

Introduction

Nous avons déjà mis en place précédemment un serveur TOIP mais n'avons jamais abordé la notion d'accès derrière une box FAI ou celle du chiffrement hors TLS via RTSP (chiffrement symétrique ARS)

Dans le cadre de la mise en place du serveur Asterisk, une problématique est apparue liée à l'évolution des versions du logiciel. Les versions récentes (notamment Asterisk 21 et supérieures) ont supprimé le module historique **chan_sip** au profit de **PJSIP**, ce qui rend les anciennes commandes et méthodes de configuration (comme l'utilisation de sip.conf et la commande sip show users) inopérantes. Cette incompatibilité a nécessité un retour vers une version antérieure d'Asterisk (18 LTS) et une compilation manuelle avec activation explicite du module chan_sip via menuselect. Une fois ce module correctement compilé et chargé, il a été possible de retrouver un fonctionnement conforme aux anciennes pratiques, permettant une configuration simplifiée des utilisateurs SIP et des tests via un client softphone.

Au niveau des ports et protocole

La stack protocolaire

Téléphone SIP → RTP (voix) + SIP (signalisation) → Asterisk → Trunk SIP → PSTN

- **SIP** (port 5060 UDP/TCP) : signalisation (INVITE, ACK, BYE...)
- **RTP** (ports 10000-20000 UDP) : flux audio en temps réel
- **SRTP / TLS** : versions chiffrées des deux

Le problème de la box FAI (NAT)

C'est le cœur du défi. La box fait du **NAT** (Network Address Translation), donc :

Internet (IP publique) ↔ Box (NAT) ↔ Asterisk (IP privée 192.168.x.x)

Problèmes typiques :

- Le client SIP externe envoie ses paquets vers ton IP publique, mais la box ne sait pas à qui les transférer
- Asterisk annonce son IP privée dans les headers SIP → le correspondant externe ne peut pas joindre cette IP

Solutions pour traverser le NAT

A) Port Forwarding sur la box

Rediriger les ports entrants vers Asterisk :

Port	Protocole	Usage
5060	UDP	SIP
10000-20000	UDP	RTP (media)
5061	TCP/TLS	SIP sécurisé

B) Configurer Asterisk côté NAT

Dans `sip.conf`:

ini

```
[general]
externip=<ton_IP_publicue> ; IP publique de la box
localnet=192.168.0.0/24 ; ton réseau local
nat=yes ; active la correction NAT
```

Asterisk va alors substituer l'IP privée par l'IP publique dans les headers SIP.

C) IP publique dynamique → DDNS

Les FAI changent souvent l'IP publique. Solution : un service **DDNS** (No-IP, DynDNS...) avec un client sur la box ou un script :

`monasterisk.ddns.net` → mis à jour automatiquement → pointe vers ton IP publique

Ajout TLS

Je créer un clef privée et un certificat qui contiendra une clef publique

Il faut d'abord activer TLS et lui attribuer un port

```
                ; Optionally add a port number
tlsenable=yes          ; Enable server for incoming
tlsbindaddr=0.0.0.0:5061 ; IP address for TLS serv
                ; Optionally add a port number
                ; Remember that the IP address
                ; certificate, so you don't w
```

On créer la clef privé ensuite on genere un certif en se bsant sur cette dernière et ne pas oublier l'autorite de certification

```
openssl req -new -key asterisk.key -out asterisk.csr
```

```
openssl x509 -req -day 3655 -in asterisk.csr -signkey asterisk.key -out asterisk.pem
```

Dans sip.conf

```
tlscertfile=/ssl/asterisk.pem ; Certificate chain (*.pem format only) to use for TLS connections
; The certificates must be sorted starting with the subject's certificate
; and followed by intermediate CA certificates if applicable. If the
; file name ends in _rsa, for example "asterisk_rsa.pem", the files
; "asterisk_dsa.pem" and/or "asterisk_ecc.pem" are loaded
; (certificate, intermediates, private key), to support multiple
; algorithms for server authentication (RSA, DSA, ECDSA). If the chains
; are different, at least OpenSSL 1.0.2 is required.
; Default is to look for "asterisk.pem" in current directory

tlsprivatekey=/ssl/asterisk.pem ; Private key file (*.pem format only) for TLS connections.
; If no tlsprivatekey is specified, tlscertfile is searched for
; for both public and private key.

tlscafile=/ssl/asterisk.crt
;
; If the server your connecting to uses a self signed certificate
; you should have their certificate installed here so the code can
; verify the authenticity of their certificate.
;

;tlscapath=</path/to/ca/dir>
; A directory full of CA certificates. The files must be named with
; the CA subject name hash value.
; (see man SSL_CTX_load_verify_locations for more info)

;tlsdontverifyserver=[yes|no]
; If set to yes, don't verify the servers certificate when acting as
; a client. If you don't have the server's CA certificate you can
; set this and it will connect without requiring tlscafile to be set.
; Default is no.

tlscipher=ALL
; A string specifying which SSL ciphers to use or not use
; A list of valid SSL cipher strings can be found at:
; http://www.openssl.org/docs/apps/ciphers.html#CIPHER_STRINGS

tlsclientmethod=tlsv1 ; values include tlsv1, sslv3, sslv2.
; Specify protocol for outbound client connections.
; If left unspecified, the default is the general-
; purpose version-flexible SSL/TLS method (sslv23).
; With that, the actual protocol version used will
; be negotiated to the highest version mutually
; supported by Asterisk and the remote server, i.e.
; TLSv1.2. The supported protocols are listed at
```

Ensuite

```
chown -R asterisk:asterisk /ssl
```

```
chmod 755 /ssl
```

```
chmod 600 /ssl/asterisk.key /ssl/asterisk.pem
```

```
chmod 644 /ssl/asterisk.crt
```

