configuration du système de téléphonie.
- `make basic-pbx` simplifie le processus de configuration initiale d'Asterisk en générant automatiquement une configuration de base pour un système PBX, ce qui permet aux utilisateurs de démarrer rapidement avec Asterisk et de commencer à effectuer des appels téléphoniques.
- `make config` est une commande importante qui finalise le processus de configuration d'Asterisk en copiant les fichiers de configuration dans les bons emplacements et en préparant le système pour le démarrage du service Asterisk.

Une fois terminé on lance Asterisk et on vérifie son statut :

- `service asterisk start`
- `service asterisk status`

```
root@debian:/# service asterisk start
root@debian:/# service asterisk status
● asterisk.service - LSB: Asterisk PBX
   Loaded: loaded (/etc/init.d/asterisk; generated)
   Active: active (running) since Mon 2024-04-29 09:08:01 CEST; 8s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 28537 ExecStart=/etc/init.d/asterisk start (code=exited, status=0/SUCCESS)
    Tasks: 46 (limit: 4603)
   Memory: 32.9M
      CPU: 438ms
   CGroup: /system.slice/asterisk.service
           └─28549 /usr/sbin/asterisk

avril 29 09:08:00 debian systemd[1]: Starting asterisk.service - LSB: Asterisk PBX...
avril 29 09:08:01 debian asterisk[28537]: Starting Asterisk PBX: asterisk.
avril 29 09:08:01 debian systemd[1]: Started asterisk.service - LSB: Asterisk PBX.
root@debian:/#
```

On passe à la modification des fichiers de configuration. `cd /etc/asterisk`

NB = Les “ ; ” en début de ligne sert à commenté à la place des “ # ”

Ce fichier permet de choisir la langue.

- `nano asterisk.conf`

```
GNU nano 7.2                  asterisk.conf *
[options]
; If we want to start Asterisk with a default verbosity for the verbose
; or debug logger channel types, then we use these settings (by default
; they are disabled).
;verbose = 5
;debug = 2

; User and group to run asterisk as. NOTE: This will require changes to
; directory and device permissions.
;runuser = asterisk           ; The user to run as. The default is root.
;rungroup = asterisk          ; The group to run as. The default is root

languageprefix = yes
defaultlanguage = fr
```

**Les fichiers suivant étant trop long a screenshot je les ai ajouter au github qui contient cette DOC**

Ce fichier permet de créer les utilisateurs.

- `nano pjsip.conf`

Ce fichier permet de configurer les numéros d'appels de nos utilisateurs

- `nano extensions.conf`

Ce fichier permet de configurer la messagerie vocale

- `nano voicemail.conf`

Puis un petit `service asterisk restart` pour appliquer les configurations

Afin de rentrer dans la console de commande Asterisk, il faut taper :

- asterisk -rvvvvvvv ( plus il y a des v plus il a de commentaires dans la console )

```
=====
Connected to Asterisk 21.2.0 currently running on debian (pid = 28696)
Unable to read or write history file '/root/.asterisk_history'
debian*CLI> █
```

debian\*CLI> signifie que je suis dans la console Asterisk

pjsip list aors =====> commande pour voir les Users.

