INSTALLER & CONFIGURER UN SERVEUR VOIP

Réseaux et Télécommunications

21 FEVRIER 2025

LA PLATEFORME - MARSEILLE
J-F. Andriatsimeva – M. Camara – R. Koehler – T. Veran

Table des matières

Introduction et contextualisation

Investir dans un serveur VOIP, oui mais pour combien ?

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

Dans un **environnement cloud** (VoIP hébergée), les coûts sont nettement inférieurs grâce à l'absence de maintenance matérielle et la flexibilité du service, pareillement pour un **environnement on-premises** (IPBX interne), les coûts de maintenance restent présents mais inférieurs à ceux d'un système téléphonique classique.

En revanche, ce type d'infrastructure peut se révéler coûteuse s'il subsiste un besoin de sécurité qui n'est pas bien gérée (besoin de firewall VoIP, VPN, chiffrement) ou si tout simplement l'infrastructure réseau n'est pas adaptée (besoin d'une meilleure bande passante ou QoS).

Matrice comparative :

	VOIP		***Téléphonie Traditionnelle***	
	Coûts Opérationnels	**Coûts de Maintenance**	**Coûts Opérationnels**	**Coûts de Maintenance**
Coût des appels	Faible (Internet, forfaits illimités)	Pas d'impact direct	Élevé (surtaxes longues distances)	Pas d'impact direct
Infrastructure	Utilise le réseau existant (moins cher)	Peut nécessiter une mise à niveau du réseau	Nécessite câblage dédié (cher)	Nécessite maintenance physique
Équipement	Téléphones IP, softphones (moins cher)	Peu d'entretien, mises à jour logicielles	PABX, lignes fixes coûteuses	Matériel physique vieillissant
Évolutivité	Facile, ajout de lignes via logiciel	Peu de maintenance nécessaire	Ajout de lignes coûteux et complexe	Nécessite intervention technique
Frais d'abonnement	Moins chers (services cloud)	Dépend de l'hébergement (cloud vs. on-premises)	Plus chers (lignes fixes et abonnements)	Frais de support et d'entretien élevés
Mises à jour	Automatiques en cloud	Facile (logiciel, IPBX en cloud)	Manuelles	Intervention technicien requise
Dépannage	Possible à distance (moins coûteux)	Maintenance logicielle simple	Doit être physique (plus long et cher)	Dépend de l'opérateur téléphonique
Redondance et sécurité	Facile avec services cloud	Sécurité renforcée avec chiffrement	Plus difficile, dépend du réseau fixe	Nécessite matériel et plan de reprise après sinistre

L’expérience utilisateur en téléphonie VOIP : flexibilité, automatisation et scalabilité

Flexibilité

Un serveur VoIP offre un large panel de configuration permettant de personnaliser l’expérience utilisateur :

ASPECT	CALL CENTER	ENTREPRISE (STANDARD TELEPHONIQUE)
VOLUME D'APPELS	Volume élevé, nécessité de gestion de files d'attente complexes.	Volume modéré, gestion d'appels plus simple.
REPARTITION DES APPELS (ACD)	Configuration avancée avec règles personnalisées (compétences, priorités, etc.).	Configuration basique, souvent basée sur la disponibilité.
INTEGRATION CRM/LOGICIELS	Intégration poussée avec CRM, outils de gestion des tickets, etc.	Intégration limitée ou optionnelle.

1. Files d'attente intelligentes

- **Personnalisation :** Les files d'attente peuvent être configurées avec des messages d'attente personnalisés, de la musique en attente, ou des informations sur le temps d'attente estimé.

  ➔ **Expérience utilisateur :** Cela améliore l'expérience de l'appelant en le gardant informé et engagé pendant l'attente.

---

2. Répartition des appels (ACD - Automatic Call Distribution)

- **Personnalisation :** Les règles de répartition des appels peuvent être configurées en fonction de critères spécifiques (compétences des agents, langue, priorité de l'appel, etc.).

  ➔ **Expérience utilisateur :** Cela permet de diriger les appels vers les agents les plus qualifiés, réduisant les temps d'attente et améliorant la satisfaction client.

---

3. Intégration avec les outils CRM et autres logiciels

- **Personnalisation :** Les serveurs VoIP peuvent être intégrés à des CRM, des systèmes de gestion des tickets, ou d'autres outils métiers pour afficher des informations contextuelles lors des appels.

➔ **Expérience utilisateur :** Les agents ont accès à des données pertinentes en temps réel, ce qui permet des interactions plus personnalisées et efficaces.

Automatisation

*Un serveur VoIP peut être intégré ou connecté à d'autres systèmes (Annuaires de contacts (AD), CRM) nuus permettant d'automatiser certaines tâches et de faciliter la communication avec nos clients. L' intégration avec un **CRM** (ex. Salesforce, HubSpot) permet d'une part d'afficher les informations clients lorsqu'un appel est reçu., tandis que l'intégration d'un **annuaire de contacts (AD, LDAP)** facilite la gestion des utilisateurs et des permissions.*

Quelques noms parmi les fournisseurs de services VOIP

1. Netxtiva
2. Ringcentral
3. OVH telecom

Architecture VOIP possible pour trois sites marchands :

1. ***Amazon*** (Service client Amazon) → Utilisation de centres d'appels VoIP pour la gestion des commandes et réclamations.
2. ***Uber*** (Support client pour chauffeurs et passagers) → Intégration VoIP pour le support en cas de problème avec une course.
3. ***Airbnb*** (Service assistance voyageurs et hôtes) → Utilisation de la VoIP pour gérer les litiges et fournir un support rapide.
4. ***Banques en ligne*** (ex. Revolut, N26) → Support client via VoIP intégré à leur application.

Architecture possible entre deux types de prestataires :

SITE E-COMMERCE	SITE DE SERVICE
Serveur VoIP (Asterisk, Twilio, 3CX) pour gérer les appels clients (commandes, réclamations).	**Serveur VoIP (Asterisk, Twilio, FreePBX)** pour le support technique ou la prise de rendez-vous.
CRM (Salesforce, HubSpot) pour afficher l'historique des commandes lors d'un appel.	**CRM (Zoho, Freshdesk, Salesforce Service Cloud)** pour suivre les demandes des clients.
Système IVR (Interactive Voice Response) pour guider les clients vers le bon service (suivi de commande, retour produit).	**Système IVR et chatbot vocal** pour automatiser les demandes courantes (prise de RDV, FAQ).
Intégration avec une boutique en ligne (Shopify, WooCommerce) pour automatiser les rappels et confirmations.	**Connexion avec une application mobile ou web** pour contacter le service client en un clic.
Annuaires AD/LDAP pour gérer les comptes des agents de support client.	**Annuaires AD/LDAP** pour gérer les accès des conseillers et techniciens.

Enfin éventuellement, le volet analyse et reporting est à considérer pour les deux types d'organisation, avec la mise en place d'outils comme ELasticsearch pour le site-e-commerce, ou PowerBI pour le site de service, afin de s'assurer de la qualité des services fournis.