C'est le même pem pour certificat et key dans le fichier de conf

Et dans console asterisk

Reload

Ensuite vérifier que le service écoute bien sur le port 5061 et dans transport mettre : TLS puis relancer

```
tlsenable=yes ; Enable server for incoming TLS (secure) connections (default is no)
tlsbindaddr=0.0.0.0:5061 ; IP address for TLS server to bind to (0.0.0.0 binds to all interfaces)
; Optionally add a port number, 192.168.1.1:5063 (default is port 5061)
; Remember that the IP address must match the common name (hostname) in the
; certificate, so you don't want to bind a TLS socket to multiple IP addresses.
; For details how to construct a certificate for SIP see
; http://tools.ietf.org/html/draft-ietf-sip-domain-certs

;tcpauthtimeout = 30 ; tcpauthtimeout specifies the maximum number
; of seconds a client has to authenticate. If
; the client does not authenticate before this
; timeout expires, the client will be
; disconnected. (default: 30 seconds)

;tcpauthlimit = 100 ; tcpauthlimit specifies the maximum number of
; unauthenticated sessions that will be allowed
; to connect at any given time. (default: 100)

;websocket_enabled = true ; Set to false to prevent chan_sip from listening to websockets. This
; is needed when using chan_sip and res_pjsip_transport_websockets on
; the same system.

;websocket_write_timeout = 100 ; Default write timeout to set on websocket transports.
; This value may need to be adjusted for connections where
; Asterisk must write a substantial amount of data and the
; receiving clients are slow to process the received information.
; Value is in milliseconds; default is 100 ms.
transport=tls ; Set the default transports. The order determines the primary default transport.
; If tcpenable=no and the transport set is tcp, we will fallback to UDP.

srvlookup=yes ; Enable DNS SRV lookups on outbound calls
; Note: Asterisk only uses the first host
; in SRV records
; Disabling DNS SRV lookups disables the
; ability to place SIP calls based on domain
; names to some other SIP users on the Internet
; Specifying a port in a SIP peer definition or
; when dialing outbound calls will suppress SRV
; lookups for that transport.
```

Je me suis connecté avec user 6000

Pour vérifier si il est bien en TLS j execute cette commande

```
sip show peer 6000
```

Je vois bien ici TLS

```

VM Extension : asterisk
LastMsgsSent : 0/0
Call limit : 0
Max forwards : 0
Dynamic : Yes
Callerid : "Sadek Adel" <>
MaxCallBR : 384 kbps
Expire : 522
Insecure : no
Force rport : Auto (Yes)
Symmetric RTP: No
ACL : No
ContactACL : No
DirectMedACL : No
T.38 support : No
T.38 EC mode : Unknown
T.38 MaxDtgrm: 4294967295
DirectMedia : Yes
PromiscRedir : No
User=Phone : No
Video Support: No
Text Support : No
Ign SDP ver : No
Trust RPID : No
Send RPID : No
Path support : No
Path : N/A
TrustIDOutbnd: Legacy
Subscriptions: Yes
Overlap dial : No
DTMFmode : rfc2833
Timer T1 : 500
Timer B : 32000
ToHost :
Addr->IP : 86.195.60.74:30883
Defaddr->IP : (null)
Prim.Transp. : TLS
Allowed.Trsp : TLS
Def. Username: asadek
SIP Options : (none)
Codecs : (ulaw|alaw|gsm|h263)
Auto-Framing : No
Status : Unmonitored
Useragent : SessionTalk 7.0.4
Reg. Contact : sip:6000@192.168.0.31:34228;rinstance=be1d50bdac1f9985;transport=TLS
Qualify Freq : 60000 ms
Keepalive : 0 ms

```

Chiffrement des appels après signaux SIP

Le chiffrement des appels (la voix elle-même) ne passe pas par TLS mais par **SRTP (Secure RTP)**. Dans Asterisk avec chan_sip, on l'active côté utilisateurs avec encryption=yes (et côté client il faut aussi activer SRTP). SRTP chiffre le flux audio RTP avec des algorithmes standards : en pratique **AES** (souvent AES-128) en mode compteur (**AES-CM**) pour le chiffrement, et **HMAC-SHA1** pour l'authentification/intégrité des paquets. Les clés de session SRTP sont généralement négociées via **SDES dans le SDP**, qui est lui-même protégé par TLS (d'où l'intérêt d'avoir TLS actif). Ainsi, TLS sécurise la signalisation (échanges SIP, mots de passe, clés), et SRTP sécurise le contenu de l'appel (voix chiffrée de bout en bout entre client et serveur).

Je retourne dans le fichier de configuration des utilisateurs

Et je rajoute ses directives

```
[6000]
type=friend
host=dynamic
allow=ulaw
fullname = Sadek Adel
username=asadek
secret=test
context=work

encryption=yes
avpf=yes
force_avp=yes
icesupport=yes
[6001]
type=friend
host=dynamic
allow=ulaw
fullname = Choucheh Sadek
username=choucheh
secret=test
context=work
encryption=yes
avpf=yes
force_avp=yes
icesupport=yes
```

encryption=yes

Active **SRTP** pour ce peer. Le média (voix RTP) est chiffré (AES-CM 128 + HMAC-SHA1). Les clés SRTP sont échangées via **SDES** dans le SDP (donc à utiliser avec **TLS** pour protéger cet échange).

avpf=yes

Active le profil **RTP/AVPF** (RFC 4585) au lieu de RTP/AVP. Ça permet le **feedback RTCP rapide** (utile avec WebRTC et certains clients modernes).

force_avp=yes

Force l'utilisation du **profil AVP/AVPF** même si le client ne le propose pas clairement. Utile pour assurer la compatibilité avec SRTP/clients WebRTC, mais peut casser avec de vieux téléphones SIP.

icesupport=yes

Active **ICE (Interactive Connectivity Establishment)** pour aider à traverser le NAT. Le client propose des candidats (IP/ports) et Asterisk choisit le meilleur chemin média. Utile surtout pour clients mobiles/WebRTC.

