

VoIP

BACHELOR IT – ADMINISTRATION SECURITE DES SI
YOUCEF OUARHLENT
LUCAS SAVIOZ
RIJA RASOANAIVO

Contexte

Son coût réduit. En effet, les appels VoIP sont généralement moins chers que les appels téléphoniques traditionnels.

Peut-on dire la même chose de ses coûts opérationnels et de maintenance ?

1. Coûts opérationnels initiaux :

- **VoIP** : Pour passer aux appels VoIP, vous pourriez devoir investir davantage au départ. Cela inclut l'achat de téléphones IP et d'autres équipements spécifiques, ainsi que la formation du personnel pour utiliser ce nouveau système.

- **Téléphonie traditionnelle** : Les systèmes téléphoniques traditionnels nécessitent généralement moins d'investissement initial, car vous utilisez les lignes téléphoniques existantes et les téléphones standard.

2. Coûts d'exploitation à long terme :

- **VoIP** : Une fois le système mis en place, les coûts d'exploitation peuvent être réduits car les appels VoIP passent par le même réseau que vos données informatiques, ce qui peut vous faire économiser sur les frais de ligne téléphonique traditionnels.

- **Téléphonie traditionnelle** : Vous continuerez à payer des frais de ligne téléphonique réguliers, ce qui peut s'avérer plus coûteux à long terme.

3. Coûts de maintenance :

- **VoIP** : Les systèmes VoIP nécessitent une surveillance et une maintenance régulières pour assurer un bon fonctionnement. Cela comprend les mises à jour logicielles, la gestion de la bande passante et la résolution des problèmes liés à la qualité des appels.

- **Téléphonie traditionnelle** : Bien qu'ils nécessitent également de la maintenance, les systèmes téléphoniques traditionnels peuvent parfois être plus simples à entretenir car ils reposent sur une infrastructure physique éprouvée.

En résumé, tandis que les appels VoIP sont souvent moins chers que les appels téléphoniques traditionnels, le coût total peut varier en fonction de la taille de votre entreprise, de la complexité de votre système et de vos besoins en termes de qualité et de fiabilité des appels.

Sa flexibilité : Un serveur VoIP offre un large panel de configuration permettant de personnaliser l'expérience utilisateur.

Pourriez-vous dire en quoi la configuration VoIP d'un call center serait différente de la configuration VoIP d'un standard téléphonique d'une entreprise ?

La téléphonie IP offre une **flexibilité considérable** par rapport à la téléphonie standard, avec **des avantages tels que l'absence de limites géographiques et d'itinérance**, une **gamme étendue de fonctionnalités** et une **absence de contrats** contraignants. Elle permet également aux entreprises de **se déployer géographiquement plus facilement** en utilisant **un seul système téléphonique VoIP** pour plusieurs bureaux, **grâce à la nature du PBX hébergé**. Cette **flexibilité s'étend aux centres d'appels, où la configuration VoIP diffère de celle d'un standard téléphonique traditionnel** en raison de la nécessité de **gérer des volumes élevés d'appels**, des fonctionnalités avancées spécifiques aux centres d'appels et une **intégration aux systèmes CRM**, tout en garantissant une flexibilité et une évolutivité maximales.

Définition :

Un CRM (Customer Relationship Management) est un logiciel centralisant toutes les données des clients et des prospects d'une entreprise. Il enregistre les interactions avec les clients et partage ces informations entre les différents départements. Un CRM aide les entreprises à gérer leurs relations client en offrant un accès instantané à toutes les demandes, préférences et échanges passés. Il permet aux équipes de fournir un service personnalisé en consignnant les appels, courriels et réunions clients, et en planifiant des actions futures telles que des relances ou des présentations. Ainsi, aucune opportunité n'est manquée, ce qui favorise la croissance de l'entreprise.

Un PBX, ou Private Branch Exchange, est un système téléphonique privé permettant aux entreprises de gérer efficacement leurs communications internes et externes. Il offre des fonctionnalités avancées telles que la messagerie vocale, le transfert d'appels et la conférence téléphonique, tout en permettant des économies de coûts en acheminant les appels via des réseaux internes. Le PBX est scalable et peut être déployé de différentes manières, offrant ainsi aux entreprises une flexibilité pour répondre à leurs besoins en communication.

Bien que la téléphonie IP réduise considérablement les frais téléphoniques, ces économies ne sont pas son seul avantage. La téléphonie IP offre un service beaucoup plus flexible que la téléphonie standard. Vous pouvez effectuer vos appels avec très peu de restrictions. Sans limitations géographiques, pratiques ou contractuelles, la téléphonie IP s'avère très flexible.

Itinérance sans limites

La téléphonie IP n'a pas de limites géographiques. Les frais interurbains et d'itinérances tels qu'on les connaît n'existent pas. Avec le forfait VoIP approprié, vous avez accès à des appels internationaux illimités et de l'itinérance gratuite.

La téléphonie IP permet aussi aux entreprises de s'étaler géographiquement avec un seul système téléphonique VoIP. Puisque la téléphonie IP fonctionne par PBX hébergé, plusieurs bureaux ou édifices peuvent se connecter à un même système VoIP. Répandre son entreprise

est beaucoup plus facile avec un système VoIP. Cette simplicité s'applique aussi aux branches internationales de votre entreprise.

Une large gamme de fonctionnalités

La téléphonie VoIP offre aussi une plus vaste flexibilité en matière de fonctionnalités que la téléphonie standard. Les fournisseurs VoIP vous offrent ainsi une plus grande sélection de fonctionnalités que les fournisseurs standards. Ces fonctionnalités incluent la messagerie avancée, la messagerie envoyée au courriel, un bloqueur d'identification de l'appelant, des appels à 3 lignes, des conférences téléphoniques et une réceptionniste automatique.

Un choix sans contrat

Possiblement la flexibilité la plus importante des trois, plusieurs fournisseurs VoIP offrent une liberté contractuelle incomparable. Avec la téléphonie IP, changer de plan ou de fournisseur est une affaire très simple. Plusieurs fournisseurs VoIP vous permettent d'annuler votre plan à tout moment et sans conséquence. L'absence de lignes téléphoniques propriétaires évite aussi un changement d'infrastructures.

Avec une telle liberté géographique, fonctionnelle et contractuelle, il va de soi que la téléphonie IP est l'option téléphonique la plus flexible au marché.

Les différents logiciels utilisés dans un centre d'appel :

Source principale : <https://fr.slideshare.net/apoorvatyagi2/call-centre-architecture>

ACD (Automatic Call Distribution) :

L'Automatic Call Distribution (ACD) ou distribution automatique des appels est un système qui permet de rediriger efficacement les appels entrants d'un contact center vers les agents disponibles. Ainsi, les entreprises peuvent gérer facilement ces flux d'appels et réduire le temps d'attente des appelants afin d'augmenter leur taux de satisfaction.

Source : <https://www.ringover.fr/automatic-call-distribution>

CTI (Couplage Téléphonie Informatique) :

La fonctionnalité CTI est très souvent utilisée par les centres d'appels, qui gèrent un nombre important d'appels, et cherchent continuellement à optimiser leur productivité. Un système CTI tel qu'un standard téléphonique VoIP de type logiciel permet avant tout de diminuer les coûts. Des logiciels clients de type soft phones existent, qui peuvent être téléchargés sur Windows, et qui rendent l'utilisation d'un téléphone physique obsolète. Un simple casque suffit, et les opérations les plus courantes telles que répondre, transférer ou mettre en pause un appel, sont faites en un clic de souris. Plus rentable, plus flexible et plus évolutif qu'un standard traditionnel, une solution permettant le couplage téléphonie-informatique permet aussi la fonction de hot desking, c'est-à-dire la possibilité de faire correspondre une session de travail à un agent plutôt qu'un poste téléphonique, permettant ainsi à chaque employé de travailler depuis différents ordinateurs, voire même à distance depuis leur domicile.

Source : [CTI, ou Couplage Téléphone Informatique : 8 atouts phares \(3cx.fr\)](#)

Predictive Dialing :

Les composeurs prédictifs s'appuient sur l'IA et le machine learning pour passer des appels à partir d'une liste de numéros prédéfinie de façon à éliminer toute perte de temps. Pour cela,

différents numéros sont composés simultanément. Les appels sont ensuite assignés automatiquement aux agents disponibles.

L'automate d'appel prédictif détermine le meilleur moment pour composer les numéros lorsque l'agent est le plus susceptible de pouvoir prendre un appel. L'objectif est d'éviter les temps d'inactivité en assurant un flux d'appel constant.

Utilisé pour passer un grand volume d'appels en un temps réduit, il permet également d'augmenter significativement les conversations directes entre l'agent et les clients ou prospects.

Source : [Power dialer ou predictive dialer : quel automate d'appel choisir | Aircall](#)

CMS (Call Management System) :

Le client moyen raccroche au bout de 2 minutes, le temps d'attente moyen étant de 56 secondes. Les appels interrompus sont des affaires perdues et une mesure par laquelle votre expérience client est évaluée.