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 et reproduire une architecture locale et la déployer sur d'autres sites...

Quelques différences notables en matière de scalabilité pour deux types d'organisations différentes :

ASPECT	CALL CENTER	ENTREPRISE (STANDARD TELEPHONIQUE)
BESOIN EN SCALABILITE	**Critique** : Doit gérer des pics d'appels et des variations de volume élevées.	**Modéré** : Scalabilité liée à la croissance progressive de l'entreprise.
TYPE DE SOLUTION PRIVILEGIE	**Cloud** : Scalabilité instantanée, ajout/suppression de lignes et agents en temps réel.	**Hybride (Cloud + On-Premise)** : Équilibre entre flexibilité et coûts.
GESTION DES RESSOURCES	**Dynamique** : Répartition des appels en temps réel, intégration de ressources externes si nécessaire.	**Simplifiée** : Ajout d'extensions ou de lignes pour nouveaux employés ou départements.
ADAPTABILITE AUX PICS	**Haute** : Doit absorber des pics d'appels soudains (campagnes, crises, etc.).	**Faible** : Volume d'appels stable, pics moins fréquents.
MODELE DE TARIFICATION	**Flexible (pay-as-you-go)** : Paiement en fonction de l'utilisation.	**Forfaitaire** : Coûts prévisibles, adaptés à une croissance progressive.
INTEGRATION EXTERNE	**Oui** : Peut intégrer des centres de contact externalisés ou des solutions tierces.	**Limitée** : Moins besoin d'intégration externe.

Le chiffrement d'appels sous téléphonie VOIP

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.

Les chiffrements les mieux adaptés à la VoIP :

PROTOCOLE	USAGE	ALGORITHME DE CHIFFREMENT	SECURITE
TLS (SIPS)	Chiffrement du signal SIP	AES-128/256-GCM, ChaCha20 (TLS 1.3)	Protège l'authentification et les métadonnées
SRTP	Chiffrement de la voix et vidéo	AES-128/256-GCM	Protège le contenu des appels

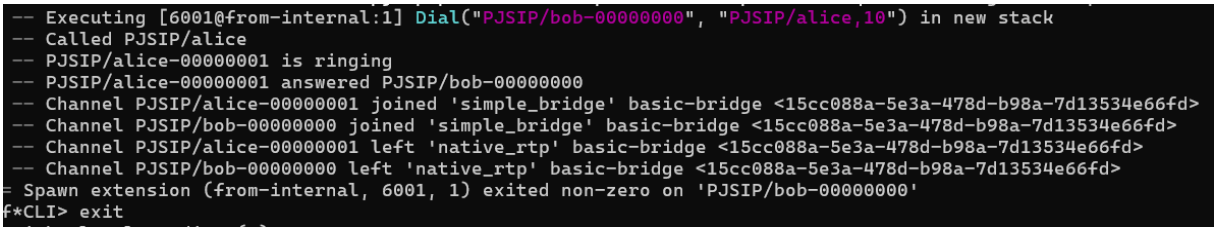
ZRTP	Échange de clés pour SRTP	Diffie-Hellman (DHE, ECDH)	Empêche les écoutes clandestines

On peut également traiter de quelques algorithmes comme **AES**[1], **HMAC**[2], **RSA** [3]et **ECDHE**[4], qui complètent ces protocoles en offrant des mécanismes de chiffrement et d'authentification plus performants.

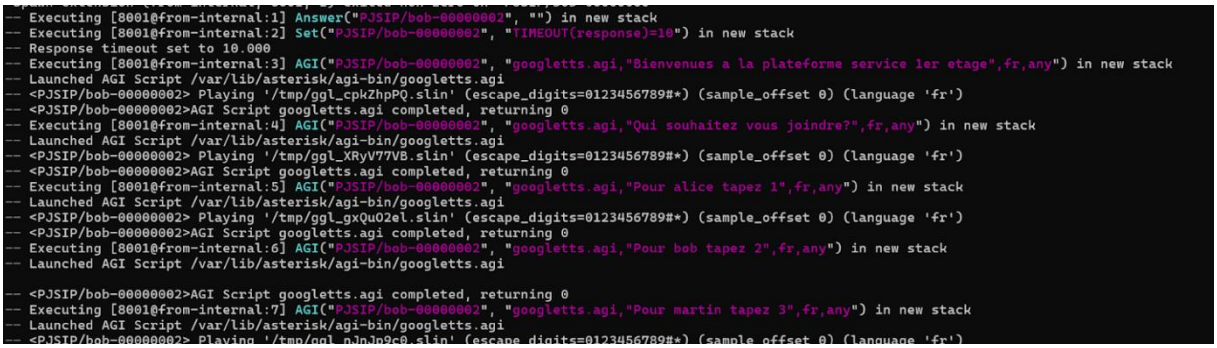
[1] Utilisé dans SRTP et ZRTP pour chiffrer les flux multimédias (Efficace pour les flux multimédias en temps réel).
[2] Authentification des paquets dans SRTP (Empêche la modification des paquets en transit, souvent combiné avec SHA-256).
[3] Authentification des serveurs et échange de clés dans TLS (permet un échange sécurisé des clés de chiffrement, idéal pour les certificats TLS).
[4] Utilisé avec TLS pour sécuriser les connexions VoIP avec une protection optimale (sécurité renforcée avec des clés plus courtes que RSA).

Présentation fonctionnelle

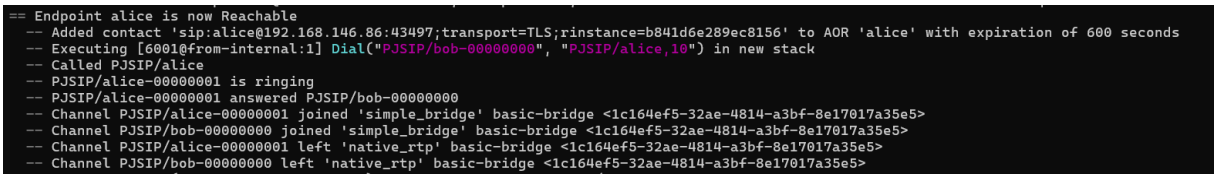
Premier appel UDP



Appel IVR 8001



Appel TLS



Veille technologique

Le marché de la téléphonie VoIP, de plus en plus automatisé (et sécurisé) se voit désormais aussi gagné par les solutions cloud et UCaaS (Unified Communications as a Service). Les solutions Open Source comme Asterisk demeurent populaires, elles nécessitent souvent une expertise technique pour être configurées et utilisées efficacement. Cependant, la croissance des menaces cybernétiques et les attaques de phishing constituent une menace croissante pour les entreprises, nécessitant des protocoles de chiffrement avancés et des solutions anti-fraude pour protéger les données sensibles. Enfin, l'optimisation des performances VoIP est un enjeu clé pour les entreprises, avec des technologies comme SD-WAN (Software-Defined Wide Area Networking), QoS (Quality of Service) avancée et IA (Intelligence Artificielle) qui peuvent aider à améliorer la qualité de service, à réduire les coûts et à accroître la productivité.