En résumé

- encryption=yes → **chiffre la voix (SRTP)**
- avpf=yes / force_avp=yes → **profil RTP moderne + feedback**
- icesupport=yes → **meilleure traversée NAT**

```
--- Including switch 'Lua/' in context 'local'
--- Including switch 'DUNDI/e164' in context 'dundi-e164-switch'
--- Including switch 'DUNDI/e164' in context 'ael-dundi-e164-switch'
--- Time to scan old dialplan and merge leftovers back into the new: 0.000479 sec
--- Time to restore hints and swap in new dialplan: 0.000012 sec
--- Time to delete the old dialplan: 0.000094 sec
--- Total time merge_contexts_delete: 0.000585 sec
--- pbx_lua successfully loaded 51 contexts (enable debug for details).
--- Reloading module 'app_ami.so' (Answering Machine Detection Application)
--- Reloading module 'pbx_config.so' (Text Extension Configuration)
== Setting global variable 'CONSOLE' to 'Console/dsp'
== Setting global variable 'TRUNK' to 'DAHDI/G2'
== Setting global variable 'TRUNKMSD' to '!'
[Apr 25 03:53:42] WARNING[3876462]: pbx_config.c:1955 pbx_load_config: != Unknown directive: static at line 26 of extensions.conf -- IGNORING!!!
[Apr 25 03:53:42] WARNING[3876462]: pbx_config.c:1955 pbx_load_config: != Unknown directive: writeprotect at line 31 of extensions.conf -- IGNORING!!!
[Apr 25 03:53:42] WARNING[3876462]: pbx_config.c:1955 pbx_load_config: != Unknown directive: clearglobalvars at line 93 of extensions.conf -- IGNORING!!!
--- Including switch 'DUNDI/e164' in context 'dundi-e164-switch'
--- Including switch 'DUNDI/e164' in context 'ael-dundi-e164-switch'
--- Including switch 'Lua/' in context 'local'
--- Including switch 'Lua/' in context 'demo'
--- Including switch 'Lua/' in context 'public'
--- Including switch 'Lua/' in context 'default'
--- Time to scan old dialplan and merge leftovers back into the new: 0.000286 sec
--- Time to restore hints and swap in new dialplan: 0.000004 sec
--- Time to delete the old dialplan: 0.000059 sec
--- Total time merge_contexts_delete: 0.000269 sec
--- pbx_config successfully loaded 51 contexts (enable debug for details).
[Apr 25 03:53:42] WARNING[3876462]: pbx.c:8797 ast_context_verify_includes: Context 'local' tries to include nonexistent context 'iaxtel700'
--- Reloading module 'app_miniva.so' (Mini VoiceMail (A minimal Voicemail e-mail System))
--- Reloading module 'res_clialiases.so' (CLI Aliases)
--- Reloading module 'res_http_post.so' (HTTP POST support)
--- Reloading module 'app_voicemail.so' (Comedian Mail (Voicemail System))
--- Reloading module 'app_alarmreceiver.so' (Alarm Receiver for Asterisk)
--- Reloading module 'app_followme.so' (Find-Me/Follow-Me Application)
--- Reloading module 'pbx_ael.so' (Asterisk Extension Language Compiler)
== Setting global variable 'CONSOLE-AEL' to 'Console/dsp'
== Setting global variable 'AXINFO-AEL' to 'guest'
== Setting global variable 'OUTBOUND-TRUNK' to '*zap/g2'
== Setting global variable 'OUTBOUND-TRUNKMSD' to '!'
--- Including switch 'DUNDI/e164' in context 'ael-dundi-e164-switch'
--- Including switch 'Lua/' in context 'default'
--- Including switch 'Lua/' in context 'public'
--- Including switch 'Lua/' in context 'demo'
--- Including switch 'Lua/' in context 'local'
--- Including switch 'DUNDI/e164' in context 'dundi-e164-switch'
--- Time to scan old dialplan and merge leftovers back into the new: 0.000286 sec
--- Time to restore hints and swap in new dialplan: 0.000004 sec
--- Time to delete the old dialplan: 0.000060 sec
--- Total time merge_contexts_delete: 0.000350 sec
--- pbx_ael successfully loaded 51 contexts (enable debug for details).
[Apr 25 03:53:42] WARNING[3876462]: pbx.c:8797 ast_context_verify_includes: Context 'local' tries to include nonexistent context 'iaxtel700'
--- Reloading module 'Func_odbc.so' (ODBC lookups)
--- Reloading module 'app_queue.so' (True Call Queueing)
[Apr 25 03:53:42] NOTICE[3876462]: app_queue.c:9449 reload_queue_rules: queue.rules.conf has not changed since it was last loaded. Not taking any action.
Reloading MGCP
[Apr 25 03:53:42] NOTICE[3876438]: chan_mgcp.c:4695 reload_config: Unable to load config mgcp.conf, MGCP disabled
Reloading SIP
== Using SIP CoS mark 4
== TLS/SSL certificate ok
== Using SIP CoS mark 4
[Apr 25 03:53:46] NOTICE[3876443]: chan_sip.c:80555 sip_poke_noanswer: Peer '6000' is now UNREACHABLE! Last qualify: 0
--- Registered SIP '6000' at 78.240.123.89:11302
[Apr 25 03:54:01] NOTICE[3876602]: chan_sip.c:25089 handle_response_peerpoke: Peer '6000' is now Reachable. (290ms / 2000ms)
> Using SIP RTP CoS mark 5
> 0x7f20f0024d0d --- Strict RTP learning after remote address set to: 172.50.0.105:4016
--- Executing [6000@work:1] Dial("SIP/6001-00000006", "SIP/6000,20") in new stack
== Using SIP RTP CoS mark 5
--- Called SIP/6000
--- SIP/6000-00000007 is ringing
--- SIP/6000-00000007 is ringing
> 0x7f20f00f9900 --- Strict RTP learning after remote address set to: 10.191.93.241:4000
--- SIP/6000-00000007 answered SIP/6001-00000006
--- Channel SIP/6000-00000007 joined 'simple_bridge' basic-bridge <3246593b-908b-41f3-9ec8-1d3695256326>
--- Channel SIP/6001-00000006 joined 'simple_bridge' basic-bridge <3246593b-908b-41f3-9ec8-1d3695256326>
> 0x7f20f00f9900 --- Strict RTP qualifying stream type: audio
> 0x7f20f00f9900 --- Strict RTP switching source address to 78.240.123.89:11302
> 0x7f20f0024d0d --- Strict RTP qualifying stream type: audio
> 0x7f20f0024d0d --- Strict RTP switching source address to 86.195.60.74:6689
> 0x7f20f0024d0d --- Strict RTP learning complete - Locking on source address 86.195.60.74:6689
> 0x7f20f0024d0d --- Strict RTP learning after remote address set to: 172.50.0.105:4016
> 0x7f20f00f9900 --- Strict RTP learning complete - Locking on source address 78.240.123.89:11302
> 0x7f20f0024d0d --- Strict RTP learning complete - Locking on source address 86.195.60.74:6689
--- Channel SIP/6000-00000007 left 'simple_bridge' basic-bridge <3246593b-908b-41f3-9ec8-1d3695256326>
--- Channel SIP/6001-00000006 left 'simple_bridge' basic-bridge <3246593b-908b-41f3-9ec8-1d3695256326>
== Spawn extension (work, 6000, 1) exited non-zero on 'SIP/6001-00000006'
vps-d10640cc=CLI>
```

Sur ce test, nous avons utilisé le logiciel **MicroSIP** sur Windows afin de simuler un client SIP simple compatible avec TLS et SRTP.