## Installation des logiciels Clients sur une autre console VM afin de tester le fonctionnement de mon serveur VoIP

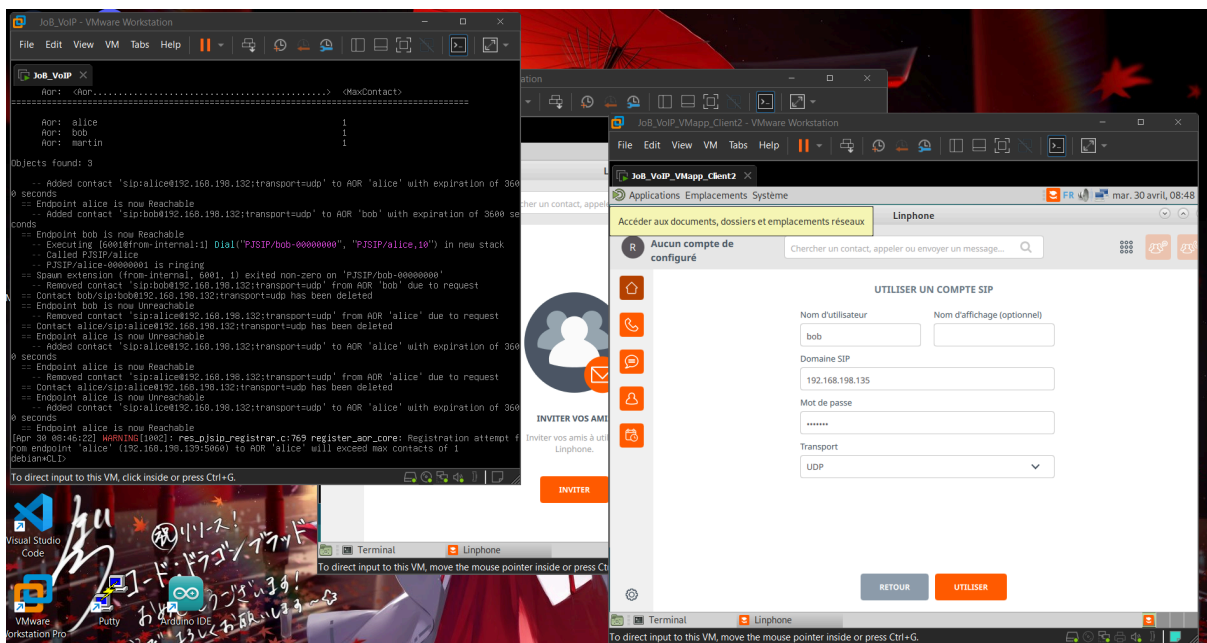
Sur ma vm Linux client j'ai téléchargé l'app image Linphone puis chmod +x et on lance le fichier pour démarrer le logiciel.

```
root@Linux:~# cd Téléchargements/
root@Linux:~/Téléchargements# chmod +x Linphone-5.2.4.AppImage
root@Linux:~/Téléchargements# ./Linphone-5.2.4.AppImage
```

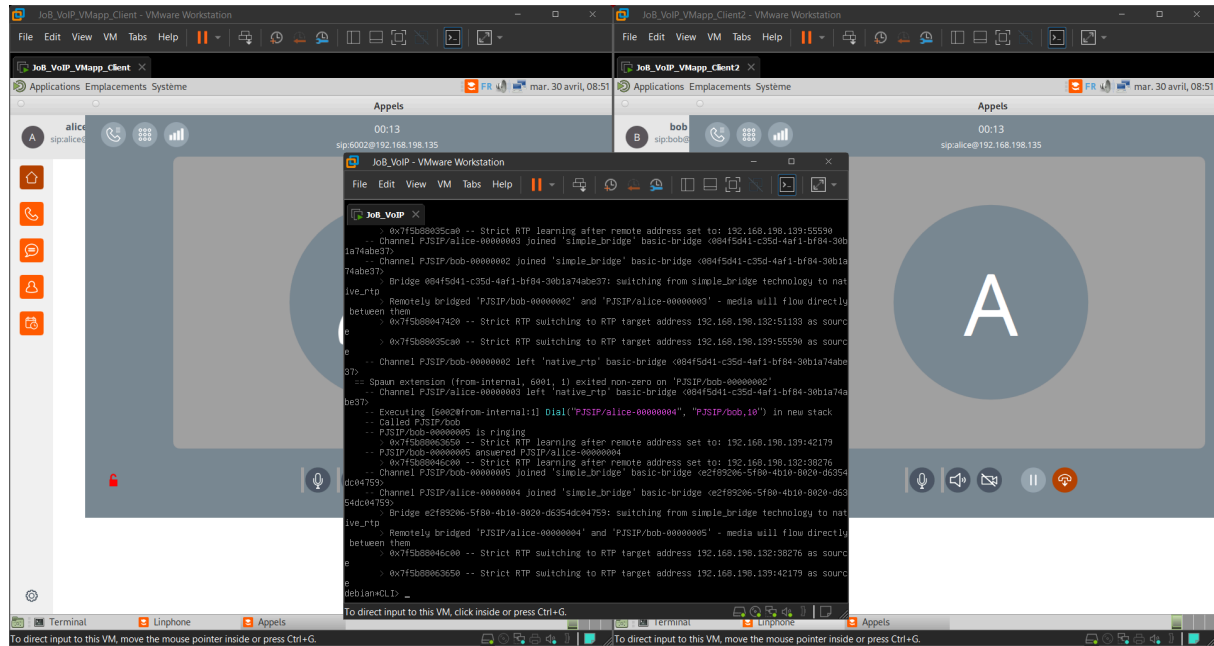
NB = - respecter les majuscules minuscules des utilisateurs saisi dans le fichier config, j'avais écrits Alice plutôt que alice lors de la création de mon premier compte SIP.

