



lycée d'enseignement
général et technologique 
Ozenne
Haute-Garonne
académie
Toulouse

COMPTE RENDU TP TOIP INSTALLATION D'UN SERVEUR ASTERISK



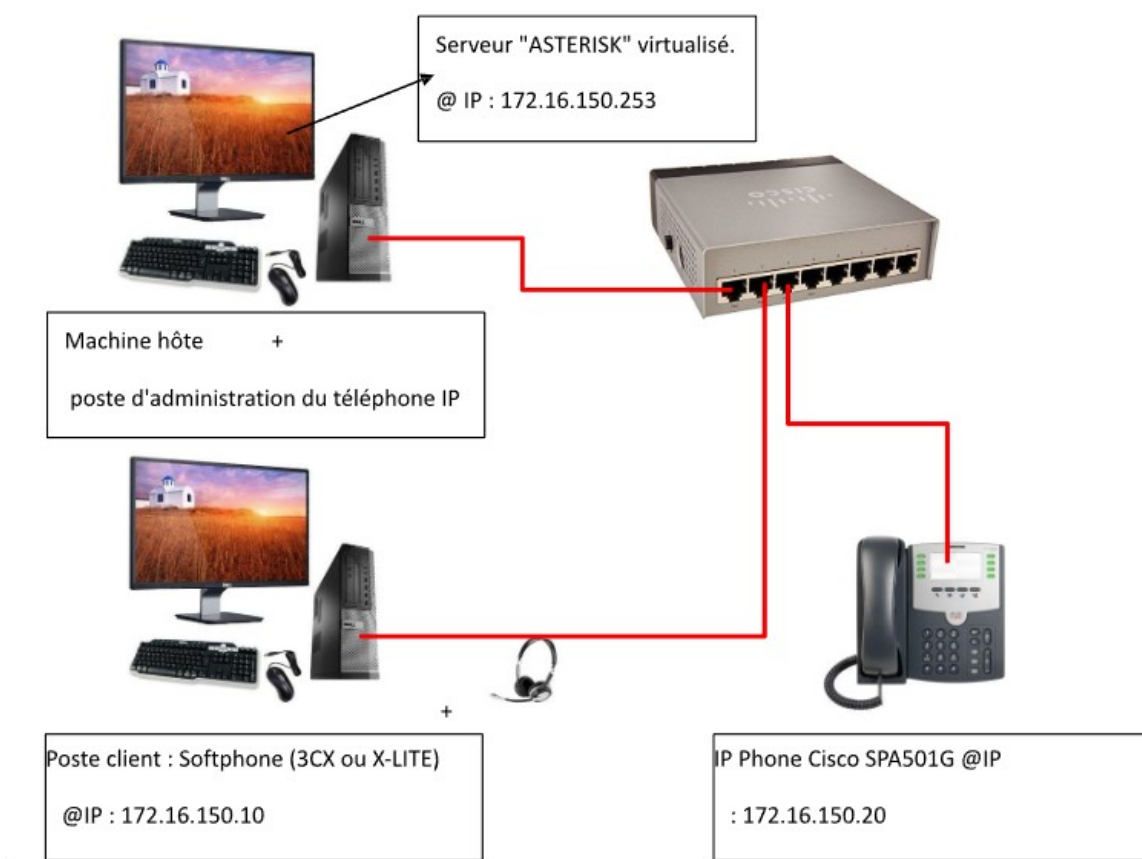
Sommaire

Contexte.....	3
Architecture.....	3
Préparation de la VM.....	4
Procédure d'installation serveur asterisk.....	5
Configuration PJSIP.....	6
Vérification et Tests.....	9
Conclusion.....	11

Contexte

Ce compte rendu présente l'installation et la configuration d'un téléphone privé sur IP (IPBX) basé sur un serveur Asterisk (version 18). L'objectif est de fournir des services de téléphonie de base tels que enregistrement des postes utilisateurs, appels internes et messagerie vocale.

Architecture



Préparation de la VM

Système d'exploitation : Debian 10

Éditer le fichier interfaces réseau :

```
nano /etc/network/interfaces
```

L'interface de loopback

```
auto lo iface lo inet loopback
```

L'interface principale

```
auto ens18  
iface ens18 inet static  
address 172.16.149.253  
netmask 255.255.0.0  
gateway 172.16.0.1  
dns-server 172.16.0.100  
dns-domain sio.local
```

Appliquer les changements :

```
systemctl restart networking
```

Installation des dépendances

```
apt-get update apt update && sudo apt install -y build-essential wget openssl libssl-  
dev libxml2-dev libncurses5-dev uuid-dev sqlite3 libsqlite3-dev pkg-config libjansson-  
dev libsrtp2-dev libedit-dev
```

Procédure d'installation serveur asterisk

Installation de la version 18 d'asterisk

```
cd /usr/src
```

```
sudo wget https://downloads.asterisk.org/pub/telephony/certified-asterisk/asterisk-certified-18.9-cert12.tar.gz
```