La capture montre le journal d'exécution d'Asterisk lors d'un appel entre deux extensions. On y observe que la signalisation est bien établie, les deux clients sont enregistrés et l'appel est accepté (answered), puis les canaux sont reliés dans un

simple_bridge. On voit également les mécanismes de **Strict RTP learning**, où Asterisk apprend dynamiquement les adresses sources des flux audio, ce qui confirme que le trafic média circule.

Cependant, un problème de flux est apparu : les clients tentaient initialement de communiquer **directement entre eux (peer-to-peer)** pour le média RTP. Ce comportement est classique en SIP mais pose problème dans un contexte réel, notamment derrière du NAT ou lorsque certains flux VoIP ne sont pas autorisés ou filtrés.

Pour corriger cela, nous avons appliqué des directives côté Asterisk (directmedia=no, nat=force_rport,comedia, etc.) afin de **forcer le passage du flux audio via le serveur Asterisk**. Cette approche garantit un meilleur contrôle du trafic, une compatibilité accrue avec les environnements réseau contraints et permet d'assurer la continuité du service même lorsque les communications directes entre clients ne sont pas possibles.

Ajout d'un trunk OVH

Je vais me baser sur cette documentation : https://help.ovhcloud.com/csm/fr-voip-creeer-redirection-presentation?id=kb_article_view&sysparm_article=KB0039282

Je dois disposer d'un numéro alias OVH + une ligne SIP ovh cloud

The screenshot shows the OVH VoIP configuration interface. At the top, a progress bar indicates seven steps: 1. Lignes/Téléphones, 2. Panier, 3. Accessoires, 4. Récapitulatif, 5. Contacts, 6. Paiement, and 7. Confirmation. The main content area is titled 'Configurez votre ligne VoIP' and features a 'Choix du forfait' section with three options: 'Découverte' (0,99 €/mois), 'Entreprise' (4,99 €/mois), and 'Entreprise + mobile inclus' (14,99 €/mois). To the right, a 'Résumé de la commande' box shows the selected 'Forfait VoIP Découverte' for 0,99 €/mois, a 'Numéro de téléphone non géographique France', and an 'Option de ligne' for 0,99 €. The total price is 0,99 € ex. TVA, with a note that the next month's price will be 0,99 € after verification.

Je choisis cette offre je vais voir si ça englobe tout le numero alias et la ligne VOIP ou juste la ligne VOIP

Votre ligne VoIP a été ajoutée à votre panier

Forfait : Forfait VoIP Découverte
Numéro : Numéro non géographique France
Option mobile : Appels vers les mobiles à la seconde

Quantité :

Configurez une autre ligne

Récapitulatif de votre commande

Cette page est le récapitulatif détaillé de votre commande, il ne s'agit pas de votre facture.

Prix ex. TVA (EUR)

Forfait VoIP Découverte 1 ligne		0,99 € /mois
Numéro de téléphone non géographique France		
Option de ligne Appels vers les mobiles à la seconde		
	Total ex. TVA Inclus dès le premier mois d'utilisation	0,99 €
	20% TVA	0,20 €
	Total TTC	1,19 €

Tableau de bord Bare Metal Cloud Hosted Private Cloud Public Cloud Web Cloud **Télécom** Sunrise Marketplace

Navigation classique Navigation beta Découvrez notre nouveau Manager Beta Français Adel Sadek

Commander

- Accès Internet
- Téléphonie
 - Gérer mes reversements
 - sa1660325-ovh-1
 - SIP 0033972124733**
- SMS
- Fax
- OverTheBox
- Opérations

VOIP / sa1660325-ovh-1 / Lignes / 0033972124733 / Gestion des appels / Renvoi d'appel

< Retour à la gestion des appels

Renvoi d'appel

Configurez un renvoi de vos appels entrants sous diverses conditions.
Certains opérateurs filtrent les appels redirigés. Il est donc possible que la ligne distante ne réceptionne pas les appels redirigés.

Configuration

Renvoi de tous les appels

Renvoyer vers un Répondeur

au numéro 0033972124733

Renvoi quand il n'y a pas de réponse

avant 25 seconde(s)

Renvoyer vers un

Je ne peux pas configurer maintenant car il faut que je prenne un numero d'alias

<https://www.ovhcloud.com/fr/phone/numeros/>

Je re appuie sur commander ici

Numéros Fixes

Choisissez le numéro qui vous convient

Numéro fixe SDA Redirection

0,20 €

HT/mois
soit 0,24 € TTC/mois

Commander

- ✓ Zone géographique ou non
- ✓ 01, 02, 03, 04 et 05 ou 09
- ✓ Portabilité offerte
- ✓ Frais d'installation offerts
- ✓ Aucune période d'engagement
- ✓ Numéros non consécutifs pour des redirections d'appels uniquement

Dans la limite de 1000 numéros par client et par mois

Types de numéros

Numéros géographiques (01, 02, ...)

Associez votre activité à un numéro fixe

À partir de : 1.00 €

HT / mois
(soit 1.20 € TTC)



Numéros non-géographiques (09)

Associez votre activité à un numéro fixe

À partir de : 1.00 €

HT / mois
(soit 1.20 € TTC)



Numéros à valeur ajoutée (08)

Soyez joignable grâce à un numéro à tarification spéciale

À partir de : 1.00 €

HT / mois
(soit 1.20 € TTC)

Vérifier l'identité

Commander un numéro international (+32, +44, ...)