Les systèmes de gestion des appels peuvent aider à gérer de gros volumes de clients. Le système de gestion des appels peut permettre aux entreprises d'offrir des expériences d'appel plus personnalisées. Cela contribue également à augmenter l'efficacité globale de votre processus de gestion des appels. Avec une bonne configuration de distribution automatique d'appels, vos appels seront attribués à l'agent le plus qualifié qui sera disponible.

Source : [Système de gestion des appels Voxco - Voxco](#)

Logiciel de comptabilisation des appels :

En utilisant un logiciel de gestion d'appel téléphonique, vous pouvez effectuer aisément votre suivi client. Il compile automatiquement tout l'historique d'appel et les solutions déjà proposées, ce qui permet d'orienter les réponses suivantes.

Le logiciel de gestion d'appel téléphonique fournit également une statistique des appels. Ce paramètre vous permet d'effectuer en temps réel, ou plus tard, une analyse des performances de votre entreprise et identifier ainsi les points à améliorer.

Source : [Logiciel de gestion d'appel téléphonique : pour faciliter vos tâches -Kavkom](#)

Logiciel de gestion des appels :

L'analyse des appels est une catégorie d'outils qui mesure les données individuelles et globales relatives aux appels entrants et sortants. Le suivi des appels comprend le volume d'appels entrants, la durée de l'appel, le temps d'attente, la rapidité de la réponse et bien d'autres paramètres.

Source : [VoIP Analytics & Call Reports | VoIP Features | Nextiva](#)

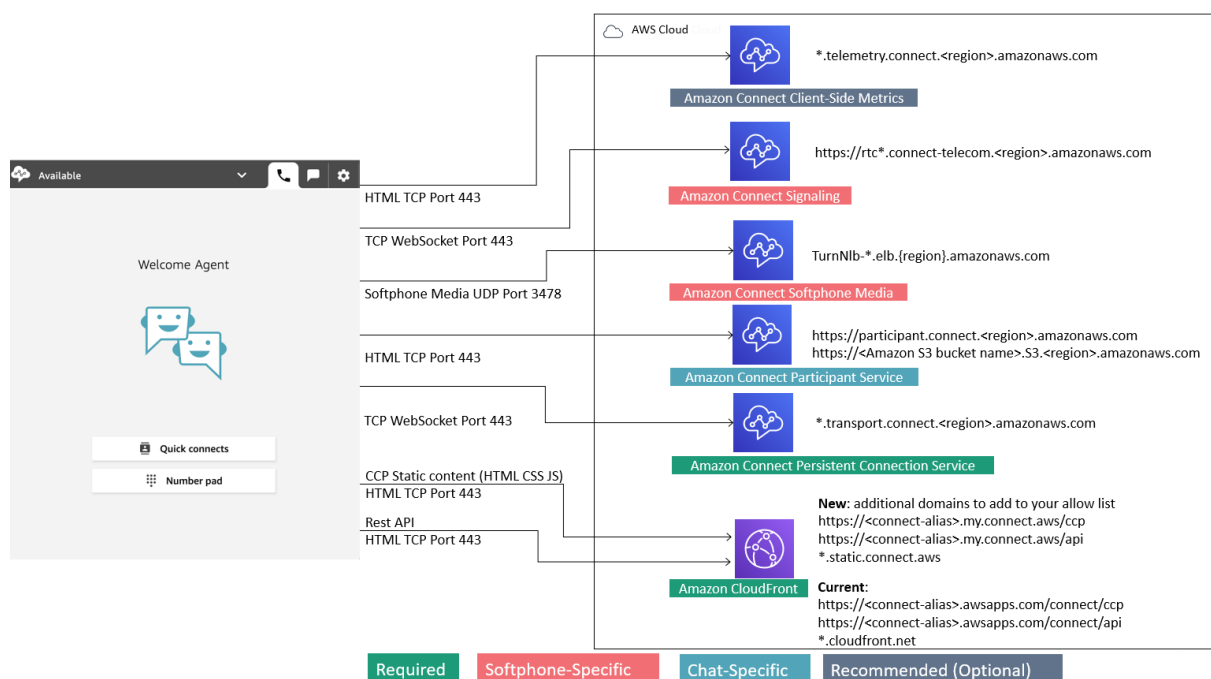
Son intégration : Un serveur VoIP peut être intégré ou connecté à d'autres systèmes (Annuaire de contacts (AD), CRM) vous permettant d'automatiser certaines tâches et de faciliter la communication avec vos clients.

Identifiez des sites marchands ou de service dont customer service implique des services VoIP, donnez quelques exemples et décrivez une architecture possible de leur système.

Amazon:

Utilisez Amazon Connect, un service VoIP cloud, pour gérer les appels clients, les chats et les e-mails. Cela permet aux agents du service client de gérer les interactions client depuis une seule interface et d'accéder aux informations client à partir de plusieurs systèmes.

Source : [Set up your network - AWS Direct Connect for Amazon Connect](#)



Serveur VoIP cloud: Amazon Connect est hébergé dans le cloud AWS, ce qui signifie qu'il n'y a pas de matériel sur site à installer ou à gérer.

Téléphones IP: Amazon propose une variété de téléphones IP d'entreprises partenaires, tels que Poly et Yealink.

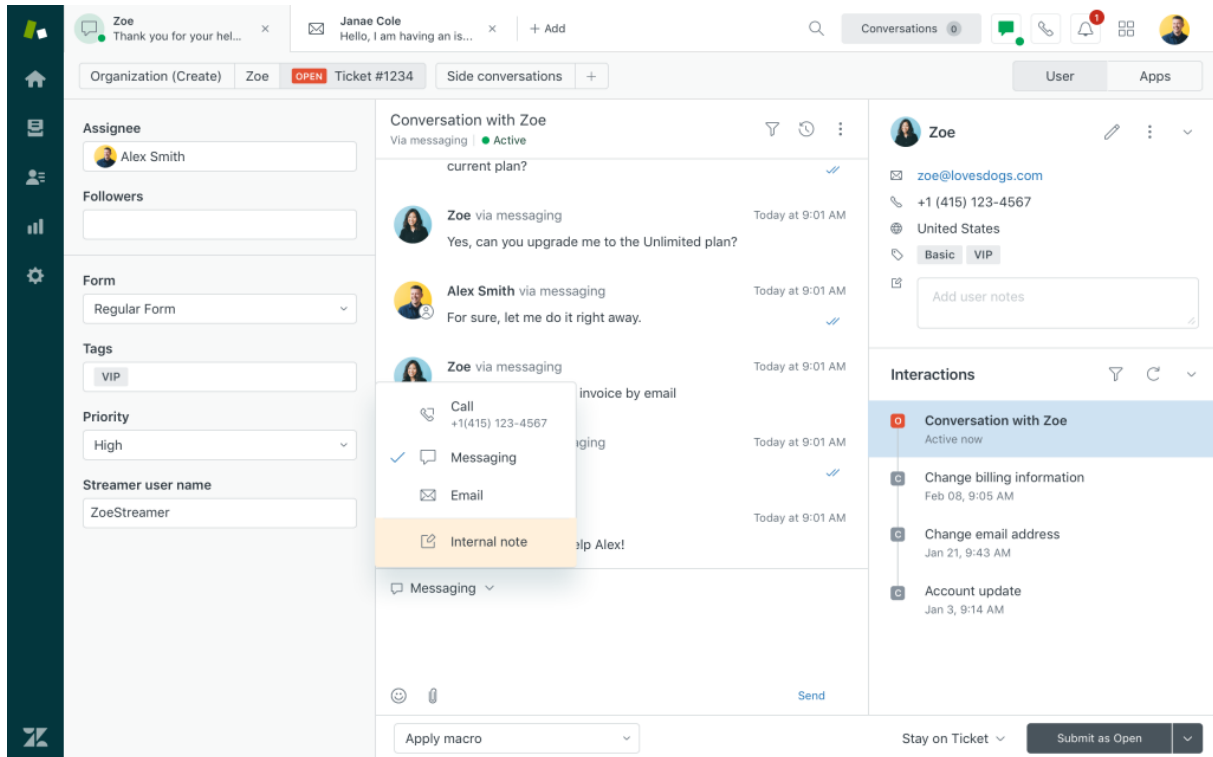
Logiciel de centre d'appels: Amazon Connect propose un logiciel de centre d'appels intégré qui comprend des fonctionnalités telles que l'acheminement des appels, la gestion des files d'attente et la création de rapports.

Intégrations CRM: Amazon Connect s'intègre à une variété de CRM, tels que Salesforce et Microsoft Dynamics 365.

GoDaddy:

Utilise Zendesk Talk, une solution de téléphonie VoIP basée sur le cloud, pour fournir une assistance téléphonique à ses clients. Zendesk Talk s'intègre à d'autres produits Zendesk, tels que le système de billetterie et le CRM, ce qui permet aux agents d'avoir une vue complète de l'historique et des interactions des clients.

Source : [Zendesk Talk: Overview of Pricing Plans and Key Features \(getvoip.com\)](https://getvoip.com/Zendesk-Talk-Overview-of-Pricing-Plans-and-Key-Features/)



Serveur VoIP cloud: Zendesk Talk est également hébergé dans le cloud, ce qui élimine le besoin de matériel sur site.