Solutions existantes sur le marché (Opensource/payantes)

Panorama général :

Type	Solutions	Avantages	Inconvénients
Open Source	Asterisk[5], FreePBX, FusionPBX, Kamailio, OpenSIPS	Personnalisation totale, pas de coût de licence, compatible avec plusieurs protocoles.	Maintenance technique élevée, configuration complexe.
Hybride (Open Source + Interface graphique)	FreePBX (basé sur Asterisk), Issabel, Elastix	Administration simplifiée avec UI Web, large communauté.	Moins flexible qu'Asterisk brut.
Solutions payantes (On-premises)	3CX, Cisco CallManager, Avaya IP Office, Mitel	Solution clé en main, support technique, intégration avec CRM.	Coût élevé, dépendance à l'éditeur.
Solutions Cloud VoIP / UCaaS	Twilio, RingCentral, Zoom Phone, Microsoft Teams, Google Voice	Pas d'infrastructure à gérer, mises à jour automatiques, scalabilité.	Abonnement mensuel, dépendance à un fournisseur cloud.

Nouvelles technologies et protocoles

Protocole / Technologie	Description	Avantages
Opus Codec	Codec audio avancé conçu pour VoIP et WebRTC.	Meilleure qualité audio, faible latence, adaptabilité aux réseaux instables.
EVS (Enhanced Voice Services)	Codec utilisé en VoLTE/VoNR (5G).	Qualité audio HD+, robustesse aux pertes de paquets.
SRTP (Secure Real-time Transport Protocol)	Chiffrement des flux RTP (voix et vidéo).	Protection contre l'écoute clandestine et le vol d'informations.
ZRTP (Zimmermann Real-time Transport Protocol)	Chiffrement E2E sans infrastructure PKI.	Sécurisation des appels VoIP sans tiers de confiance.
STIR/SHAKEN	Authentification d'identité pour appels SIP.	Protection contre le spoofing et le spam vocal.
QUIC/SIP over QUIC	Remplace TCP/UDP pour SIP.	Faible latence, meilleure fiabilité sur les réseaux instables.
SIP-I (SIP for ISUP Interworking)	Intégration SIP/PSTN avancée.	Meilleure compatibilité avec les réseaux télécoms traditionnels.

Évolutions des produits existants

S'agissant de solutions open-source :

→ **Plus de personnalisation** (ex. Asterisk, FreePBX)

Asterisk 20+	FreePBX
Amélioration du support WebRTC pour les appels VoIP depuis un navigateur. Optimisation des performances SIP et de la gestion des appels à grande échelle.	Meilleure intégration avec **Microsoft Teams** et **Zoom Phone**.

[5] En bleu, les principales solutions VOIP sur le marché.

Support renforcé des API RESTful pour faciliter l'intégration avec des CRM et ERP.	Ajout de fonctionnalités avancées de **failover (basculement automatique en cas de panne)**. **Améliorations de sécurité** avec support natif de **TLS 1.3 et SRTP activé par défaut**.

S'agissant de solutions cloud et commerciales :

→ **Fonctionnalités d'analyse vocale et de call tracking** (ex. 3CX, Twilio Voice API)

3CX	Twilio Voice API
Intégration complète avec WhatsApp et Facebook Messenger pour unifier la communication VoIP et messagerie. Support WebRTC natif amélioré pour les appels sans installation de logiciel. Renforcement des outils de vidéoconférence et partage d'écran.	Ajout de **l'IA générative pour l'analyse des conversations en temps réel**. Nouveaux outils de **call tracking et scoring des appels** via machine learning. Meilleure **latence mondiale grâce à des routes SIP optimisées**.

De manière générale, l'optimisation réseau :

- Déploiement du VoNR (Voice over New Radio) avec la 5G pour remplacer VoLTE.
- Standardisation des appels WebRTC pour permettre des appels VoIP sans plugins ni logiciels.

Avantages et inconvénients

Problèmes de sécurité

Panorama des problèmes de sécurité VOIP :

Problème de Sécurité	Description	Solutions
Eavesdropping (Écoute clandestine)	Interception des appels VoIP non chiffrés.	Chiffrement **SRTP, ZRTP, DTLS-SRTP**.
Spoofing & Vishing	Usurpation d'identité pour du phishing vocal.	Authentification forte, protocole **STIR/SHAKEN**.
DoS (Denial of Service) & Flooding	Surcharge du serveur VoIP via des requêtes malveillantes.	Firewall VoIP, anti-DDoS, SBC (Session Border Controller).
SPIT (Spam over IP Telephony)	Envoi massif d'appels/spams VoIP.	Listes noires, filtrage intelligent via IA.
Fraude téléphonique (Toll Fraud)	Piratage de comptes VoIP pour des appels internationaux frauduleux.	Surveillance des logs, restrictions d'appel géographiques.

Il apparaît que les solutions cloud imposent des mesures de sécurité natives mais restent vulnérables aux attaques DDoS.

Performances et optimisations

Problème	Impact	Solutions
Latence et Jitter	Voix hachée, retards.	QoS (Quality of Service), codecs optimisés (Opus, G.722).
Pertes de paquets	Coupures audios.	SD-WAN, FEC (Forward Error Correction).
Déconnexion des appels	Interruption de la communication.	SBC, monitoring des connexions SIP.
Charge serveur élevée	Ralentissements, saturation.	Load balancing, déploiement en cluster.

Pistes d'optimisations :

- **Déploiement du codec Opus** pour une meilleure qualité audio avec faible bande passante.
- **SD-WAN pour garantir une meilleure QoS** et éviter les pertes de paquets.

Exemples d'implémentation

Concepts et les protocoles utilisés par notre serveur

Installation de Asterisk

1/ Pré -Installation Asterisk(.tar gz)

Après avoir installé Debian, nous avons déjà des outils utiles tels que nano, wget et tar dans le paquet. Nous allons également installer curl pour nos besoins et mettre à jour le système d'exploitation.