À partir de : 1.00 €

HT / mois



[Retour au choix des types de numéros](#)

Commander un numéro géographique (fr)

Nombre de numéros :

Nombre de numéros :

1 numéro

Zone géographique :

Sarcelles

Choix du numéro

Numéro standard

0033185430247

1.00 € HT / mois

Numéro facile à retenir

0033185430043

5.00 € HT / mois

Coordonnées du titulaire du numéro

Commande traité je patiente un peu

Service	Description	Type	Type de service
0033185430247		Alias	Aucun type
0033972124733		Ligne	Sip

10 sur 2 résultats

Je configure cette ligne

Configuration du numéro

À quoi sert la configuration du numéro ? 

Redirection d'appels

La redirection d'appels vous permet de rediriger les appels reçus vers la ligne ou numéro de votre choix.



File d'appels

La file d'appels vous permet de rediriger un appel entrant vers plusieurs lignes, rattachées ou non à votre identifiant OVHcloud.



Ensuite en bas cliquer paramétrer

0033185430247 

0033185430247

Mon numéro

Configuration 

Coordonnées

Configuration : Redirection d'appels

Qu'est-ce que la redirec

Choix du service

Rechercher 

Sa1660325-Ovh-1


0033972124733

Lignes SIP

Vos appels entrants

Ligne vers laquelle vos appels

Aucun

 Sélectionner une ligne

Veillez noter que la législation interd

Pour faire une redirection vers

Ensuite je valide

Vos appels entrants

Ligne vers laquelle vos appels seront redirigés

0033972124733

 Sélectionner une ligne

Veillez noter que la législation interdit les redirections vers les numéros surtaxés.

Pour faire une redirection vers une ligne externe, il vous faut configurer votre numéro en [File d'appels](#)

Vos appels sortants (facultatif)

Activer la présentation de votre numéro lors des appels sortants


Attention : si vous avez paramétré la présentation d'un autre numéro sur la ligne, il sera remplacé par ce numéro.

Ensuite il faut acheter un SIP TRUNK

<https://www.ovhcloud.com/fr/phone/sip-trunk/>



- Connectez-vous à votre espace client OVHcloud : <https://www.ovhtelecom.fr/espaceclient/>
- Cliquez sur le lien "**Accéder à l'ancienne interface**".
- Cliquez sur l'icône "**Téléphonie**".
- Cliquez sur votre trunk.
- Cliquez sur "**Téléphone**" dans le menu "**Navigation**".
- Cliquez sur "**Codecs**".
- Cliquez sur "**Gérer**".
- Cochez la case "**Amélioration de la présentation du numéro appelé**".
- Cliquez sur "**Valider**" pour confirmer la configuration.


VoIP / sa1660325-ovh-1 / Lignes / 0033972124831 / Téléphone

0033972124831 

0033972124831

 Guides

 Gestion des musiques Répondeur Carnets de contacts Téléphone Assistance Coordonnées 

 Touches programmables

Informations générales >	Codecs >	Paramètres Plug & Phone personnalisés
Redémarrer	Commander un téléphone VoIP >	Commander des accessoires >
Rattacher la ligne à un équipement actuel >		

On met ce codec pour améliorer la compatibilité

Codecs définis actuellement :

g711, g729

Modifier les codecs

Choisissez un ordre de priorité pour vos codecs :

- g711
- g711, i
- g729, g711
- g711, g729
- g729, g711, i
- g711, g729, i
- g722, g729, g711
- g722, g711, g729
- g722, g729, g711, i
- g722, g711, g729, i

[Appliquer à plusieurs lignes](#)

[Modifier les codecs](#)


Ce rendre ensuite dans le fichier **sip.conf**

- Pour copier votre fichier : **cp /etc/asterisk/sip.conf /etc/asterisk/sip.conf.bak**
- Pour ouvrir votre fichier sip.conf : **vim /etc/asterisk/sip.conf**

On utilise un nouveau fichier vierge


Je modifie le mot de passe SIP ici

VoIP / sa1660325-ovh-1 / Lignes / 0033972124831

0033972124831 

0033972124831

 **Gestion** [Consommation](#) [Gestion des appel](#)

 Informations générales

- Mot de passe SIP >
- Domaine SIP
- Gérer la langue >
- Changer d'offre
- Commander un casque >
- Résiliation de la

Nos conseils

```
[general]
defaultexpiry=1800 ; Temps de register de la ligne.
context=trunk-ovh ; Nom du context pour le trunk dans sip.conf
bindport=5060 ; Port d'ecoute.
bindaddr=0.0.0.0 ; Adresse d'ecoute.
srvlookup=no ; Autoriser les appels via noms DNS
register => 0033972124831: @siptrunk.ovh.net ; Authentfication du trunk. La syntaxe est username:p
disallow=all ; Gestion des codecs pour autoriser que le G7111
allow=ulaw ; Gestion des codecs pour autoriser que le G7111
allow=alaw ; Gestion des codecs pour autoriser que le G7111

[trunk-ovh]
type=friend ; Definit le type d'appels : peer = appels sortants / user = appels entrants / friend = les d
host=siptrunk.ovh.net ; Nom du serveur SIP du trunk.
context=ovh-sip ; Nom du contexte pour le trunk dans extensions.conf et gérer les appels entrants.
language=fr ; Langue de la ligne.
insecure=invite,port
username=0033972124831 ; Username du trunk.
secret= ; Mot de passe du trunk.
```

User.conf

```

Invite de commandes - ssh rc x + v
GNU nano 5.4 ./users.conf *
qualify=yes
qualifyfreq=15
keepalive=15

[330]
username=330 ; Username pour l'auth.
type=friend ; Definit le type d'appels : peer = appels sortants / user = appels entrants / friend = les deux
secret=test ; Mot de passe de l'extension.
callerid="0033366725520" <0033366725520> ; Numero du DDI à présenter
nat=yes ; L'extension est utilisee derriere un routeur utilisant le NAT.
host=dynamic ; L'extension s'enregistre elle meme.
context=sortant-ovh ; Context a utiliser qui sera definit dans extensions.
conf language=fr ; Langue de l'extension.

[520]
username=520
type=friend
secret=test
callerid="0033185450330" <0033185450330>
nat=yes
host=dynamic
context=sortant-ovh
language=fr

