

Servidor VoIP Asterisk

Configuración de sistemas de telefonía IP

Laboratorio de Redes, Sistemas y Servicios, 3ºD GIT

Alberto Monterroso Barco
Luis Pastor Camarillo

Contenido

Introducción	2
Instalación	2
Procedemos a configurar e instalar Asterisk.....	3
Configuración de la centralita	5
PJSIP.CONF	5
EXTENSIONS.CONF	6
Voicemail.....	6
Audios en español	7
Menú IVR.....	7
Base de datos	8
Agentes.....	10
Interconexión de centralitas	12
Conclusiones	14

Introducción

Se deberá configurar la red en modo Bridge, si se usa el modo NAT funcionará todo menos la voz, por lo tanto, poco uso para una centralita telefónica.

Aplicaciones usadas:

- MicroSIP (Windows)
- MizuDroid (Android)

La aplicación de Zoiper se descartó ya que no funcionaba el teclado.

Instalación

Siguiendo los pasos de [este tutorial](#) vamos a instalar y configurar el software de Asterisk

```
sudo apt update  
sudo apt -y upgrade
```

Instalar las dependencias necesarias:

```
sudo apt -y install git curl wget libnewt-dev libssl-dev libncurses5-dev subversion libsqlite3-dev build-essential libjansson-dev libxml2-dev uuid-dev
```

Descargamos y extraemos la última versión de Asterisk disponible en su web

```
cd ~  
wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-18-current.tar.gz  
tar xvf asterisk-18-current.tar.gz
```

Si queremos las librerías mp3

```
cd asterisk-18*/  
contrib/scripts/get_mp3_source.sh
```

Comprobamos que esté todo instalado

```
sudo contrib/scripts/install_prereq install
```

Procedemos a configurar e instalar Asterisk

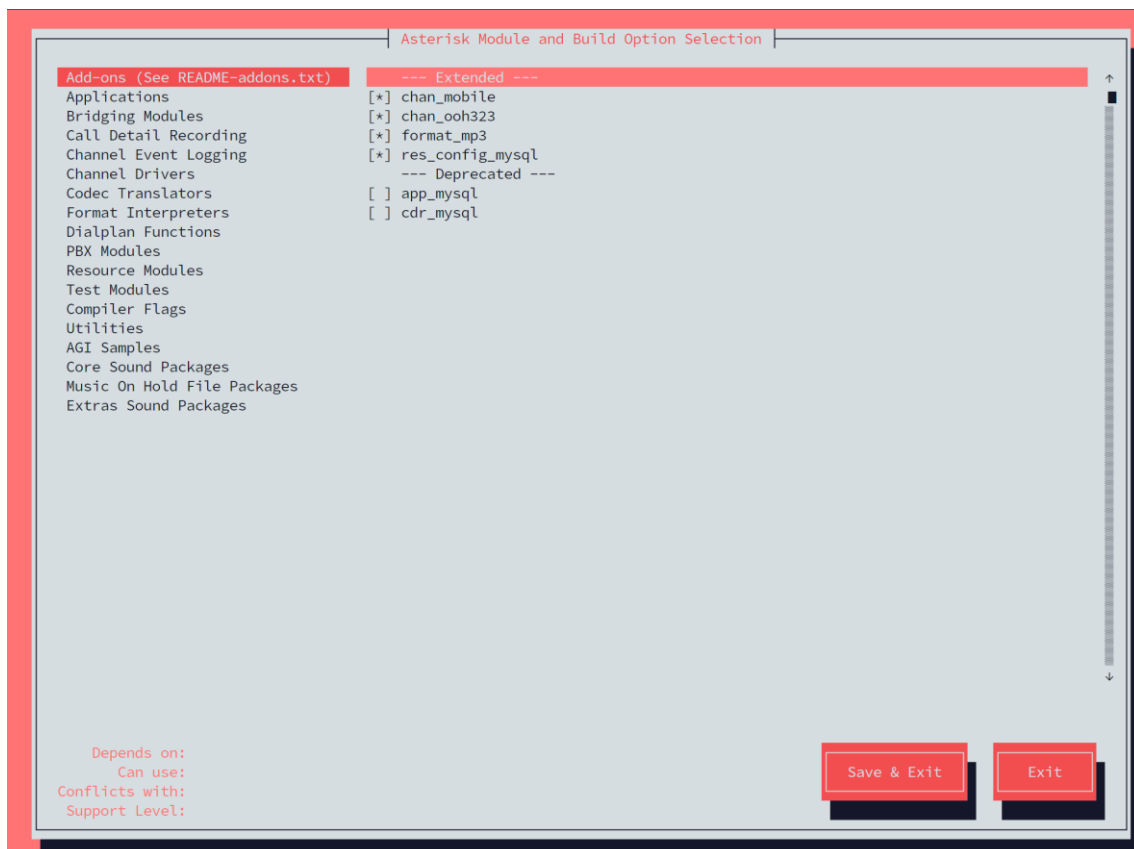
En la carpeta de Asterisk usamos el script de configuración

```
./configure
```