apt update

apt upgrade

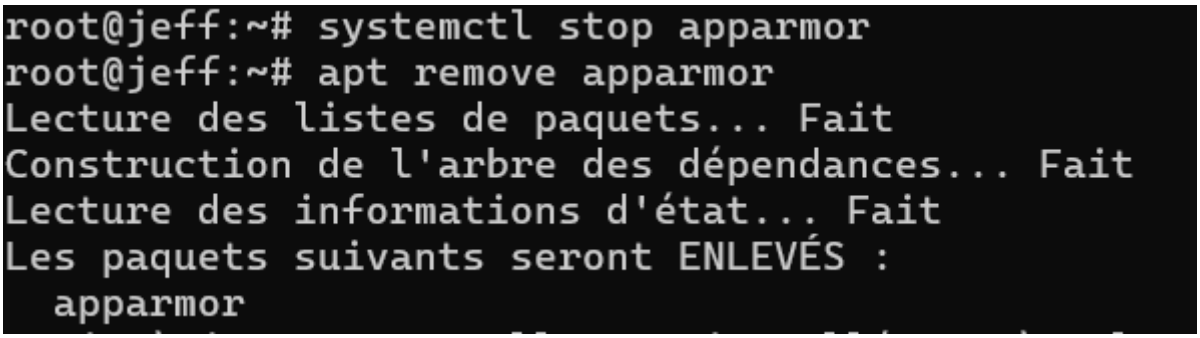
apt install curl

```
jeff@jeff:~$ su -
Mot de passe :
root@jeff:~# apt install curl
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  libcurl4
Les NOUVEAUX paquets suivants seront installés :
  curl libcurl4
```

Nous allons désinstaller le système de contrôle d'accès par défaut d'Apparmor. Il pourrait interférer avec le fonctionnement de notre astérisque.

systemctl stop apparmor

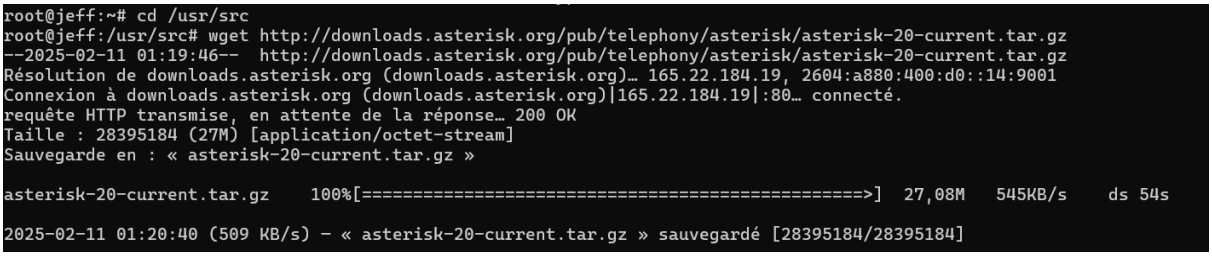
apt remove apparmor



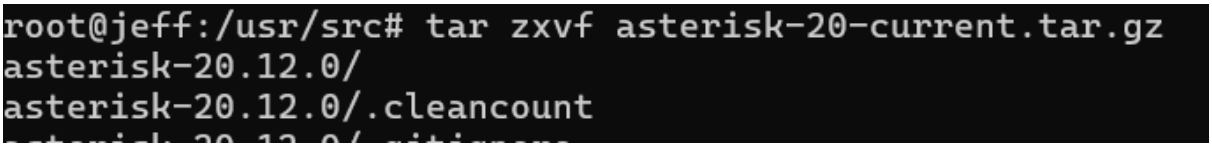
Nous allons maintenant télécharger le programme d'installation d'Asterisk et le décompresser.

cd /usr/src

wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-20-current.tar.gz



tar zxvf asterisk-20-current.tar.gz



rm -rf asterisk-20-current.tar.gz

Dans l'étape suivante, nous allons préparer l'environnement pour la compilation et l'exécution du serveur Asterisk sur le système Debian. Nous utiliserons les outils et le script" install prereq" fournis avec le projet Asterisk, qui aident à automatiser le processus d'installation. Le script identifie et installe les dépendances système nécessaires à la compilation et à

l'exécution d'Asterisk, ainsi que les bibliothèques, outils ou autres composants nécessaires au bon fonctionnement d'Asterisk.

cd asterisk-20/*

contrib/scripts/install_prereq install

```
root@jeff:/usr/src# cd asterisk-20*/
root@jeff:/usr/src/asterisk-20.12.0# contrib/scripts/install_prereq install
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
```

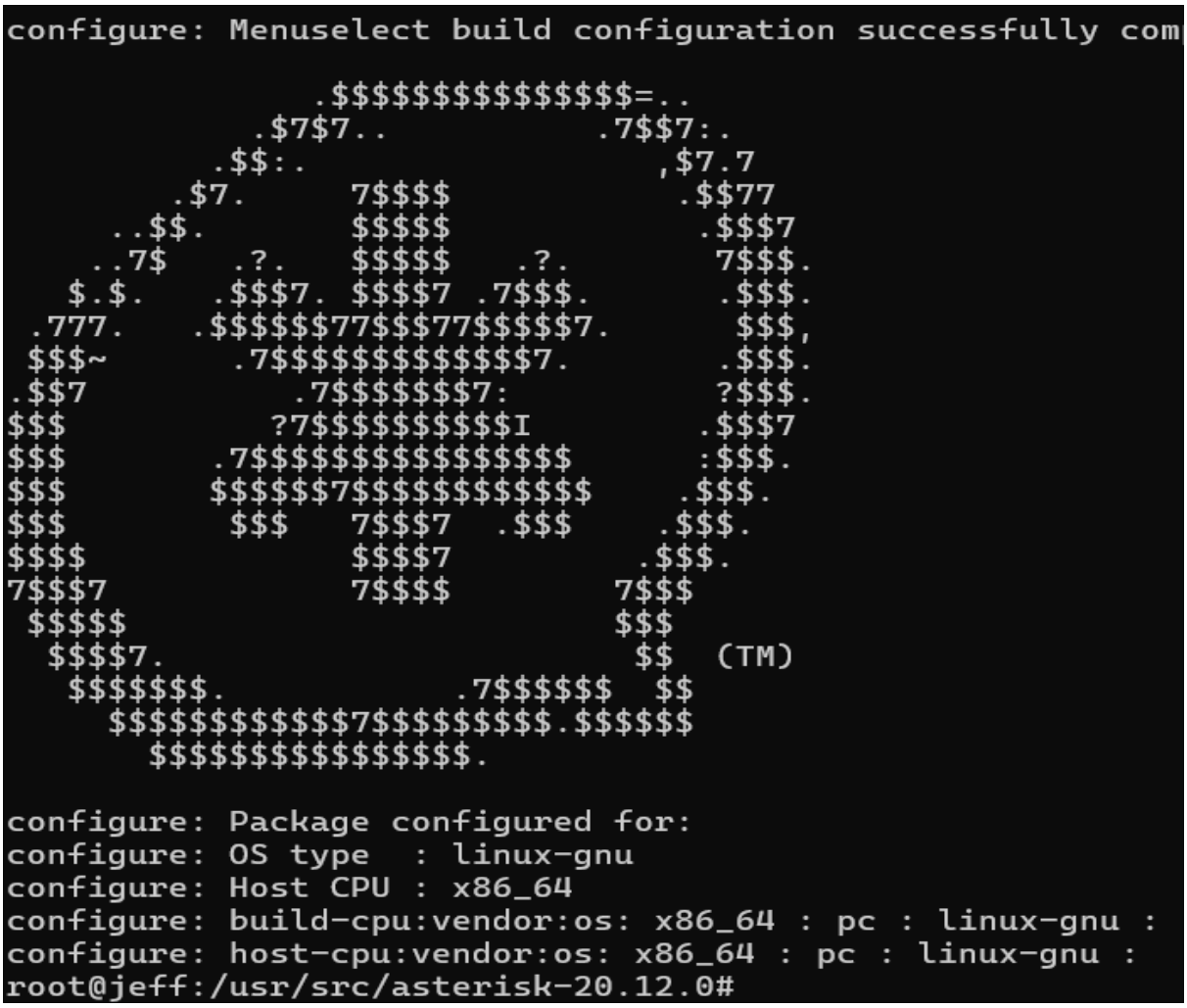
A la fin de l'exécution du script on peut voir:

```
##############################################
## install completed successfully
##############################################
root@jeff:/usr/src/asterisk-20.12.0#
```