```

Extension.conf

```

GNU nano 5.4 extensions.conf
;
; The "General" category is for certain variables.
;
[general]

[sortant-ovh] ; Si un appel arrive sur 330 => Ca fait sonner l'extension 330.
exten => 330,1,Dial(SIP/330,10,tr)
exten => 330,2,HangUp() ; Si un appel arrive sur 520 => Ca fait sonner l'extension 520.
exten => 520,1,Dial(SIP/520,10,tr)
exten => 520,2,HangUp() ; Sortir avec le trunk. On autorise que les appels sur les 01 > 07 et 09.
exten => _0[1-7]XXXXXXXX,1,Dial(SIP/${EXTEN}@trunk-ovh)
exten => _09XXXXXXXX,1,Dial(SIP/${EXTEN}@trunk-ovh)

[ovh-sip] ;Redirection de l'alias 0366725520 vers l'extension 520.
exten => 0366725520,1,Ringing(1)
exten => 0366725520,2,Dial(SIP/520,10,tm) ;Redirection de l'alias 0185450330 vers l'extension 330.
exten => 0185450330,1,Ringing(1)
exten => 0185450330,2,Dial(SIP/330,10,tm)
exten => s,1,Ringing(1) ; Attendre une seconde en faisant retentir la sonnerie du telephone de l'apellant
exten => s,2,Dial(SIP/330,25,tm) ; L'appel est transfere sur le poste 330. Sans reponse apres 25 secondes il passe a l'
exten => s,3,Hangup(16) ; La communication est termine

[work]
exten => _6XXX,1,DIAL(SIP/${EXTEN},20)
exten => _6XXX,2,Hangup()
;

```

Ensuite je redemarre asterisk

Pour afficher tout les trunk

sip show peers

```

ected because extension not found in context 'trunk-ovh'.
/ps-d10640cc*CLI> sip show peers
Name/username      Host                Dyn Forcerport Comedia  ACL Port  Status  Description
330/330            (Unspecified)      D Yes      Yes      0         Unmonitored
520/520            (Unspecified)      D Yes      Yes      0         Unmonitored
000/asadek         (Unspecified)      D Yes      Yes      0         UNKNOWN
001/choucheh       88.168.80.218      D Yes      Yes      36568    UNREACHABLE
003/ayman          (Unspecified)      D Yes      Yes      0         UNKNOWN
trunk-ovh/0033972124831  91.121.129.23    Auto (No)  No      5060     Unmonitored
5 sip peers [Monitored: 0 online, 3 offline Unmonitored: 1 online, 2 offline]
== Using SIP RTP CoS mark 5

```

Voici un bloc de doc sérieux que tu peux intégrer directement.

Configuration du trunk SIP OVH sur Asterisk avec chan_sip

Dans notre cas, la ligne OVH SIP Trunk fournit les informations suivantes :

Numéro / Login SIP : 0033972124831

Authorization user name : 0033972124831

Domain / Registrar : sip-domain.io

Proxy sortant : sa1660325-ovh-1.sip-proxy.io

IP résolue du proxy : 137.74.239.8

IP publique du VPS Asterisk : 144.217.14.63

Le problème rencontré était que le domaine sip-domain.io ne se résolvait pas correctement depuis le VPS. Asterisk essayait donc d'envoyer le REGISTER vers une destination nulle, ce qui produisait des erreurs de type :

```
getaddrinfo("sip-domain.io"): No address associated with hostname
```

```
sip_xmit ... to (null) returned -1: Invalid argument
```

En testant directement le proxy sortant OVH, on constatait qu'il répondait bien :

```
nslookup sa1660325-ovh-1.sip-proxy.io
```

Résultat :

```
sa1660325-ovh-1.sip-proxy.io canonical name = asbc.n11.prod.nowi.ovh
```

```
Address: 137.74.239.8
```

Cependant, enregistrer directement le trunk sur le proxy sortant provoquait une erreur :

```
SIP/2.0 404 Domain not bound
```

Cela signifie que le proxy est bien le point d'entrée réseau, mais que le domaine SIP attendu par OVH reste sip-domain.io.

La solution consiste donc à forcer localement la résolution de sip-domain.io vers l'IP du proxy OVH :

```
echo "137.74.239.8 sip-domain.io" >> /etc/hosts
```

Ainsi, Asterisk continue d'utiliser le bon domaine SIP sip-domain.io, tout en envoyant réellement les paquets vers l'infrastructure OVH.

Fichier /etc/asterisk/sip.conf

[general]

context=default

bindport=5060

bindaddr=0.0.0.0

defaultexpiry=1800

srvlookup=yes

; NAT / IP publique du serveur Asterisk

externip=144.217.14.63

localnet=127.0.0.0/255.0.0.0

; Sécurité de base

allowguest=no

alwaysauthreject=yes

; Codecs autorisés

disallow=all

allow=alaw

allow=ulaw

; Enregistrement SIP OVH

; Format :

; register => login:motdepasse@registrar/numero

register => 0033972124831:votremotdepasse@sip-domain.io/0033972124831

[trunk-ovh]

type=peer

; Proxy OVH utilisé comme point d'entrée réseau

host=sa1660325-ovh-1.sip-proxy.io

; Domaine SIP attendu par OVH pour l'identité de la ligne

fromdomain=sip-domain.io

; Identifiants SIP OVH

username=0033972124831

fromuser=0033972124831

authuser=0033972124831

secret=votremotdepasse

; Contexte utilisé pour router les appels entrants dans extensions.conf

context=ovh-sip

language=fr

; Options SIP

insecure=invite,port

qualify=yes

nat=force_rport,comedia

; Codecs du trunk

disallow=all

allow=alaw

allow=ulaw

Vérification de l'enregistrement

Après modification du fichier :

```
asterisk -rx "sip reload"
```

```
asterisk -rx "sip show registry"
```

Le résultat attendu est :

```
Host      Username  Refresh State  Reg.Time
sip-domain.io  0033972124831 1800  Registered
```

Dans le debug SIP, la confirmation correcte apparaît sous cette forme :

```
SIP/2.0 200 OK
```

```
Outbound Registration: Expiry for sip-domain.io is 1800 sec
```

Explication du problème

Au départ, plusieurs erreurs pouvaient faire croire à un problème d'identifiants ou de configuration Asterisk. En réalité, il y avait deux points distincts :

1. sip-domain.io ne se résolvait pas correctement depuis le VPS.
2. Le proxy OVH sa1660325-ovh-1.sip-proxy.io répondait bien, mais ne devait pas remplacer le domaine SIP dans le REGISTER.