Téléphones IP: Zendesk Talk prend en charge une variété de téléphones IP, y compris les modèles de marques telles que Cisco, Avaya et Yealink.

Logiciel de centre d'appels: Zendesk Talk propose un logiciel de centre d'appels intégré avec des fonctionnalités telles que le routage des appels, la gestion des files d'attente et la création de rapports.

Intégrations CRM: Zendesk Talk s'intègre nativement au CRM Zendesk et à d'autres CRM populaires tels que Salesforce et HubSpot.

Shopify:

Utilise Shopify Ping, un service de messagerie vocale et vidéo intégré à sa plateforme de e-commerce. Shopify Ping permet aux marchands de communiquer directement avec leurs clients par téléphone ou vidéo, ce qui peut aider à résoudre les problèmes et à augmenter les conversions.

Serveur VoIP cloud: Shopify Ping est hébergé dans le cloud Shopify, ce qui signifie qu'il est intégré à la plateforme de e-commerce Shopify.

Application mobile: Shopify Ping n'utilise pas de téléphones IP traditionnels. Au lieu de cela, les agents et les clients utilisent une application mobile pour passer et recevoir des appels vocaux et vidéo.

Logiciel de centre d'appels: Shopify Ping n'inclut pas de logiciel de centre d'appels intégré, mais il peut être intégré à d'autres logiciels de centre d'appels tiers.

Intégrations CRM: Shopify Ping s'intègre nativement au CRM Shopify et peut être intégré à d'autres CRM populaires.

Sa scalabilité : Les serveurs VoIP sont flexibles et peuvent être mis à échelle facilement. Un bon administrateur VoIP peut augmenter ou diminuer le nombre d'utilisateurs, augmenter les capacités d'appels simultanées sans interruption de service. Une équipe peut aussi cloner reproduire une architecture locale et la déployer sur d'autres sites...

La scalabilité des serveurs VoIP est une caractéristique essentielle qui permet aux entreprises de s'adapter facilement à leurs besoins changeants en communication. Voici quelques points développant cette caractéristique :

1. Ajout ou suppression d'utilisateurs : Un administrateur VoIP compétent peut facilement ajuster le nombre d'utilisateurs du système en ajoutant de nouveaux utilisateurs ou en supprimant ceux qui ne sont plus nécessaires. Cela permet à l'entreprise de maintenir un système de communication adapté à sa taille et à ses besoins actuels.

2. Augmentation des capacités d'appels simultanés : Les capacités d'appels simultanés d'un serveur VoIP peuvent être augmentées sans interruption de service. Cela signifie que l'entreprise peut répondre à des pics de demandes sans compromettre la qualité des appels ou l'expérience utilisateur.

3. Déploiement sur plusieurs sites : Une équipe peut cloner et reproduire une architecture VoIP locale et la déployer sur d'autres sites de l'entreprise sans difficulté majeure. Cela permet une expansion géographique fluide de l'infrastructure de communication, ce qui est particulièrement utile pour les entreprises disposant de plusieurs bureaux ou sites.

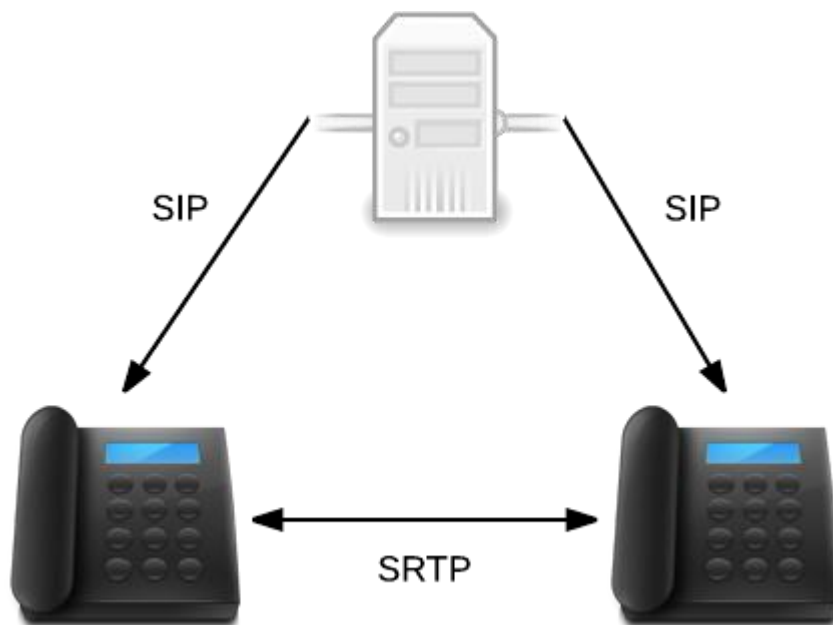
4. Flexibilité dans l'ajout de nouvelles fonctionnalités : En plus d'augmenter le nombre d'utilisateurs et de capacités d'appels, les serveurs VoIP peuvent être adaptés pour intégrer de nouvelles fonctionnalités en fonction des besoins de l'entreprise. Cela permet à l'entreprise de rester agile et de répondre rapidement aux évolutions du marché.

La scalabilité des serveurs VoIP offre aux entreprises la flexibilité nécessaire pour ajuster leur système de communication en fonction de leur croissance et de leurs besoins changeants, tout en garantissant une continuité de service et une évolutivité sans interruption.

Sa sécurité : Il est possible de configurer le chiffrement pour protéger les informations d'authentification et les appels vocaux. Il est donc plus sécurisé qu'un système téléphonique traditionnel. **Effectuez quelques recherches sur les chiffrements les mieux adaptés à la VoIP.**

Les protocoles de chiffrement les plus adaptés à la VoIP :

SRTP (Secure Real-time Transport Protocol) : Il s'agit du protocole de chiffrement standard pour les communications VoIP. SRTP assure la confidentialité et l'intégrité des appels en chiffrant les flux audio et en authentifiant les participants.



- **RTP** est responsable du transport des données en temps réel, mais ne sécurise pas leur contenu.
- **SRTP** sécurise le contenu des transmissions RTP en ajoutant du chiffrement et de l'authentification.

Projet

On commence par l'installation de la machine virtuelle debian en graphique pour plus de confort, on fait un **apt update** && **apt upgrade**.

Puis on télécharge les paquets asterisk en récupérant le lien sur le site asterisk

Pour avoir les options ifconfig etc il suffit de télécharger le paquet suivant `apt install net-tools`

```
← https://downloads.asterisk.org/pub/telephony/asterisk/
Index of /pub/telephony/asterisk/

../
misc/                                04-Nov-2013 19:45      -
old-releases/                       18-Mar-2024 13:25      -
pending/                             18-Mar-2024 13:25      -
releases/                            29-Nov-2012 00:04      -
uebmim/                              18-Mar-2024 13:18     29985
ChangeLog-18-current.md             18-Mar-2024 13:18     29985
ChangeLog-18.22.0.md                18-Mar-2024 13:22     29980
ChangeLog-20-current.md             18-Mar-2024 13:22     29980
ChangeLog-20.7.0.md                 18-Mar-2024 13:25     31271
ChangeLog-21.2.0.md                 18-Mar-2024 13:25     31271
README-18-current.md                18-Mar-2024 13:18     11218
README-18.22.0.md                   18-Mar-2024 13:18     11218
README-20-current.md                18-Mar-2024 13:22     11218
README-20.7.0.md                    18-Mar-2024 13:22     11218
README-21-current.md                18-Mar-2024 13:25     11218
README-21.2.0.md                    18-Mar-2024 13:25     11218
asterisk-18-current-patch.md5        18-Mar-2024 13:18        64
asterisk-18-current-patch.sha1       18-Mar-2024 13:18        72
asterisk-18-current-patch.sha256     18-Mar-2024 13:18        96
asterisk-18-current-patch.tar.gz     18-Mar-2024 13:18    298924
asterisk-18-current-patch.tar.gz.asc 18-Mar-2024 13:18       833
asterisk-18-current.md5              18-Mar-2024 13:18        58
asterisk-18-current.sha1              18-Mar-2024 13:18        66
asterisk-18-current.sha256           18-Mar-2024 13:18        90
asterisk-18-current.tar.gz            18-Mar-2024 13:18   28489141
asterisk-18-current.tar.gz.asc        18-Mar-2024 13:18       833
asterisk-18.22.0-patch.md5           18-Mar-2024 13:18        64
asterisk-18.22.0-patch.sha1          18-Mar-2024 13:18        72
asterisk-18.22.0-patch.sha256        18-Mar-2024 13:18        96
asterisk-18.22.0-patch.tar.gz         18-Mar-2024 13:18    298924
asterisk-18.22.0-patch.tar.gz.asc     18-Mar-2024 13:18       833
asterisk-18.22.0.md5                  18-Mar-2024 13:18        58
asterisk-18.22.0.sha1                  18-Mar-2024 13:18        66
asterisk-18.22.0.sha256               18-Mar-2024 13:18        90
asterisk-18.22.0.tar.gz               18-Mar-2024 13:18   28489141
asterisk-18.22.0.tar.gz.asc            18-Mar-2024 13:18       833
asterisk-20-current-patch.md5         18-Mar-2024 13:22        63
asterisk-20-current-patch.sha1        18-Mar-2024 13:22        71
asterisk-20-current-patch.sha256      18-Mar-2024 13:22        95
asterisk-20-current-patch.tar.gz      18-Mar-2024 13:22   298656
asterisk-20-current-patch.tar.gz.asc  18-Mar-2024 13:22       833

Paramétrage de libitm1:amd64 (12.2.0-14) ...
Paramétrage de libc-devtools (2.36-9+deb12u6) ...
Paramétrage de libctf0:amd64 (2.40-2) ...
Paramétrage de libgprofng0:amd64 (2.40-2) ...
Paramétrage de libgcc-12-dev:amd64 (12.2.0-14) ...
Paramétrage de libc6-dev:amd64 (2.36-9+deb12u6) ...
Paramétrage de binutils-x86-64-linux-gnu (2.40-2) ...
Paramétrage de binutils (2.40-2) ...
Paramétrage de gcc-12 (12.2.0-14) ...
Paramétrage de linux-compiler-gcc-12-x86 (6.1.85-1) ...
Paramétrage de linux-headers-6.1.0-20-amd64 (6.1.85-1) ...
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u6) .
..
root@debianVOIP:/usr/src# ls
linux-headers-6.1.0-20-amd64  linux-headers-6.1.0-20-common  linux-kbuild-6.1
root@debianVOIP:/usr/src# ls -l
total 8
drwxr-xr-x 4 root root 4096 23 avril 11:42 linux-headers-6.1.0-20-amd64
drwxr-xr-x 4 root root 4096 23 avril 11:42 linux-headers-6.1.0-20-common
lrwxrwxrwx 1 root root 23 11 avril 23:17 linux-kbuild-6.1 -> ../lib/linux-kbuild-6.1
root@debianVOIP:/usr/src# wget https://downloads.asterisk.org/pub/telephony/asterisk/asterisk-21.2.0.tar.gz
```