Utilisation de la commande ./configure dans le chemin suivant :

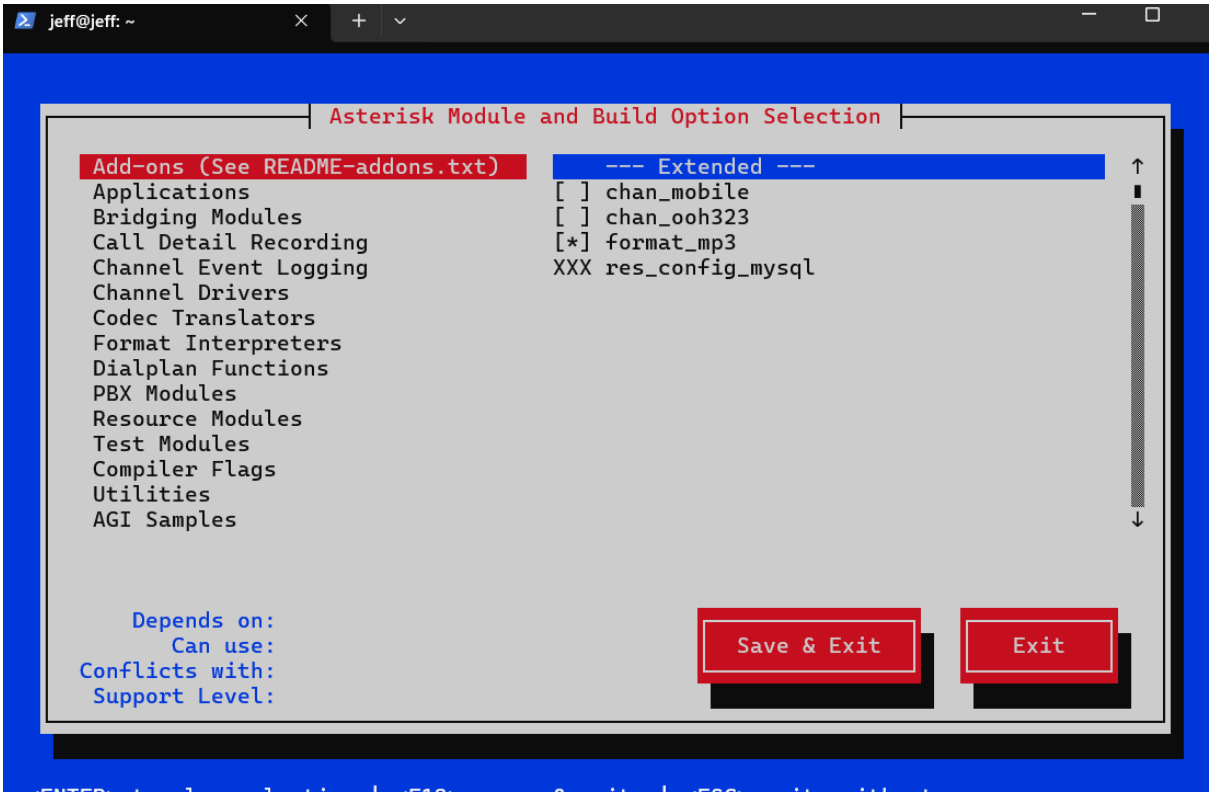
cd /usr/src/asterisk-20/*

Résultat:



Ensuite nous allons devoir taper la commande :
make menuselect, pour pouvoir atterrir dans un menu ou il faudra sélectionner les bons modules.

Configuration de Asterisk



Modules de bases

Module	Description
`app_dial`	Permet de composer un numéro et établir un appel
`app_echo`	Teste l'écho sur une ligne, utile pour le dépannage
`app_playback`	Joue un fichier audio pendant un appel
`app_voicemail`	Active la messagerie vocale (optionnel mais utile)
`app_transfer`	Permet de transférer un appel vers un autre poste

Protocole VOIP

Ces modules permettent à Asterisk de gérer les communications VoIP.

Module	Description
`chan_pjsip`	**Pilote SIP moderne recommandé** pour les appels VoIP
`res_pjsip`	Ressources nécessaires pour le protocole PJSIP
`chan_sip`	**Ancien pilote SIP** (déprécié, ne pas utiliser sauf nécessité)
`chan_iax2`	Protocole IAX2 pour connecter plusieurs serveurs Asterisk
`res_srtp`	Gestion du **chiffrement SRTP** pour sécuriser les appels
`res_rtp_asterisk`	Gestion du **trafic RTP** (voix et vidéo)

Codecs Audio

Les codecs déterminent la qualité audio et la consommation de bande passante.

Module	Description
`codec_ulaw` et `codec_alaw`	Codecs **G.711** (qualité standard)
`codec_gsm`	Compression **GSM**, bon compromis qualité/bande passante
`codec_g722`	**HD Voice (G.722)**, utilisé pour les communications haute qualité
`codec_opus`	**Opus**, excellent codec pour la VoIP (bonne compression et qualité)

Format de fichier audio

Ces modules permettent à Asterisk d'enregistrer et lire des fichiers audio.

Module	Description
`format_wav`	Enregistre et joue des fichiers **WAV** (standard)
`format_mp3`	Support du format **MP3** (nécessite `addons`)
`format_gsm`	Enregistrement en **GSM**, économise de l'espace disque

Module reseaux et securite

Ces modules sont nécessaires pour les communications sécurisées et le réseau.

Module	Description
`res_http_websocket`	**Obligatoire pour WebRTC** et interfaces web
`res_crypto`	Support du **chiffrement RSA, AES** (sécurité)
`res_resolver_unbound`	Gestion avancée du **résolveur DNS**

Gestions des bases de données

Asterisk peut stocker ses logs d'appels dans des bases de données.