Quand le REGISTER était envoyé directement vers :

```
sa1660325-ovh-1.sip-proxy.io
```

OVH répondait :

```
404 Domain not bound
```

Cela indiquait que le domaine utilisé dans le REGISTER n'était pas celui attendu par l'infrastructure OVH.

La bonne logique est donc :

Domaine SIP / Registrar : sip-domain.io

Proxy réseau réel : sa1660325-ovh-1.sip-proxy.io

Comme sip-domain.io ne se résolvait pas, on l'a associé localement à l'IP du proxy OVH dans /etc/hosts.

Validation finale

Après correction du mot de passe SIP et de la résolution locale, OVH a répondu :

SIP/2.0 401 Unauthorized

Cette réponse est normale : le serveur demande une authentification SIP Digest.

Asterisk a ensuite renvoyé un REGISTER avec l'en-tête Authorization, puis OVH a répondu :

SIP/2.0 200 OK

Ce 200 OK confirme que le trunk SIP OVH est correctement enregistré sur Asterisk.

```
[Apr 26 22:08:05] NOTICE[4083736][C-000000cb]: chan_sip.c:19648 send_check_user_failure_response: Failed to authenticate device <sip:29629@144.217.14.63>;ta
p:665252160 for INVITE, code = -1
vps-d10640cc*CLI> sip show registry
Host                               dnsmgr Username           Refresh State           Reg.Time
sip-domain.io:5060                 N           003397212483           1785 Registered       Sun, 26 Apr 2026 22:04:01
1 SIP registrations.
vps-d10640cc*CLI>
```

Fichier extension.conf

```
[sortant-ovh]
; Appels internes
exten => 330,1,Dial(SIP/330,10,tr)
same => n, Hangup()

exten => 520,1,Dial(SIP/520,10,tr)
same => n, Hangup()

; Sortie via trunk OVH
; Autorise 01 à 07
exten => _0[1-7]XXXXXXXX,1,NoOp(Appel sortant OVH vers ${EXTEN})
same => n,Dial(SIP/trunk-ovh/${EXTEN},60)
same => n, Hangup()

; Autorise 09
exten => _09XXXXXXXX,1,NoOp(Appel sortant OVH vers ${EXTEN})
same => n,Dial(SIP/trunk-ovh/${EXTEN},60)
same => n, Hangup()

[ovh-sip]
; Appel entrant sur le numéro OVH au format international
exten => 0033972124831,1,NoOp(Appel entrant OVH 0033972124831)
same => n, Ringing()
same => n,Dial(SIP/520,25,tm)
same => n, Hangup()

; Appel entrant sur le numéro OVH au format national
exten => 0972124831,1,Goto(ovh-sip,0033972124831,1)

; Fallback si OVH envoie l'appel sur "s"
exten => s,1,NoOp(Appel entrant OVH sans DID)
same => n, Ringing()
same => n,Dial(SIP/520,25,tm)
same => n, Hangup()
```

J'ai reload le dialplan et je l'affiche pour voir si la cofn a bien été prise en compte

```
root@vps-d10640cc:/etc/asterisk# asterisk -rx "dialplan reload"
asterisk -rx "dialplan show ovh-sip"
asterisk -rx "dialplan show sortant-ovh"
Dialplan reloaded.
[ Context 'ovh-sip' created by 'pbx_config' ]
'0033972124831' => 1. NoOp(Appel entrant OVH 0033972124831) [extensions.conf:37]
                  2. Ringing() [extensions.conf:38]
                  3. Dial(SIP/520,25,tm) [extensions.conf:39]
                  4. Hangup() [extensions.conf:40]
'0972124831' => 1. Goto(ovh-sip,0033972124831,1) [extensions.conf:43]
's' => 1. NoOp(Appel entrant OVH sans DID) [extensions.conf:46]
        2. Ringing() [extensions.conf:47]
        3. Dial(SIP/520,25,tm) [extensions.conf:48]
        4. Hangup() [extensions.conf:49]

-- 3 extensions (9 priorities) in 1 context. --
[ Context 'sortant-ovh' created by 'pbx_config' ]
'330' => 1. Dial(SIP/330,10,tr) [extensions.conf:17]
        2. Hangup() [extensions.conf:18]
'520' => 1. Dial(SIP/520,10,tr) [extensions.conf:20]
        2. Hangup() [extensions.conf:21]
'_09XXXXXXX' => 1. NoOp(Appel sortant OVH vers ${EXTEN}) [extensions.conf:30]
                2. Dial(SIP/trunk-ovh/${EXTEN},60) [extensions.conf:31]
                3. Hangup() [extensions.conf:32]
'_0[1-7]XXXXXXX' => 1. NoOp(Appel sortant OVH vers ${EXTEN}) [extensions.conf:25]
                   2. Dial(SIP/trunk-ovh/${EXTEN},60) [extensions.conf:26]
                   3. Hangup() [extensions.conf:27]

-- 4 extensions (10 priorities) in 1 context. --
root@vps-d10640cc:/etc/asterisk#
```

Parfait, là on va faire une **suite propre, claire et exploitable** pour ta doc, avec **explication du dialplan + ce qu'on a corrigé**.

Suite – Routage des appels (Dialplan Asterisk)

Introduction

Une fois le trunk SIP OVH correctement enregistré, la prochaine étape consiste à **router les appels**.

Dans Asterisk, cela se fait via le fichier :

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

Ce fichier contient ce qu'on appelle le **dialplan**.