- Le protocole utilisé par le SIP est le protocole UDP, veillez à bien sélectionner celui-ci lors de la création d'un utilisateur.

On se retrouve donc avec une VM serveur + 2 VM clients



Et voila ce que l'on obtient quand Alice appel Bob



Pour la partie Automatisation et création du standard :

- `nano /etc/apt/sources.list` pour supprimer le CD-Rom des sources comme d'hab
- `apt update / upgrade` pour mettre ma vm à jour
- `apt install python3-pip` pour installer le gestionnaire de librairie python
- `pip install gTTS` qui est l'API google Text-To-Speech

Toutes ces manipulations me permettent de créer un fichier vocal qui sert de message d'accueil pour mon standard.

Ensuite nous allons dans le répertoire qui permet de stocker les fichiers audio de mon serveur Asterisk

- `cd /var/lib/asterisk/sounds/fr`

Et pour finir nous utilisons l'API pour créer notre message d'accueil

- `gtts-cli "Bienvenue à La Plateforme, tu es actuellement sur mon serveur Asterisk qui souhaites tu joindre ? Faites le 1 pour Alice, le 2 pour Bob, le 3 pour Martin" -l fr --output bonjour.mp3`

**Attention à bien mettre le -l fr sinon vous aurais du FRANÇAIS.**

Il était impossible de lire un fichier mp3 j'ai donc installer sox qui permet de convertir les fichiers mp3 en fichier ulaw compatible avec asterisk.

- `apt install perl libwww-perl sox mpg123` ==> Paquets de SOX
- `apt install libsox-fmt-mp3` ==> Paquets du Format MP3

Et pour finir dans le fichier extensions.conf, il faut faire la configuration du standard.

```
;!
; Automatically generated configuration file
; Filename: extensions.conf (/etc/asterisk/extensions.conf)
; Generator: Manager
; Creation Date: Tue Jun 20 08:31:38 2023
;!

[from-internal]
exten => 6001,1,Dial(PJSIP/alice,10)
exten => 6002,1,Dial(PJSIP/bob,10)
exten => 6003,1,Dial(PJSIP/martin,10) ; Après 10 secondes envoi vers la règle 2
exten => 6099,1,VoiceMailMain() ; 6099 Numéro de téléphone du répondeur
exten => 7000,1,Goto(menu,s,1) ; 7000 Numéro du standard pour l'exercice Automatisation
; Règles 2
exten => 6001,2,VoiceMail(6001) ; Appel répondeur compte 6001
exten => 6002,2,VoiceMail(6002) ; Appel répondeur compte 6002
exten => 6003,2,VoiceMail(6003) ; Appel répondeur compte 6003

; Config du Standard
[menu]
exten => s,1,Answer()
    same => n,Playback(bonjour) ; Lecture du Fichier bonjour
    same => n,WaitExten(5)
    same => n,Goto(menu,s,1)

exten => 1,1,Dial(PJSIP/alice)
    same => n,Hangup()

exten => 2,1,Dial(PJSIP/bob)
    same => n,Hangup()

exten => 3,1,Dial(PJSIP/martin)
    same => n,Hangup()
```

**Attention pour le " Dial " a mettre PJSIP suivis de l'identité de l'utilisateur et non son numéro !**