`cdr_csv`	Stocke les logs d'appels en **CSV** (fichiers texte)
`func_odbc`	Permet à Asterisk d'accéder à une base de données via ODBC

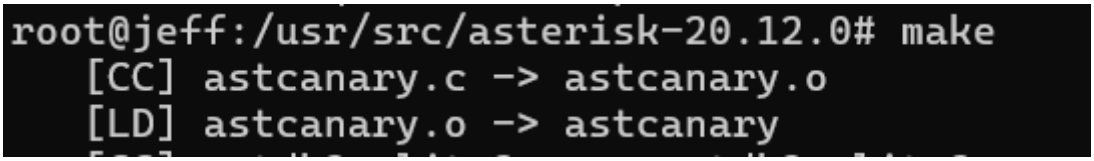
Outils supplémentaires

Ces outils facilitent l'administration et le dépannage d'Asterisk.

Module	Description
`menuselect.makeopts`	**Personnalisation des options de compilation**
`astdb2sqlite3`	Convertit la base de données interne d'Asterisk **(AstDB)** en SQLite

Une fois vos sélections effectuées, cliquez sur Enregistrer et quitter.

Invocation de la commande [Make] qui **compilera** Asterisk en prenant en compte les modules sélectionnés et leur configuration.



Résultat après invocation de la commande [Make]:

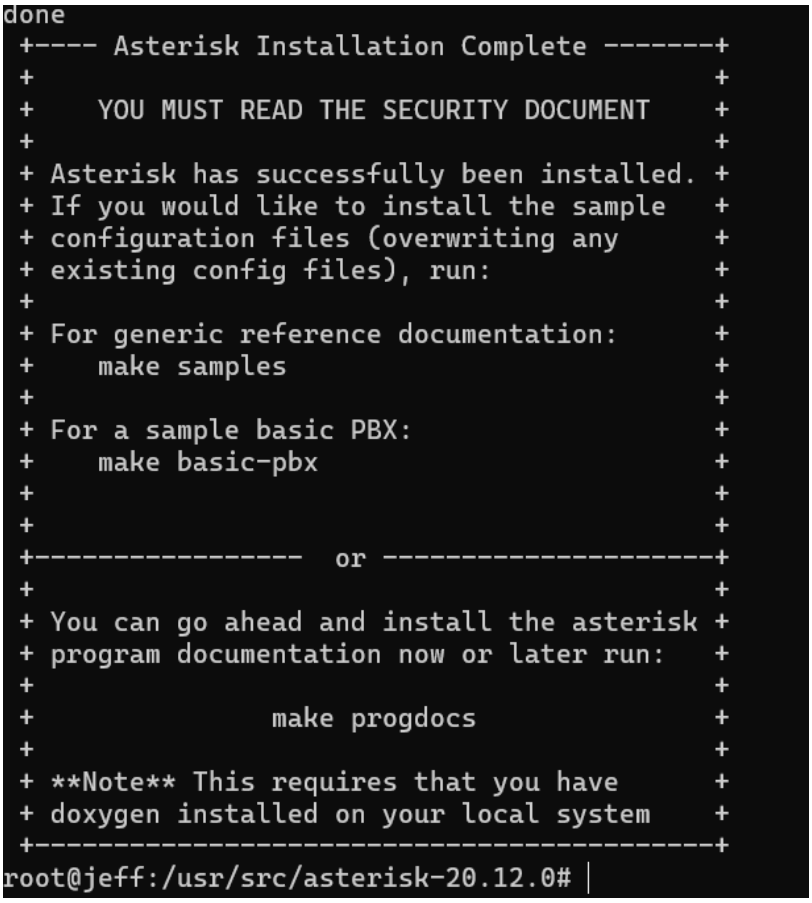
```
************************************************************
***                                                      ***
***    ---> IMPORTANT INFORMATION ABOUT format_mp3 <---   ***
***                                                      ***
*** format_mp3 has been selected to be installed, but the ***
*** MP3 decoder library has not yet been downloaded into  ***
*** the source tree.  To do so, please run the following  ***
*** command:                                              ***
***                                                      ***
***          $ contrib/scripts/get_mp3_source.sh         ***
***                                                      ***
************************************************************

Building Documentation For: channels pbx apps codecs formats cdr cel bridges funcs tests mai
n res addons
 +---------- Asterisk Build Complete ----------+
 + Asterisk has successfully been built, and +
 + can be installed by running:              +
 +                                           +
 +               make install                +
 +-------------------------------------------+
root@jeff:/usr/src/asterisk-20.12.0#
```

On va installer Asterisk compilé:

La commande *make install* installera le logiciel compilé sur le système.

Résultat:



Nous pouvons maintenant générer des exemples de fichiers de configuration et les déplacer vers le répertoire d'exemples, où nous pourrons les utiliser comme base pour une configuration ultérieure de nos services.

make samples

mkdir /etc/asterisk/samples

mv /etc/asterisk/.* /etc/asterisk/samples/*

La commande suivante make basic-pbx va créer une configuration et un scénario de PBX (Private Branch Exchange) de base dans Asterisk. Le scénario PBX est un exemple de configuration d'Asterisk en tant que central téléphonique privé qui permet d'établir des appels entre des numéros de téléphone internes. Voici exactement ce que fait la commande :

1. **Créer une configuration de base** : la commande génère une configuration Asterisk de base, qui inclut des fichiers de configuration pour les utilisateurs, les extensions, les contextes et les itinéraires sortants.
2. **Configuration utilisateur** : La commande peut créer des configurations utilisateur, ce qui vous permet d'établir des connexions entre eux au sein du système PBX.

3. **Configuration des extensions** : le scénario PBX contient des exemples d'extensions qui définissent les actions à entreprendre lorsque les utilisateurs composent des extensions spécifiques.
4. **Contextes** : La configuration du PBX est divisée en contextes, qui définissent quels numéros de téléphone et extensions sont disponibles dans quelle partie du système.
5. **Routes sortantes** : un scénario PBX peut également inclure une configuration de route sortante qui détermine les numéros que les utilisateurs peuvent composer pour les appels externes.

La dernière commande d'installation créera des fichiers de démarrage :

make config

Nous sommes maintenant prêts à exécuter le système Asterisk.