Le dialplan = **la logique du serveur téléphonique**

- Qui appelle qui
- Comment sortir vers l'extérieur
- Comment gérer les appels entrants

Structure du dialplan

Un dialplan est organisé en **contextes** :

[sortant-ovh]

[ovh-sip]

Chaque contexte correspond à :

- un type d'appel
 - ou une source (interne / externe)
-

1. Contexte des appels sortants

[sortant-ovh]

; Appels internes

exten => 330,1,Dial(SIP/330,10,tr)

same => n,Hangup()

exten => 520,1,Dial(SIP/520,10,tr)

same => n,Hangup()

; Appels vers l'extérieur (01 à 07)

exten => _0[1-7]XXXXXXXX,1,NoOp(Appel sortant OVH vers \${EXTEN})

same => n,Dial(SIP/\${EXTEN}@sip-domain.io,60)

same => n,Hangup()

; Appels vers les numéros en 09

exten => _09XXXXXXXX,1,NoOp(Appel sortant OVH vers \${EXTEN})

same => n,Dial(SIP/\${EXTEN}@sip-domain.io,60)

same => n,Hangup()

Explication détaillée

Appels internes

exten => 330,1,Dial(SIP/330,10,tr)

Si quelqu'un compose **330**

- Asterisk appelle l'extension SIP 330
- pendant 10 secondes

Pattern des numéros

_0[1-7]XXXXXXXX

Ça veut dire :

- commence par 0
- suivi de 1 à 7
- puis 8 chiffres

✓ Exemples :

- 0123456789
- 0699850123

Variable **`\${EXTEN}`**

`\${EXTEN}`

C'est le numéro composé

Exemple :

Dial(SIP/**`\${EXTEN}`**@sip-domain.io)

 devient :

Dial(SIP/0699860123@sip-domain.io)

Pourquoi on a changé ça ?

Avant (erreur)

Dial(SIP/**`\${EXTEN}`**@trunk-ovh)

Résultat :

400 Wrong domain

👉 OVH refusait car :

- mauvais domaine SIP
-

Après (corrigé)

Dial(SIP/\${EXTEN}@sip-domain.io)

Là :

- domaine correct
 - authentification OK
 - appel accepté
-

2. Contexte des appels entrants OVH

[ovh-sip]

; Appel entrant (format international)

exten => 0033972124831,1,NoOp(Appel entrant OVH)

same => n, Ringing()

same => n, Dial(SIP/520,25,tm)

same => n, Hangup()

; Format national

exten => 0972124831,1,Goto(ovh-sip,0033972124831,1)

; Fallback

exten => s,1,NoOp(Appel entrant OVH sans DID)

same => n, Ringing()

same => n, Dial(SIP/520,25,tm)

same => n, Hangup()

Explication

Cas 1 : OVH envoie ton numéro complet

0033972124831

👉 Asterisk :

- fait sonner
 - envoie vers extension 520
-

Cas 2 : OVH envoie en local

0972124831

On redirige vers le format international :

Goto(...)

Cas 3 : OVH envoie "s"

👉 Ça arrive souvent

exten => s,1,...

On met un fallback

3. Ce qu'on a corrigé (important pour la doc)

1. Mauvais domaine SIP

Erreur :

400 Wrong domain

✓ Correction :

fromdomain=sip-domain.io

Dial(...@sip-domain.io)

2. Mauvais REGISTER

Erreur :

404 Domain not bound

✓ Correction :

- garder sip-domain.io
 - forcer DNS via /etc/hosts
-

3. Mauvais mot de passe

Erreur :

403 Wrong login or password

✓ Correction :

- mot de passe corrigé
-

4. Authentification SIP

407 authentication required

NORMAL

Asterisk :

- reçoit 407
 - renvoie avec auth
 - appel passe
-

5. Résultat final

183 Progress

180 Ringing

Signifie :

- appel accepté
 - en cours
 - ça sonne
-

4. Validation du dialplan

Commande :

```
asterisk -rx "dialplan show"
```

Permet de vérifier que :

- le contexte existe
 - les règles sont chargées
-

Conclusion

On a mis en place :

- ✓ un serveur Asterisk fonctionnel
 - ✓ un trunk OVH opérationnel
 - ✓ un routage sortant fonctionnel
 - ✓ un routage entrant prêt
-

Étape suivante (logique)

Tester :

1. appel entrant vers ton 09
2. vérifier que ça sonne extension 520
3. sinon debug :

```
asterisk -rvvv
```

```
sip set debug on
```

Si tu veux, prochaine étape :

sécuriser ton serveur (fail2ban + firewall SIP)

ou connecter ton téléphone proprement (Zoiper / MicroSIP / iOS)

Appel depuis mon téléphone vers 520



Gestion de la présentation du numéro (Caller ID) avec OVH

Dans le cadre de l'utilisation d'un trunk SIP OVH, la gestion du **Caller ID** (numéro présenté lors des appels sortants) obéit à certaines contraintes imposées par l'opérateur. Contrairement à une configuration purement locale Asterisk, il n'est pas possible d'afficher arbitrairement n'importe quel numéro. OVH n'autorise que la présentation de numéros **explicitement rattachés au trunk**, à savoir le numéro principal ou des numéros alias préalablement configurés dans l'espace client.

Pour mettre en place cette fonctionnalité, il est nécessaire d'acheter un numéro supplémentaire (alias) et de l'associer au trunk SIP OVH. Une fois cette association réalisée et les options de présentation activées côté OVH, il devient possible d'utiliser ce numéro comme identifiant sortant.

Côté Asterisk, la configuration peut se faire soit dans le fichier des utilisateurs via la directive callerid, soit de manière plus robuste directement dans le dialplan. L'utilisation du dialplan permet de contrôler dynamiquement le numéro présenté selon le contexte d'appel, indépendamment de la configuration des postes SIP.

Exemple dans le dialplan :

```
exten => _0[1-9]XXXXXXXX,1,NoOp(Appel sortant)
```

same => n,Set(CALLERID(num)=0972124831)

same => n,Dial(SIP/\${EXTEN}@sip-domain.io)

Dans cet exemple, tous les appels sortants utiliseront le numéro 0972124831 comme identifiant. Il est également possible de définir plusieurs règles pour utiliser différents numéros selon les extensions ou les types d'appels.

En résumé, la gestion du Caller ID avec OVH repose sur une double configuration :

- côté opérateur, en déclarant les numéros autorisés,
- côté Asterisk, en définissant dynamiquement le numéro présenté dans le dialplan.

Toute tentative d'utilisation d'un numéro non autorisé peut entraîner un refus d'appel ou une substitution automatique par le numéro principal du trunk.

Schema

Client SIP (TLS + SRTP)

↓

Internet

↓

Box NAT

↓

Asterisk (VPS)

↓

Trunk OVH

↓

PSTN

Conclusion

Cette architecture permet de sécuriser les communications VoIP tout en assurant une compatibilité avec les contraintes réseau (NAT, IP dynamique). Toutefois, elle nécessite un durcissement supplémentaire pour un usage en production.