Décompression du fichier

```
sudo tar xvfz asterisk-certified-18.9-cert12.tar.gz
cd asterisk-certified-18.9-cert12
```

Configuration avec pjproject embarqué (utilisation de pjp et non sip) :

```
sudo contrib/scripts/install_prereq install
sudo ./configure --with-pjproject-bundled
```

Compilation et installation

```
make
make install
make samples
make config
```

Démarrage du service

```
sudo systemctl start asterisk
```

```
root@debian12:~# /etc/init.d/asterisk status
• asterisk.service - LSB: Asterisk PBX
  Loaded: loaded (/etc/init.d/asterisk; generated)
  Active: active (running) since Mon 2025-12-08 15:52:09 CET; 41min ago
    Docs: man:systemd-sysv-generator(8)
  Process: 427 ExecStart=/etc/init.d/asterisk start (code=exited, status=0/SUCCESS)
    Tasks: 56 (limit: 2306)
  Memory: 117.4M
     CPU: 42.881s
  CGroup: /system.slice/asterisk.service
          └─452 /usr/sbin/asterisk

déc. 08 15:52:08 debian12 systemd[1]: Starting asterisk.service - LSB: Asterisk PBX...
déc. 08 15:52:09 debian12 asterisk[427]: Starting Asterisk PBX: asterisk.
déc. 08 15:52:09 debian12 systemd[1]: Started asterisk.service - LSB: Asterisk PBX.
root@debian12:~# _
```

Configuration PJSIP

Configuration du channel driver PJSIP avec définition du transport UDP et création des utilisateurs Michel (6007) et Yves (6006).

Fichier : [/etc/asterisk/pjsip.conf](#)

```
; --- CONFIGURATION DU TRANSPORT ---
```

```
[transport-udp]
type=transport
protocol=udp
bind=0.0.0.0:5060
```

```
; --- UTILISATEUR 6007 (michel) ---
```

```
[6007]
type=endpoint
context=ozenne
disallow=all
allow=ulaw
auth=6007-auth
aors=6007-aor
```

```
[6007-auth]
type=auth
auth_type=userpass
username=6007
password=1234
```

```
[6007-aor]
type=aor
max_contacts=1
```

```
; --- UTILISATEUR 6006 (yves) ---
```

```
[6006]
type=endpoint
context=ozenne
disallow=all
allow=ulaw
auth=6006-auth
aors=6006-aor
```

```
[6006-auth]
type=auth
auth_type=userpass
username=6006
password=1234
```

```
[6006-aor]
type=aor
max_contacts=1
```

Mise en place de la logique d'appel dans le contexte [ozenne]. Des variables globales sont utilisées pour mapper les numéros de téléphone aux noms des endpoints PJSIP.

Fichier : [/etc/asterisk/extensions.conf](#)

```
[general]
static=yes
writeprotect=no
clearglobalvars=no

[globals]
; Variables pour identifier les utilisateurs par leur nom (optionnel)
SIP_6007=michel
SIP_6006=yves

[ozenne]
; --- Appel vers les numéros 6XXX ---
; On appelle directement le compte PJSIP correspondant au numéro composé (EXTEN)
exten => _6XXX,1,Dial(PJSIP/${EXTEN},15)

; Si pas de réponse (après 15s) ou occupé, direction la messagerie
exten => _6XXX,n,VoiceMail(${EXTEN}@ozenne)

; Raccrochage propre
exten => _6XXX,n,Hangup()

; --- Accès à la messagerie vocale ---
; Le 600 permet de consulter sa propre boîte vocale
exten => 600,1,VoiceMailMain(${CALLERID(num)}@ozenne)
exten => 600,n,Hangup()
```

Création des boîtes vocales pour les utilisateurs.

Fichier : [/etc/asterisk/voicemail.conf](#)

```
[ozenne]
; Syntaxe : numéro => mot_de_passe, nom_utilisateur, email (optionnel)
6007 => 1234,Michel
6006 => 1234,Yves
```

Vérification et Tests

Une fois les fichiers modifiés, il faut recharger Asterisk si possible sans interrompre les appels établis, ou redémarrer le service

Connexion à la console Asterisk

```
/usr/sbin/asterisk -rvvvv
```

Dans la console CLI Asterisk

```
core reload  
pjsip reload  
dialplan reload
```

Configuration des Softphones \$

Afin d'effectuer les tests, il est nécessaire de configurer deux softphones en utilisant les paramètres suivants :

Paramètre	User 1	User 2
Serveur	172.16.150.253	172.16.150.253
Username	michel	yves
Mot de passe	1234	1234
ID	6007	6006

Tests : Michel (6007) appelle Yves (6006)



Conclusion

La mise en place de ce serveur Asterisk est un succès : les utilisateurs Michel et Yves disposent d'un système de téléphonie fonctionnel, incluant les appels directs et une messagerie vocale personnalisée. Ce TP me démontre qu'une solution Open Source permet de créer un standard téléphonique complet, gratuit et entièrement sur mesure.