Une fois le paquet asterisk installé, on fait de même pour le paquet dahdi-linux-

complete On prend le fichier current sur le site et on tape la commande suivante:

```
root@debianVOIP:/usr/src# ls -l
total 25704
-rw-r--r-- 1 root root 26312154 18 mars 14:25 asterisk-21.2.0.tar.gz
drwxr-xr-x 4 root root 4096 23 avril 11:42 linux-headers-6.1.0-20-amd64
drwxr-xr-x 4 root root 4096 23 avril 11:42 linux-headers-6.1.0-20-common
lrwxrwxrwx 1 root root 23 11 avril 23:17 linux-kbuild-6.1 -> ../lib/linux-kb
uild-6.1
root@debianVOIP:/usr/src# wget https://downloads.asterisk.org/pub/telephony/dahdi-
linux-complete/dahdi-linux-complete-current.tar.gz
```

Après on récupère le paquet libpri en current.tar.gz également

Index of /pub/telephony/libpri/

../	16-Aug-2023 15:52	-
old/	16-Aug-2023 15:52	-
pending/	16-Aug-2023 15:52	-
releases/	16-Aug-2023 15:52	-
ChangeLog-1-current.md	16-Aug-2023 15:52	2239
ChangeLog-1.6.1.md	16-Aug-2023 15:52	2239
README-1-current.md	16-Aug-2023 15:52	1694
README-1.6.1.md	16-Aug-2023 15:52	1694
libpri-1-current-patch.md5	16-Aug-2023 15:52	60
libpri-1-current-patch.sha1	16-Aug-2023 15:52	68
libpri-1-current-patch.sha256	16-Aug-2023 15:52	92
libpri-1-current-patch.tar.gz	16-Aug-2023 15:52	5575
libpri-1-current-patch.tar.gz.asc	16-Aug-2023 15:52	833
libpri-1-current.md5	16-Aug-2023 15:52	54
libpri-1-current.sha1	16-Aug-2023 15:52	62
libpri-1-current.sha256	16-Aug-2023 15:52	86
libpri-1-current.tar.gz	16-Aug-2023 15:52	334119
libpri-1-current.tar.gz.asc	16-Aug-2023 15:52	833
libpri-1.6.1-patch.md5	16-Aug-2023 15:52	60
libpri-1.6.1-patch.sha1	16-Aug-2023 15:52	68
libpri-1.6.1-patch.sha256	16-Aug-2023 15:52	92
libpri-1.6.1-patch.tar.gz	16-Aug-2023 15:52	5575
libpri-1.6.1-patch.tar.gz.asc	16-Aug-2023 15:52	833
libpri-1.6.1.md5	16-Aug-2023 15:52	54
libpri-1.6.1.sha1	16-Aug-2023 15:52	62
libpri-1.6.1.sha256	16-Aug-2023 15:52	86
libpri-1.6.1.tar.gz	16-Aug-2023 15:52	334119
libpri-1.6.1.tar.gz.asc	16-Aug-2023 15:52	833

Puis on refait wget avec le lien récupéré

```
root@debianVOIP:/usr/src# ls -l
total 32868
-rw-r--r-- 1 root root 26312154 18 mars 14:25 asterisk-21.2.0.tar.gz
-rw-r--r-- 1 root root 7333659 17 avril 09:01 dahdi-linux-complete-current.tar.gz
drwxr-xr-x 4 root root 4096 23 avril 11:42 linux-headers-6.1.0-20-amd64
drwxr-xr-x 4 root root 4096 23 avril 11:42 linux-headers-6.1.0-20-common
lrwxrwxrwx 1 root root 23 11 avril 23:17 linux-kbuild-6.1 -> ../lib/linux-kb
uild-6.1
root@debianVOIP:/usr/src# wget https://downloads.asterisk.org/pub/telephony/libpri
/libpri-1-current.tar.gz
```

Voici les 3 fichiers téléchargé

```
root@debianVOIP:/usr/src# ls -l
total 33196
-rw-r--r-- 1 root root 26312154 18 mars 14:25 asterisk-21.2.0.tar.gz
-rw-r--r-- 1 root root 7333659 17 avril 09:01 dahdi-linux-complete-current.tar.gz
-rw-r--r-- 1 root root 334119 16 août 2023 libpri-1-current.tar.gz
```

Maintenant on les dezippé, on fait cette commande pour nos 3 fichiers, dahdi, libpri et Asterisk

```
root@debianVOIP:/usr/src# tar -xvzf asterisk-21.2.0.tar.gz
```

Installé les outils necessaire pour compiler nos fichiers pour se faire :

« **sudo apt install build-essential** » et les commande make seront disponible

La commande make est un utilitaire largement utilisé dans le développement de

logiciels, en particulier dans le contexte de la compilation de programmes à partir de leur

code source. Son principal objectif est d'automatiser le processus de compilation en utilisant des fichiers appelés makefiles; qui spécifient comment les différents fichiers source doivent être compilés et liés pour créer le programme final.

Voici quelques-unes des utilisations principales de make :

1. Compilation de logiciels: make est utilisé pour compiler des programmes à partir de leur code source. Il peut prendre en charge la compilation de projets complexes avec de nombreux fichiers source et dépendances.

2. Gestion de projets: En utilisant des makefiles, make peut gérer les dépendances entre les fichiers source et effectuer uniquement les tâches nécessaires pour reconstruire les parties modifiées du projet. Cela permet d'économiser du temps en évitant de recompiler les parties du projet qui n'ont pas été modifiées.

3. Portabilité: Les makefiles peuvent être écrits pour fonctionner sur différents systèmes d'exploitation et architectures, ce qui facilite la portabilité des projets logiciels.

4. Automatisation: make peut être utilisé pour automatiser diverses tâches liées au développement logiciel, telles que la génération de documentation, l'exécution de tests, la gestion des versions, etc.