Entramos al menú para elegir qué vamos a instalar

```
make menuselect
```

Saldrá un menú similar a este



Y, por último, compilamos e instalamos

```
sudo make  
sudo make install
```

Si todo se instala sin problemas verás una pantalla como la siguiente

```

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

```

```

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 :
alberto@osboxes:~/asterisk-18$
```

Ya tenemos Asterisk instalado, si queremos podemos añadir algunas configuraciones de ejemplo

```
| sudo make samples
| sudo make config
| sudo ldconfig
```

Para tener una PBX básica funcionando

```
| sudo make basic-pbx
```

Para entrar en la CLI de Asterisk y comprobar que funciona usaremos el comando

```
| sudo asterisk -rvvvv
```

Si no está iniciado, iniciamos el servicio

```
| sudo service asterisk start
```

Configuración de la centralita

Usaremos los archivos `pjsip.conf` para configurar los clientes y `extensions.conf` para las extensiones.

PJSIP.CONF

Abrimos el archivo `pjsip.conf` con el editor que queramos, nosotros usaremos `nano`

```
| sudo nano /etc/asterisk/pjsip.conf
```

Ya que el método de transporte será igual para todos, esta parte será común y solo es necesario definirlo una vez

```
| [transport-udp-nat]
| type=transport
| protocol=udp
| bind=0.0.0.0
```

Un ejemplo de usuario podría ser el siguiente, el resto de los usuarios serían igual, pero con distinto nombre, cambiando todos los 101 por 102, tendremos a otro usuario

```
| [101]
| type=endpoint
| context=lrss
| disallow=all
| allow=alaw
| aors=101
| auth=auth101
| transport=transport-udp-nat
| callerid=delfin <101>
| language=es
|
| [101]
| type=aor
| max_contacts=1
| remove_existing=yes
|
| [auth101]
| type=auth
| username=101
| password=1234
```

Podremos comprobar que están bien configurados si aparecen en la CLI de Asterisk con el comando

```
| pjsip show endpoints
```

EXTENSIONS.CONF

Para continuar, vamos a configurar las extensiones del dialplan

```
| sudo nano /etc/asterisk/extensions.conf
```

La forma de configurar será la siguiente

```
| [lrss]
| exten => 101,1,Dial(PJSIP/101, 15, tT)
| same => n,Playback(voice-mail-system)
| same => n,VoiceMail(101@default)
| same => n,Hangup()
```

Donde

- [lrss] es el contexto
- exten => 101 define la extensión 101 a la que podremos llamar
- ,1,Dial(PJSIP/101, 15, tT) Espera a que se conteste la llamada, dirigida al cliente 101 definido antes, durante 15 segundos
- Playback(voice-mail-system) Reproducirá un audio pidiendo que dejemos un mensaje grabado para el buzón de voz
- same => n,VoiceMail(101@default) Grabará el mensaje en el buzón del cliente 101
- same => n,Hangup() Colgamos una vez finalice todo

Para que estos archivos de configuración sean cargados habrá que hacer un reload de la configuración, podemos usar los siguientes comandos:

Desde la CLI de Asterisk:

```
| dialplan reload // Recarga la configuración del dialplan
| reload // Recarga todo
```

Fuera de la CLI:

```
| sudo service asterisk reload // Recarga asterisk
| sudo service asterisk restart // Reinicia asterisk
```

Voicemail

Como hemos visto, si la llamada no es contestada se nos pedirá grabar un mensaje para el buzón de correo. Para esto configuraremos el voicemail.

```
| sudo nano /etc/asterisk/voicemail.conf
```

Con la siguiente configuración

```
| [general]
| format=wav49|gsm|wav
```

```
maxmsg=100
maxsecs=60
minsecs=3
skipms=3000
maxsilence=10
silencethreshold=128
maxlogins=3

; Cliente => Contraseña del buzón
[default]
101 => 1234
102 => 1234
```

Ahora volvemos al archivo de extensiones para añadir la extensión que nos permitirá escuchar el buzón

```
exten => 301,1,NoOp()
same => n,Answer()
same => n,VoiceMailMain(101@default,s)
; La opción s nos permite escuchar el buzón sin pedir contraseña
same => n,Hangup()
```

Audios en español

Como vemos, la voz que nos habla no suena muy española, así que usaremos un paquete de voces en español más actualizado.

Para esto seguimos las instrucciones de <https://www.asterisksounds.org/es-es/instalar>

Menú IVR

Esta configuración nos va a permitir crear un menú que nos preguntará con quien deseamos comunicarnos y nos redirigirá de una manera rápida y sencilla. Además, si el parámetro introducido no forma parte de los esperados o se agota el tiempo de espera el menú se reinicia.