Tout d'abord, nous allons ajouter le démarrage d'Asterisk au démarrage automatique: systemctl enable asterisk.service

```
root@jeff:/usr/src/asterisk-20.12.0# systemctl enable asterisk.service
asterisk.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable asterisk
```

Le message ci-dessus "asterisk.service is not a native service"ne devrait pas nous inquiéter, car Debian se charge d'ajouter le service au démarrage lui-même. L'ensemble de commandes suivantes nous permettra d'exécuter Asterisk et de vérifier son état:

-systemctl start asterisk.service

-systemctl status asterisk.service

Résultat:

```
root@jeff:/usr/src/asterisk-20.12.0# systemctl start asterisk.service
root@jeff:/usr/src/asterisk-20.12.0# systemctl status asterisk.service
● asterisk.service - LSB: Asterisk PBX
     Loaded: loaded (/etc/init.d/asterisk; generated)
     Active: active (running) since Tue 2025-02-11 12:23:42 CET; 7s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 44003 ExecStart=/etc/init.d/asterisk start (code=exited, status=0/SUCCESS)
      Tasks: 46 (limit: 2264)
     Memory: 41.9M
        CPU: 1.010s
     CGroup: /system.slice/asterisk.service
             └─44015 /usr/sbin/asterisk

févr. 11 12:23:41 jeff systemd[1]: Starting asterisk.service - LSB: Asterisk PBX...
févr. 11 12:23:42 jeff asterisk[44003]: Starting Asterisk PBX: asterisk.
févr. 11 12:23:42 jeff systemd[1]: Started asterisk.service - LSB: Asterisk PBX.
root@jeff:/usr/src/asterisk-20.12.0#
```

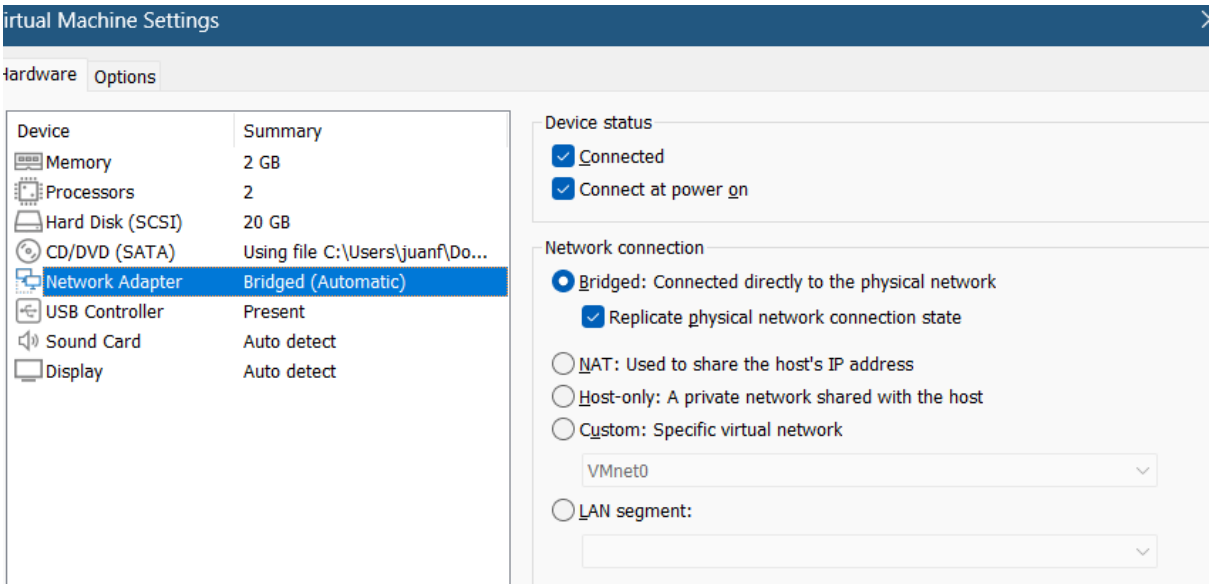
Nous pouvons maintenant lancer sur le terminal Asterisk:

asterisk -rvvvv

Pour ensuite accéder au fichier sip.conf sachant que depuis la version 17 le sip.conf s'est transformer en pjsip.conf du coup on va devoir écrire la commande :
nano /etc/asterisk/pjsip.conf puis créez vos utilisateurs.

Dans **VMware Workstation/Player** :

- Cliquez-droit sur votre VM
- Sélectionnez "Settings"
- Allez dans l'onglet "Hardware"
- Sélectionnez "Network Adapter"
- Dans le menu déroulant, choisissez "Bridged" ou "Mode pont" et "Replicate physical network connection state"
- Validez avec "OK"



Configuration pjsip

```
root@kindzu:/etc/asterisk# mkdir -p /var/log/asterisk
root@kindzu:/etc/asterisk# chown -R asterisk:asterisk /var/log/asterisk
chown: utilisateur incorrect: « asterisk:asterisk »
root@kindzu:/etc/asterisk# groupadd asterisk
root@kindzu:/etc/asterisk# useradd -r -d /var/lib/asterisk -g asterisk asterisk
root@kindzu:/etc/asterisk# chown -R asterisk:asterisk /var/log/asterisk
root@kindzu:/etc/asterisk# chmod -R 755 /var/log/asterisk
root@kindzu:/etc/asterisk# chown -R asterisk:asterisk /etc/asterisk
root@kindzu:/etc/asterisk# chown -R asterisk:asterisk /var/lib/asterisk
root@kindzu:/etc/asterisk# chown -R asterisk:asterisk /var/spool/asterisk
root@kindzu:/etc/asterisk# chown -R asterisk:asterisk /var/run/asterisk
root@kindzu:/etc/asterisk# systemctl restart asterisk
```

A mettre a la fin du fichier nano /etc/asterisk/extensions.conf

```
[internal]
exten => 1101,1,Dial(PJSIP/1101)
exten => 1102,1,Dial(PJSIP/1102)
same => n,Hangup()
```

1101 représente l'utilisateur 1, puis 1102 représente l'utilisateur 2

il faudrait ensuite pour sécuriser l'appel, générer et ajouter une clés de déchiffrage ainsi que ça certification comme ci-dessous :

```
root@jeff:~# cd /etc/asterisk
root@jeff:/etc/asterisk# mkdir certs
root@jeff:/etc/asterisk# cd certs
root@jeff:/etc/asterisk/certs# openssl req -new -x509 -days 3650 -nodes -out asterisk.pem -keyout asterisk.key
...+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*....+.......+++++++++++++++++++++++++++++++++++
.......+......+......+.....+.......+...+.......+.+........+............+..+...+.........+......+.........+...
.+................+...+..+...+.+.............+....+...+...+..+.......+..+..........+...+........+..........+
.......+..+...+....+......+...............+...+..+....+..........+..................+.+.............+......++
++++++++++++++++++
...+..+.+..+...+.......+..+......+.+...................+++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++*..+.........+...+........+...+.......+.........+...+..+....+++++
+++++++++++++++
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:FRANCE
Locality Name (eg, city) []:MARSEILLE
Organization Name (eg, company) [Internet Widgits Pty Ltd]:.
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:Asterisk jeff
Email Address []:.
```

entrer dans votre cheminement asterisk avec :

Création d'un nouveau fichier pjsip.conf via deux commandes :