En résumé, make est un outil puissant et polyvalent utilisé par les développeurs pour gérer efficacement le processus de compilation et d'automatisation dans le développement de logiciels.

```
root@debianVOIP:/usr/src/dahdi-linux-complete-3.4.0-rc1+3.4.0-rc1# make
```

On fait un make install

Pareil pour notre fichier libpri

Après on se rend sur notre fichier asterisk

```
root@debianVOIP:/usr/src/asterisk-21.2.0/contrib/scripts#
```

Puis on execute le fichier install prereq install

```
root@debianVOIP:/usr/src/asterisk-21.2.0/contrib/scripts# ./install_prereq install
```

Une fois l'installations termin  . On retourne au debut du fichiers asterisk et on fait la commande suivante

```
root@debianVOIP:/usr/src/asterisk-21.2.0# ./configure
```

Si le logo apparait c'est que asterisk est bien configur  

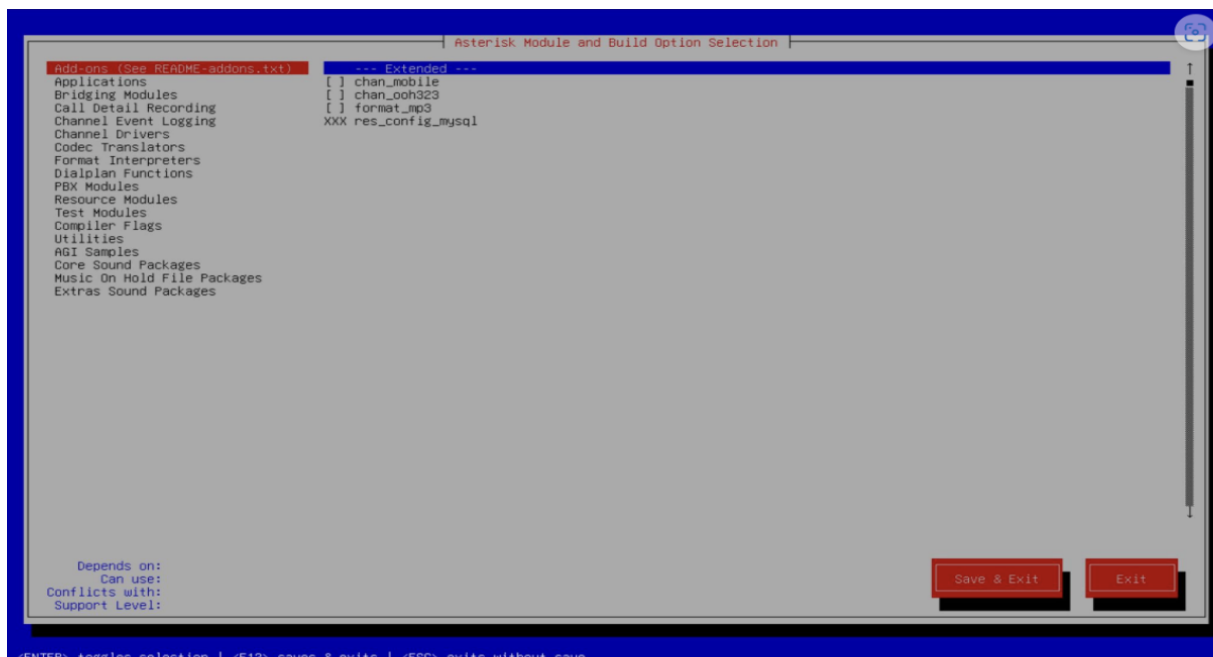
```
:checking for GTK2... no
:configure: creating ./config.status
:config.status: creating makeopts
:config.status: creating autoconfig.h
:configure: Menuselect build configuration successfully completed
```

```

$$$$$$$$$$$$$$$$$=..
. $7$7..          .7$7:..
.$$.             .,$7.7
. $7.            7$$$$
..$.             $$$$$
..7$ .?.        $$$$$ .?.    7$$$$.
$. $.          $$$7. $$$7 .7$$$ .$$$
.777.          $$$$$77$$$$77$$$$7.  $$$
$$$~           7$$$$$$$$$$$$7.      $$$
. $7           7$$$$$$$$7:          ?$$$
$$$           ?7$$$$$$$$$I          .$$$7
$$$           .7$$$$$$$$$$$$$$$$$ :$$$
$$$           $$$$$7$$$$$$$$$$$$$ .$$$
$$$           $$$ 7$$$7 .$$$ .$$$
$$$$          $$$7          .$$$
7$$$7         7$$$         7$$$
$$$$$         $$$         $$$
$$$$7.        $$ (TM)
$$$$$.        .7$$$$$ $$
$$$$$$$$7$$$$7$$$$$$$$. $$$$$$
$$$$$$$$$$$$$.
```

```
: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@debianVOIP:/usr/src/asterisk-21.2.0#
```

On tape la commande **make menuselect** pour ouvrir la fenêtre



On sauvegarde et quitte la fenetre puis on fait les commande suivante

```
Generating input for menuselect ...
CC="cc" CXX="g++" LD="" AR="" RANLIB="" CFLAGS="" LDFLAGS="" make -C menuselect CONFIGURE_SILENT="--silent" menuselect
make[2] : on entre dans le répertoire « /usr/src/asterisk-21.2.0/menuselect »
gcc -g -D_GNU_SOURCE -Wall -Wno-deprecated-declarations -DHAVE_NCURSES -c -o menuselect_stub.o menuselect_stub.c
gcc -o menuselect menuselect.o strcompat.o menuselect_stub.o -lxml2
make[2] : on quitte le répertoire « /usr/src/asterisk-21.2.0/menuselect »
menuselect/menuselect --check-deps menuselect.makeopts
menuselect/menuselect --check-deps menuselect.makeopts
menuselect changes saved!
make[1] : on quitte le répertoire « /usr/src/asterisk-21.2.0 »
root@debianVOIP:/usr/src/asterisk-21.2.0# make && make install
```

```
2024-04-23 13:10:21 [2,33 MB/s] - « asterisk-core-sounds-fi-gsm-1.0.1.taz.gz » sauvegardé [2519983/2519983]

make[1] : on quitte le répertoire « /usr/src/asterisk-21.2.0/sounds »
find rest-api -name "*.json" | while read x; do \
    /usr/bin/install -c -m 044 $x "/var/lib/asterisk/rest-api" ; \
done
+----- 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
+
+
+----- QT -----+
+
+ 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
+
+-----+
root@debianVOIP:/usr/src/asterisk-21.2.0#
```

On fait la commande make simples

La commande ``make samples`` dans Asterisk est essentielle pour compiler et installer les exemples de configuration et de scripts inclus avec le logiciel. Cela permet aux utilisateurs de comprendre et de configurer Asterisk plus facilement en leur fournissant des modèles pratiques et des guides de configuration pour divers scénarios Téléphoniques.

Puis un `make config`

La commande ``make config`` dans Asterisk génère les fichiers de configuration réels à partir des modèles fournis dans le répertoire de configuration de base, facilitant ainsi la personnalisation et le déploiement rapide d'un système de téléphonie.

On termine avec la commande : **`make install -logrotate`**

La commande ``make install-logrotate`` dans Asterisk installe un fichier de configuration spécifique pour logrotate, facilitant la gestion efficace des journaux d'Asterisk. Voici les points clés à retenir :

1. `Gestion des Journaux**`** : Elle permet de gérer automatiquement les journaux D'Asterisk en définissant les règles de rotation, d'archivage et de compression des fichiers journaux.

2. `Installation Automatique**`** : Elle installe le fichier de configuration logrotate spécifique à Asterisk, définissant les paramètres de gestion des journaux pour assurer un fonctionnement efficace du système.

3. `Prévention de la Croissance Incontrôlée des Journaux**`** : Elle aide à éviter que les fichiers journaux ne deviennent trop volumineux en les rotant périodiquement, ce qui permet de conserver un historique des événements tout en limitant l'espace disque utilisé.

En somme, ``make install-logrotate`` simplifie la gestion des journaux d'Asterisk en

automatisant le processus de rotation et d'archivage, assurant ainsi une surveillance efficace du système de téléphonie.

On installe le paquet net-tools qui est un ensemble d'outils en ligne de commande permettant de gérer divers aspects du réseau sur les systèmes Linux. L'installation de la bibliothèque ; via la commande `sudo apt install net-tools` vous donne accès à plusieurs outils réseau utiles, notamment :

- 1. ifconfig** : Affiche et configure les interfaces réseau sur votre système, y compris les adresses IP, les masques de sous-réseau et les informations sur le trafic réseau.
- 2. netstat** : Affiche les connexions réseau actives, les tables de routage, les statistiques d'interface et d'autres informations sur les connexions réseau en cours.
- 3. arp** : Affiche et modifie la table ARP (Address Resolution Protocol) du système, qui associe les adresses IP aux adresses MAC des périphériques réseau.
- 4. route** : Affiche et manipule la table de routage du système, qui détermine la manière dont les paquets réseau sont acheminés à travers le réseau.
- 5. iptunnel** : Permet de créer et de gérer des tunnels réseau, notamment des tunnels IP (IPv4 et IPv6) et des tunnels Ethernet.

Ces outils sont essentiels pour diagnostiquer, configurer et gérer les paramètres réseau sur les systèmes Linux. Ils sont largement utilisés par les administrateurs système pour surveiller et dépanner les réseaux.

Création des utilisateurs et des règles associées

Pour la création des utilisateurs nous avons besoins de nous rendre dans le fichier **pjsip.conf**, pour cela nous mettrons le chemin suivant **sudo nano /etc/asterisk/pjsip.conf** :

Configuration des comptes utilisateurs

```
GNU nano 7.2                                pjsip.conf
[general]
tlsenable=yes
tlsbindaddr=0.0.0.0:5060
tlscertfile=/etc/asterisk/keys/asterisk.pem
tlscapfile=/etc/asterisk/keys/ca.crt

[transport-udp]
type=transport
protocol=udp
bind=0.0.0.0

[transport-tls]
type=transport
protocol=tls
bind=0.0.0.0:5061
cert_file=/etc/asterisk/keys/asterisk.crt
priv_key_file=/etc/asterisk/keys/asterisk.key
method=sslv23
```

```
GNU nano 7.2                                pjsip.conf

[endpoint_internal]
type=endpoint
context=from-internal
disallow=all
allow=ulaw
language=fr

[auth_userpass]
type=auth
auth_type=userpass
transport=tls

[aor_dynamic]
type=aor
max_contacts=1
```

```
GNU nano 7.2                                pjsip.conf

[alice](endpoint_internal)
auth=alice
aors=alice
[alice](auth_userpass)
password=bonjour ; put strong passwd
username=alice

[bob](endpoint_internal)
auth=bob
aors=bob
[bob](auth_userpass)
password=bonjour ; put strong passwd
username=bob

[martin](endpoint_internal)
auth=martin
aors=martin
[martin](auth_userpass)
password=bonjour ; put strong passwd
username=martin
```

Définition des règles de routage des appels

```

GNU nano 7.2                                extensions.conf *
[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

; Règles pour acheminer les appels vers la messagerie vocale
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

```

Les ports utilisés dans le cadre du protocole SIP :

De la même manière que le célèbre port 80 a été attribué aux services web par la très sérieuse Internet Assigned Numbers Authority (IANA), le protocole SIP s'est vu officiellement attribuer les ports 5060 et 5061 par cette même organisation. Le fonctionnement du SIP repose sur l'échange de messages textes qui sont **envoyés en clair sur le port 5060**, et **en mode sécurisé sur le port 5061**. Notez que les protocoles de transmission de base d'Internet ne supportent qu'un nombre limité de ports, chacun pouvant être attribué à un service.

Mise en place du protocole TLS :

Le protocole TLS est capable de définir les clés de session correspondantes sur un canal non chiffré grâce à une technologie connue sous le nom de cryptographie à clé publique

Créer le répertoire "keys"

- `cd /etc/asterisk/keys`
-
- Mettre "ast_tls_cert" dans le répertoire keys (lien github pour le télécharger : https://github.com/asterisk/asterisk/blob/master/contrib/scripts/ast_tls_cert)
- Ce mettre en root et faire "chmod +x ast_tls_cert" dans le répertoire (si on a pas les droits)
- Faire `./ast_tls_cert -C 192.168.203.130 -O "SecurityTLS" -d /etc/asterisk/keys -b 2048`

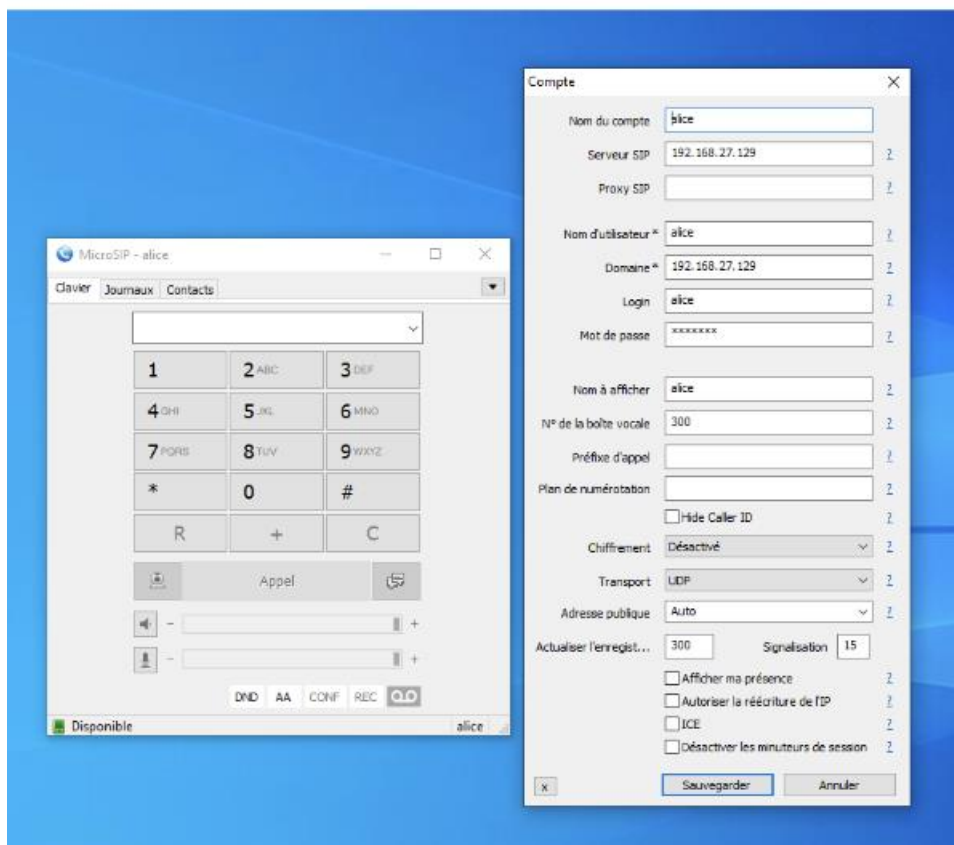
```
root@debian:/etc/asterisk/keys# chmod +x ast_tls_cert
root@debian:/etc/asterisk/keys# ./ast_tls_cert -C 192.168.203.130 -O "SecurityTLS" -d /etc/asterisk/keys
-b 2048

No config file specified, creating '/etc/asterisk/keys/tmp.cfg'
You can use this config file to create additional certs without
re-entering the information for the fields in the certificate
Creating CA key /etc/asterisk/keys/ca.key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
Creating CA certificate /etc/asterisk/keys/ca.crt
Enter pass phrase for /etc/asterisk/keys/ca.key:
Creating certificate /etc/asterisk/keys/asterisk.key
Creating signing request /etc/asterisk/keys/asterisk.csr
Creating certificate /etc/asterisk/keys/asterisk.crt
Certificate request self-signature ok
subject=CN = 192.168.203.130, O = SecurityTLS
Enter pass phrase for /etc/asterisk/keys/ca.key:
Combining key and crt into /etc/asterisk/keys/asterisk.pem
root@debian:/etc/asterisk/keys#
```

Mise en place des outils de communication

Installer un softphone comme microsip et mizudroid pour le téléphone.

Voici la configuration quand on ajoute un compte sur microsip :



Pour créer un pont entre votre machine virtuelle et votre appareil mobile, ajoutez une deuxième carte réseau à votre machine virtuelle et configurez-la en mode "Bridge" dans les paramètres de virtualisation. Redémarrez ensuite votre machine virtuelle et configurez la nouvelle carte réseau pour utiliser le même réseau que votre appareil mobile en modifiant le fichier d'interfaces réseau sur la machine virtuelle. Une fois cela fait, votre appareil mobile et votre machine virtuelle seront connectés au même réseau local, leur permettant de communiquer directement entre eux.

Pour configurer la seconde carte réseau , il suffit de se rendre dans le fichier network/interfaces pour déclarer notre second réseau

```

root@debianVOIP:/# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

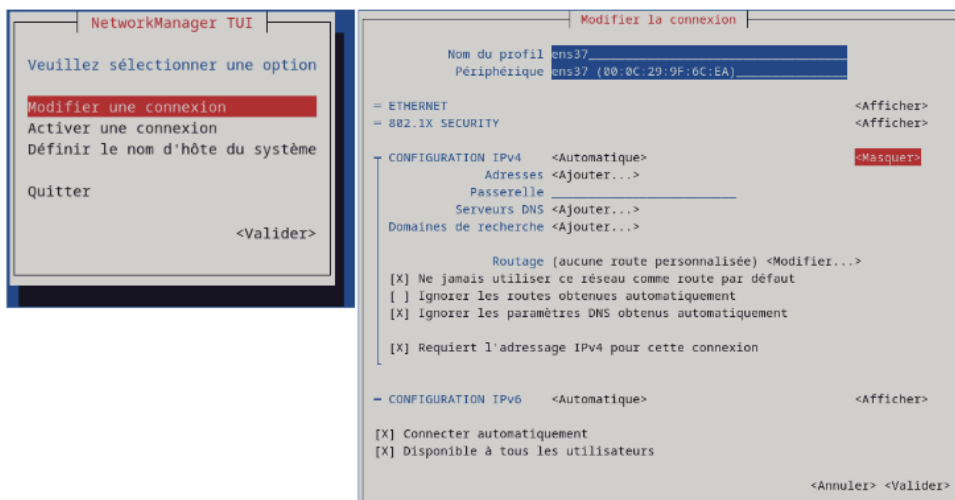
# The loopback network interface
auto lo
iface lo inet loopback

allow-hotplug ens33
iface ens33 inet static
    address 192.168.27.129/24
    netmask 255.255.255.0
    gateway 192.168.27.2
    broadcast 192.168.27.255
    dns-nameservers 192.168.27.2

allow-hotplug ens37
iface ens37 inet dhcp

```

Pour accéder au network manager : on tape nmtui



Avant sa on récupère avec ip addr notre adresse mac

Puis on va attribué une adresse static a notre ens33 qui correspond a notre nat

Modifier la connexion

Nom du profil: ens33
Périphérique: ens33 (00:0C:29:9F:6C:E0)