> /etc/asterisk/pjsip.conf

nano /etc/asterisk/pjsip.conf

Création de nos trois utilisateurs : Alice, Bob, Martin

```
  GNU nano 7.2
[transport-udp]
type=transport
protocol=udp
bind=0.0.0.0


;Templates for the necessary config sections

[endpoint_internal](!)
type=endpoint
context=from-internal
disallow=all
allow=ulaw
language=fr
transport=transport-tls

 Transport TLS
[transport-tls]
type=transport
protocol=tls
bind=0.0.0.0:5061
cert_file=/etc/asterisk/keys/certificate.pem
priv_key_file=/etc/asterisk/keys/private.key
method=tlsv1_2

[auth_userpass](!)
type=auth
auth_type=userpass

[aor_dynamic](!)
type=aor
max_contacts=4

;Definitions for our phones, using the templates above

[alice](endpoint_internal)
auth=alice
aors=alice
[alice](auth_userpass)
password=root411
username=alice
[alice](aor_dynamic)

[bob](endpoint_internal)
auth=bob
aors=bob
[bob](auth_userpass)
password=root411
username=bob
[bob](aor_dynamic)

[martin](endpoint_internal)
auth=martin
aors=martin
[martin](auth_userpass)
password=root411
username=martin
[martin](aor_dynamic)
```

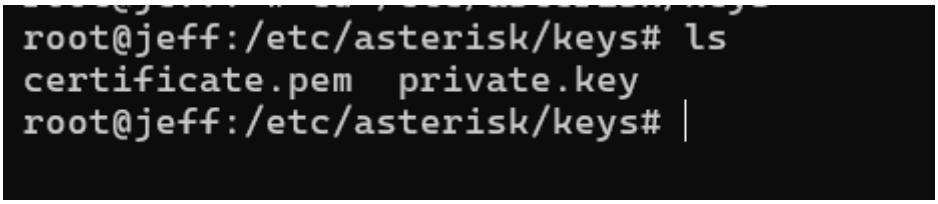
Protocoles utilisés en second lieu : TLS

Création de la clef :

mkdir /etc/asterisk/keys

```
cd /etc/asterisk/keys
openssl req -x509 -newkey rsa:2048 -keyout private.key -out certificate.pem -days 365 -nodes
```

Cheminement de la clef :



Activer les modules pjsip : avec la commande asterisk -rx "module reload res_pjsip.so"

Ensuite on va devoir s'assurer que le dossiers log existe puis créer un groupe asterisk et puis un utilisateur et à la fin on restart :

Configuration extensions.conf

```
[internal]
exten => alice,1,Dial(PJSIP/alice)
exten => bob,1,Dial(PJSIP/bob)
exten => martin,1,Dial(PJSIP/martin)
same => n,Hangup()

[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 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


;Second IVR
;Asterisk répond
exten => 8001,1,Answer()
;On met un timeout de 10 secondes pour le choix du destinatire
exten => 8001,2,Set(TIMEOUT(response)=10)
;On annonce les différents choix
exten => 8001,3,agi(googletts.agi,"Bienvenues a la plateforme service 1er etage",fr,any)
exten => 8001,4,agi(googletts.agi,"Qui souhaitez vous joindre?",fr,any)
exten => 8001,5,agi(googletts.agi,"Pour alice tapez 1",fr,any)
exten => 8001,6,agi(googletts.agi,"Pour bob tapez 2",fr,any)
exten => 8001,7,agi(googletts.agi,"Pour martin tapez 3",fr,any)
exten => 8001,8,agi(googletts.agi,"Appuyez sur dièse si vous souhaitez réécouter ce  message",fr,any)
;On attend que l'utilisateur appuis sur une touche
exten => 8001,9,WaitExten()
;Si l'utilisateur appuis sur 1 on va à la priorité 1 du numéro 6001
exten => 1,1,Goto(6001,1)
;Si l'utilisateur appuis sur 1 on va à la priorité 1 du numéro 6002
exten => 2,1,Goto(6002,1)
;Si l'utilisateur appuis sur 1 on va à la priorité 1 du numéro 6003
exten => 3,1,Goto(6003,1)
;Si l'utilisateur tape un numéro compris entre 4 et 9 et # il retourne à l'étape 4 de l'IVR
exten => _[4-9#],1,Goto(8001,4)
;Si l'utilisateur ne fais rien il retourne à l'étape 4 de l'IVR au bout de 10 secondes.
exten => t,1,Goto(8001,4)
```

Test et validation

Plan de Test : Fonctionnalité d'Appel Entrant

État initial du système

- Le serveur Asterisk 20 est opérationnel et configuré correctement.
- Les extensions SIP sont enregistrées et actives.
- Les utilisateurs finaux ont des téléphones IP configurés et connectés au réseau.

Fonctionnalité testée / périmètre du test

- Fonctionnalité : Appel entrant
- Périmètre : Vérifier que les appels entrants sont correctement acheminés vers les extensions SIP configurées.

Description du comportement attendu

- Lorsqu'un appel entrant est reçu, le serveur Asterisk doit acheminer l'appel vers l'extension SIP configurée.
- L'utilisateur final doit entendre la sonnerie et pouvoir répondre à l'appel.
- La qualité audio doit être claire et sans interruptions.

Séquence ou étapes de test

1. **Initialisation de l'appel** :
    - Utiliser un téléphone externe pour composer le numéro associé au serveur Asterisk.
2. **Acheminement de l'appel** :
    - Vérifier que l'appel est correctement acheminé vers l'extension SIP configurée.
3. **Réception de l'appel** :
    - L'extension SIP doit sonner.
    - L'utilisateur final doit pouvoir décrocher et entendre l'appelant.
4. **Qualité de l'appel** :
    - Vérifier la qualité audio de l'appel (clarté, absence de bruit, etc.).
5. **Fin de l'appel** :
    - Raccrocher l'appel et vérifier que la communication est correctement terminée.

Commentaires

- S’assurer que les configurations SIP sont correctement définies avant de commencer le test.
- Pour une extension future, utiliser des outils de monitoring pour vérifier les logs et les performances du serveur pendant le test (tableau de bord).

Résultat du test : OK

Commentaires supplémentaires : L'appel entrant a été correctement acheminé vers l'extension SIP. La qualité audio était claire et sans interruptions. Aucun problème détecté.

Conclusion

La téléphonie VOIP représente une avancée majeure en matière de communication d'entreprise, offrant flexibilité, automatisation et scalabilité. L'investissement dans un serveur VOIP doit être évalué non seulement en termes de coûts, mais aussi en tenant compte des bénéfices fonctionnels et stratégiques qu'il apporte.

L'analyse des solutions existantes, qu'elles soient open source ou propriétaires, ainsi que des évolutions technologiques, a permis de mieux comprendre les enjeux de sécurité, de performance et d'optimisation. La mise en place d'un serveur VOIP repose sur des protocoles éprouvés et des mécanismes de chiffrement avancés pour garantir la confidentialité des communications.

Grâce à l'intégration de technologies comme LDAP et le déploiement automatisé, l'implémentation d'une telle infrastructure peut être optimisée et sécurisée. Les tests et validations réalisés confirment la viabilité de la solution et soulignent l'importance d'une veille technologique continue pour anticiper les évolutions du marché.

En définitive, une solution VOIP bien conçue et sécurisée constitue un levier stratégique pour les entreprises, leur permettant de moderniser leur communication tout en maîtrisant les coûts et en améliorant l'expérience utilisateur.