= ETHERNET <Afficher>
= 802.1X SECURITY <Afficher>

CONFIGURATION IPv4 <Manuel> <Masquer>

Adresses: 192.168.27.129/24 <Supprimer> <Ajouter...>
Passerelle: 192.168.27.2
Serveurs DNS: 192.168.27.2 <Supprimer> <Ajouter...>
Domaines de recherche: <Ajouter...>

Routage (aucune route personnalisée) <Modifier...>
[] Ne jamais utiliser ce réseau comme route par défaut
[] Ignorer les routes obtenues automatiquement
[] Ignorer les paramètres DNS obtenus automatiquement
☒ Requiert l'adressage IPv4 pour cette connexion

CONFIGURATION IPv6 <Automatique> <Afficher>

☒ Connecter automatiquement
☒ Disponible à tous les utilisateurs

<Annuler> <Valider>

On redémarre la machine et ifconfig pour vérifier ou ip a

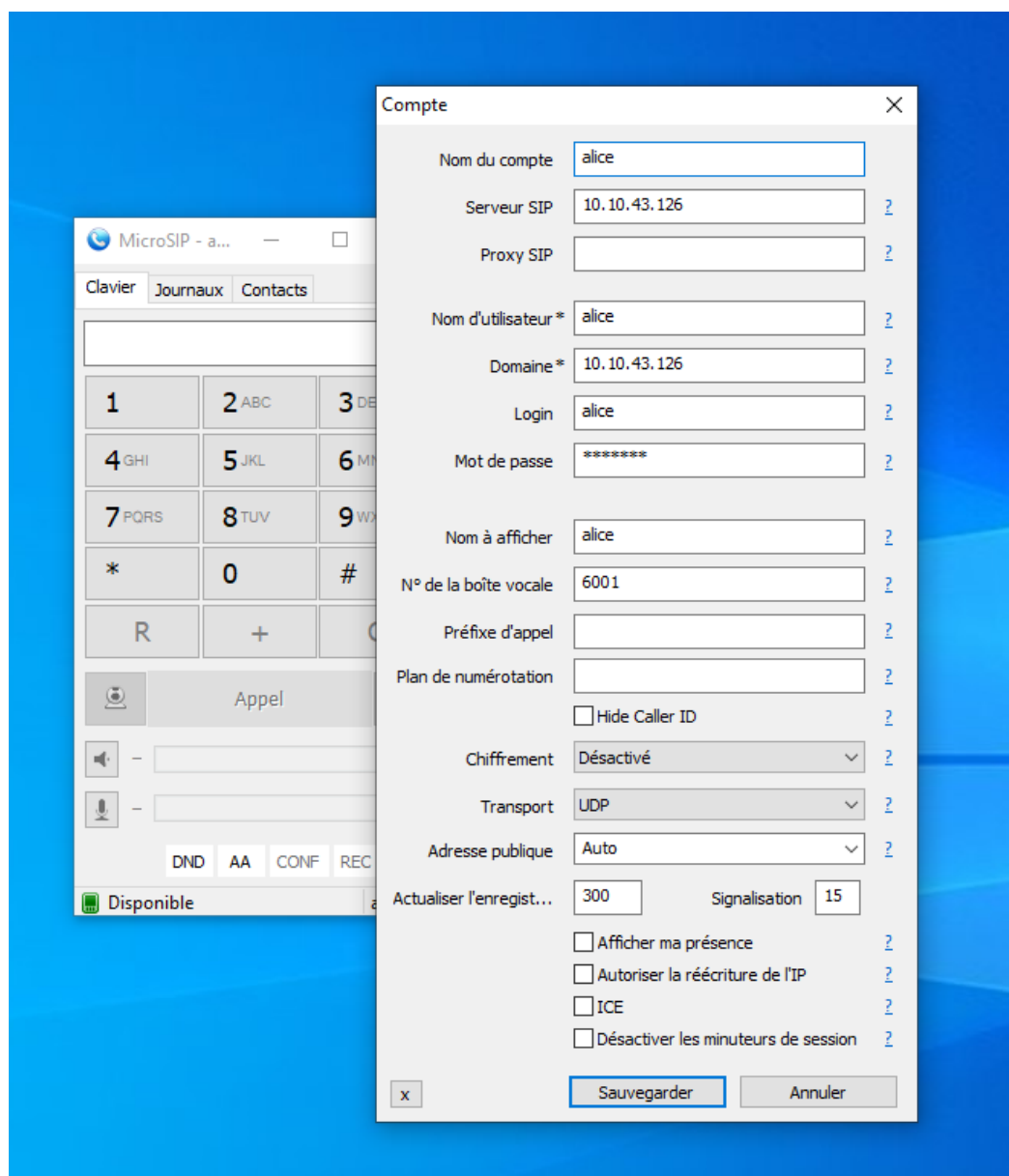
```
root@debianVOIP:/# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.27.129 netmask 255.255.255.0 broadcast 192.168.27.255
    inet6 fe80::20c:29ff:fe9f:6ce0 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:9f:6c:e0 txqueuelen 1000 (Ethernet)
    RX packets 2528 bytes 284774 (278.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 609 bytes 124769 (121.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens37: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.2.141 netmask 255.255.0.0 broadcast 10.10.255.255
    inet6 fe80::20c:29ff:fe9f:6cea prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:9f:6c:ea txqueuelen 1000 (Ethernet)
    RX packets 714115 bytes 87074082 (83.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3350 bytes 596576 (582.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

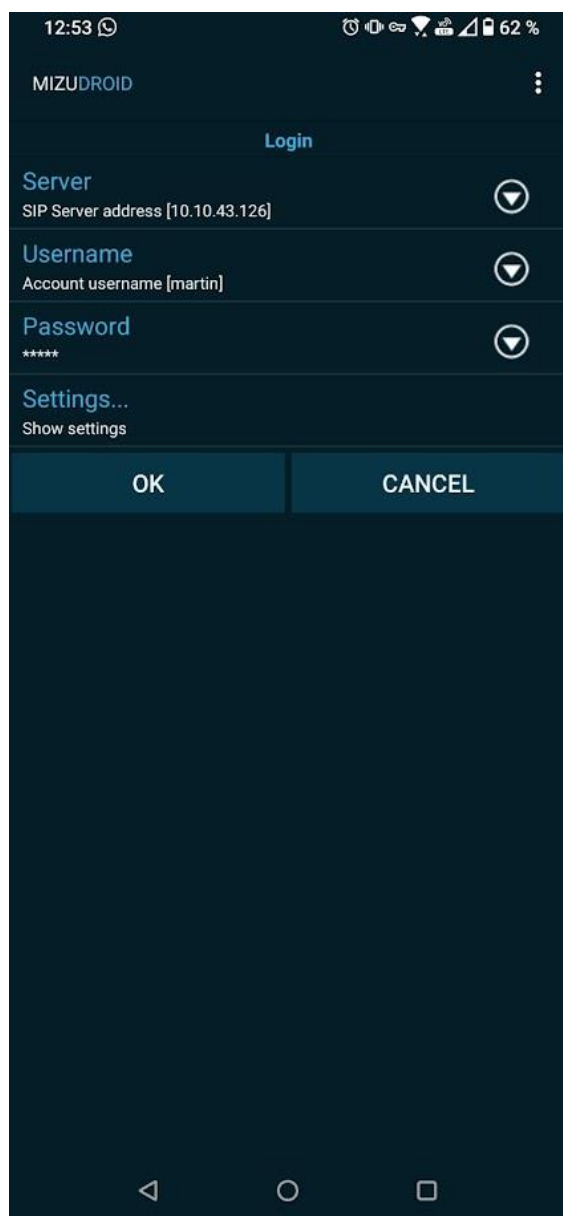
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 1243 bytes 178038 (173.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1243 bytes 178038 (173.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```


Test et validation

Configuration de l'utilisateur Alice sur l'outil MicroSIP



Configuration de l'utilisateur martin sur l'outil Mizudroid



The screenshot displays the Mizudroid application interface on an Android device. At the top, the status bar shows the time as 12:53 and various system icons. The app's title bar reads "MIZUDROID" with a menu icon on the right. Below this is a "Login" section. It contains three input fields: "Server" with the value "SIP Server address [10.10.43.126]", "Username" with the value "Account username [martin]", and "Password" with masked characters "*****". Each field has a dropdown arrow icon on its right. Below these fields is a "Settings..." option with the text "Show settings". At the bottom of the login section are two buttons: "OK" and "CANCEL". The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

12:53

MIZUDROID

Login

Server
SIP Server address [10.10.43.126]

Username
Account username [martin]

Password

Settings...
Show settings

OK CANCEL



MIZUDROID



Stage Djoanne (alice)

0:03

Accepté



alice
Stage Djoanne



MIZUDROID



Stage Djoanne (alice)



alice
Stage Djoanne



Historique des évènements :

```
=====
Current conditions: type core show license for asterisk.
=====
Connected to Asterisk 20.7.0 currently running on debianvoIP (pid = 17418)
-- Removed contact 'sip:alice@192.168.197.129:61310;ob' from AOR 'alice' due to request
== Contact alice/sip:alice@192.168.197.129:61310;ob has been deleted
== Endpoint alice is now Unreachable
-- Added contact 'sip:alice@192.168.197.129:61310;ob' to AOR 'alice' with expiration of 300 seconds
== Endpoint alice is now Reachable
-- Added contact 'sip:martin@10.10.43.113:12830' to AOR 'martin' with expiration of 90 seconds
== Endpoint martin is now Reachable
-- Executing [6003@from-internal:1] Dial("PJSIP/alice-00000006", "PJSIP/martin,10") in new stack
-- Called PJSIP/martin
-- PJSIP/martin-00000007 is ringing
> 0x7f8b5407ef20 -- Strict RTP learning after remote address set to: 10.10.43.113:15198
> 0x7f8b5407ef20 -- Strict RTP switching to RTP target address 10.10.43.113:15198 as source
-- PJSIP/martin-00000007 answered PJSIP/alice-00000006
> 0x7f8b5406e900 -- Strict RTP learning after remote address set to: 192.168.197.129:4000
-- Channel PJSIP/martin-00000007 joined 'simple_bridge' basic-bridge <d309015f-e007-4845-b404-15b7c5ae9f4d>
-- Channel PJSIP/alice-00000006 joined 'simple_bridge' basic-bridge <d309015f-e007-4845-b404-15b7c5ae9f4d>
> Bridge d309015f-e007-4845-b404-15b7c5ae9f4d: switching from simple_bridge technology to native_rtp
> Remotely bridged 'PJSIP/alice-00000006' and 'PJSIP/martin-00000007' - media will flow directly between them
> 0x7f8b5406e900 -- Strict RTP qualifying stream type: audio
> 0x7f8b5406e900 -- Strict RTP switching source address to 10.10.42.178:57173
> 0x7f8b5407ef20 -- Strict RTP learning complete - Locking on source address 10.10.43.113:15198
> 0x7f8b5406e900 -- Strict RTP learning complete - Locking on source address 10.10.42.178:57173
-- Channel PJSIP/alice-00000006 left 'native_rtp' basic-bridge <d309015f-e007-4845-b404-15b7c5ae9f4d>
-- Channel PJSIP/martin-00000007 left 'native_rtp' basic-bridge <d309015f-e007-4845-b404-15b7c5ae9f4d>
== Spawn extension (from-internal, 6003, 1) exited non-zero on 'PJSIP/alice-00000006'
-- Removed contact 'sip:martin@10.10.43.113:12830' from AOR 'martin' due to request
== Contact martin/sip:martin@10.10.43.113:12830 has been deleted
== Endpoint martin is now Unreachable
debianvoIP*CLI> █
```

Automatisation

L'installation d'un IVR nécessite de configurer le fichier "extensions.conf"

Nous avons deux choix, soit utiliser ce que l'on appelle des scripts TTS, donc de la synthèse de voix ou bien d'utiliser des audios réalisés par nous-même.

La seconde option est la plus conseillée car elle permet d'avoir une meilleure stabilité de l'IVR. En effet, elle aura pour charge d'activer seulement l'audio lors de l'appel et non un script complexe basé sur de la synthèse de voix. De plus, nous aurons un meilleur rendu et une meilleure fluidité.

Le répertoire audio de base étant rempli de son, nous allons vite nous perdre si nous stockons nos propres audios dans le même dossier. La première chose à faire est donc de créer le répertoire dans laquelle nous allons stocker nos audios.

Ci-dessous nous avons donc créé un répertoire nommé "custom" qui lui se trouve dans le répertoire source sur lequel Asterisk va puiser les audios, on peut déjà voir nos audios personnels.

```
laplateforme@debianvoIP: /var/lib/asterisk/sounds/fr/custom$ ls
accueil.ulaw  service_accueil.ulaw  service_client_priv.ulaw
accueil.wav   service_accueil.wav
```

Pour réaliser nos audios on va aller sur ce site : <https://speechgen.io/fr/>. Cela nous permet de

transformer un texte en voix.

Il faut ensuite télécharger le fichier en ".wav" et le stocker dans le dossier custom.

Asterisk prenant en charge les fichiers audios de type "ulaw", nous allons ensuite taper la commande comme suit afin de convertir notre audio avec sox :

D'abord installer le paquet installer le paquet "sox"

```
sudo apt install sox
```

puis convertir

```
sox -V accueil.wav -r 8000 -c 1 -t ul accueil.ulaw
```

Je configure ensuite mon fichier "extension.conf" pour réaliser le plan de dialogue :

Dans mon contexte [from-internal] j'ajoute l'extension 7000 qui sera le numéro pour l'IVR avec une priorité "1".

J'y ajoute aussi une commande "Background" qui me permet de jouer l'audio "pls-wait-connect-call" qui lui est natif dans asterisk.

Juste en dessous se trouve une seconde instruction de l'extension 7000. Celle-ci sera activé lorsque la première sera terminée. Elle permet de faire une redirection - Goto - vers un nouveau context qui est [IVR] avec l'extension "s" et de priorité "1".

```
GNU nano 7.2 extensions.conf
[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,Background(pls-wait-connect-call)
exten => 7000,n,Goto(IVR,s,1)

; 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
```

Dans mon contexte [IVR], j'y ajoute plusieurs instructions.

La première est notre première extension "s" de priorité "1", dans cette ligne j'ajoute une commande "Answer" qui permet de répondre à l'instruction précédente (exten => 7000,n,Goto(IVR,s,1)).

Ensuite nous jouons notre audio personnalisé "accueil" grâce à la commande Playback.

Contrairement à la commande "Background", la commande "Playback" a l'avantage d'être interrompue à tout moment, ici notre audio permet à l'appelant de choisir les services qu'il souhaite (tapez 1 pour ... tapez 2 pour ...) il n'aura pas besoin d'attendre la fin de l'audio s'il sait déjà quel numéro taper pour joindre le service qu'il souhaite.

La commande "same => n" signifie que cette ligne fait partie de l'extension précédente avec la même priorité n = 1.

```
[IVR]
exten => s,1,Answer()
same => n,Playback(custom/accueil)
same => n,WaitExten()
```

J'écris ensuite deux autres extensions qui va me permettre dans l'ordre suivant de :

1- Jouer un audio "Vous avez entré"

2- Dire le numéro tapé par l'appelant

3- Jouer notre audio personnalisé

4- Rediriger l'appel vers le service concerné (un nouveau contexte)

5- Terminer la phase

```
exten => 1,1,Playback(you-entered)
    same => n,SayNumber(${EXTEN})
    same => n,Playback(custom/service_accueil)
    same => n,Goto(alice_call,s,1)
    same => n,Hangup()

exten => 2,1,Playback(you-entered)
    same => n,SayNumber(${EXTEN})
    same => n,Playback(custom/service_client_priv)
    same => n,Goto(bob_call,s,1)
    same => n,Hangup()
```

J'ajoute une troisième extension qui permet de :

1- Jouer un audio si l'appelant tape un mauvais numéro, elle utilise une extension spécial "i" pour "invalid"

2- Redirige au début du contexte [IVR]

3- termine la phase

```
exten => i,1,Playback(option-is-invalid)
    same => n,Goto(IVR,s,1)
    same => n,Hangup()
```

Voici à quoi ressemble mon contexte IVR dans son entièreté :

[IVR]

```
exten => s,1,Answer()  
    same => n,Playback(custom/accueil)  
    same => n,WaitExten()  
  
exten => 1,1,Playback(you-entered)  
    same => n,SayNumber(${EXTEN})  
    same => n,Playback(custom/service_accueil)  
    same => n,Goto(alice_call,s,1)  
    same => n,Hangup()  
  
exten => 2,1,Playback(you-entered)  
    same => n,SayNumber(${EXTEN})  
    same => n,Playback(custom/service_client_priv)  
    same => n,Goto(bob_call,s,1)  
    same => n,Hangup()  
  
exten => i,1,Playback(option-is-invalid)  
    same => n,Goto(IVR,s,1)  
    same => n,Hangup()
```

Je crée ensuite les deux autres contextes qui vont correspondre à mes services.

Donc dans le contexte [IVR], si l'appelant tape "1", l'appel sera redirigé vers le contexte [alice_call] qui correspond à "alice" (le service accueil) qui est enregistré dans mon fichier PJSIP et au bout de 20 secondes si elle ne répond pas alors on tombera sur répondeur. Et si l'utilisateur tape "2" alors l'appel sera redirigé vers le contexte [bob_call] qui correspond à "bob" (le service de clientèle privé).

[alice_call]

```
exten => s,1,Dial(PJSIP/alice,20)  
    same => n,Hangup()
```

[bob_call]

```
exten => s,1,Dial(PJSIP/bob,20)  
    same => n,Hangup()
```

Sécurisation

L'utilisation de l'outil cockpit :

Cockpit offre une connexion graphique simple et intuitive aux serveurs Linux distants. L'interface permet aux administrateurs de gérer les utilisateurs et les groupes, de configurer les paramètres du pare-feu et d'afficher des informations sur le matériel.

En termes d'efficacité, Cockpit est facile à mettre en œuvre au sein d'une organisation où le personnel est peu familiarisé avec le CLI.

Installation :

```
root@debianvoIP:~# apt install cockpit
```