Para configurar este menú tenemos que acceder al archivo de configuración de extensiones:

```
Sudo nano /etc/Asterisk/extensions.conf
```

Aquí añadiremos la siguiente configuración:

```
[menu_ivr]
exten => s,1,Answer()
; Si número invalido o tiempo agotado se vuelve aqui

; Se reproduce el fichero de audio mientras se espera una respuesta
same => n(menu_principal),Background(if-u-know-ext-dial)
; Se esperan 10s más...
```



```

same => n,WaitExten(10)

; Llamamos a extension correspondiente
exten => 1,1,NoOp(Pulsado 1)
same => n,Dial(PJSIP/101)

; Llamamos a extension correspondiente
exten => 2,1,NoOp(Pulsado 2)
same => n,Dial(PJSIP/102)

; Si el número no es valido, volvemos al principio.
exten => i,1,NoOp(num invalido)
exten => i,n,Playback(pbx-invalid)
exten => i,n,Goto(s,menu_principal)

; Si el tiempo de espera se agota, volvemos al principio.
exten => t,1,NoOp(tiempo agotado)
exten => t,n,Playback(vm-sorry)
exten => t,n,Goto(s,menu_principal)

```

Base de datos

Para la gestión de datos de la PBX se ha conectado la misma con una base de datos, utilizando el sistema gestor de bases de datos MySQL y el conector OCBD.

Para comenzar la configuración hay que descargar primero MySQL:

```

Sudo apt-get install mysql-server
Sudo service mysqld start

```

Como queremos mejorar la seguridad de nuestra base de datos para no exponer la información de ningún usuario vamos a ejecutar también este script y establecer ciertos mecanismos de seguridad.

```

sudo /usr/bin/mysql_secure_installation

```

Ya podemos entrar a la consola de MySQL y comenzar a trabajar. Para entrar por primera vez utilizaremos:

```

mysql -u root -p

```

A continuación, tenemos que crearnos unas credenciales con las que acceder:

```

Mysql> CREATE USER 'asterisk'@'%' IDENTIFIED BY '_Contraseña_';
Mysql> CREATE DATABASE asterisk;
Mysql> GRANT ALL PRIVILEGES ON asterisk.* TO 'asterisk'@'%';

```

También podemos crear ya la base de datos:

```

CREATE TABLE cdr (

```

```
calldate timestamp NOT NULL ,
clid varchar (80) NOT NULL ,
src varchar (80) NOT NULL ,
dst varchar (80) NOT NULL ,
dcontext varchar (80) NOT NULL ,
channel varchar (80) NOT NULL ,
dstchannel varchar (80) NOT NULL ,
lastapp varchar (80) NOT NULL ,
lastdata varchar (80) NOT NULL ,
duration int NOT NULL ,
billsec int NOT NULL ,
disposition varchar (45) NOT NULL ,
amaflags int NOT NULL ,
accountcode varchar (20) NOT NULL ,
uniqueid varchar (150) NOT NULL ,
userfield varchar (255) NOT NULL
);
```

Un paso crítico es la conexión de la PBX con la base de datos mediante OCBD.

Para ello primero hay que instalar los conectores oficiales de la página de MySQL:

- **mysql-connector-odbc-setup-dbgsym_8.0.28-1ubuntu20.04_amd64.deb**

```
wget https://dev.mysql.com/get/Downloads/Connector-ODBC/8.0/mysql-connector-odbc-setup-
dbgsym_8.0.28-1ubuntu20.04_amd64.deb
```

- **mysql-connector-odbc-setup_amd64.deb**

```
wget https://dev.mysql.com/get/Downloads/Connector-ODBC/8.0/mysql-connector-odbc-setup\_8.0.28-1ubuntu20.04\_amd64.deb
```

- **mysql-connector-odbc_8.0.28-1ubuntu20.04_amd64.deb**

```
wget https://dev.mysql.com/get/Downloads/Connector-ODBC/8.0/mysql-connector-odbc\_8.0.28-1ubuntu20.04\_amd64.deb
```

También hay que instalar los client-plugins de mysql.

```
sudo apt-get install libgtk-3-0 libgtk2.0-0
sudo dpkg -i mysql-community-client-plugins_8.0.28-1ubuntu20.04_amd64.deb
```

Y ahora podemos instalar los conectores que hemos descargado antes.

```
sudo dpkg -i mysql-connector-odbc_8.0.28-1ubuntu20.04_amd64.deb
sudo dpkg -i mysql-connector-odbc-setup_8.0.28-1ubuntu20.04_amd64.deb
sudo apt-get install -f
```

A continuación, hay que modificar los siguientes archivos de configuración:

```
sudo nano /etc/odbcinst.ini
```

```
[MySQL ODBC 8.0 Unicode Driver]
```

```
Driver=/usr/lib/x86_64-linux-gnu/odbc/libmyodbc8w.so
UsageCount=1
```

```
[MySQL ODBC 8.0 ANSI Driver]
Driver=/usr/lib/x86_64-linux-gnu/odbc/libmyodbc8a.so
UsageCount=1
```

```
odbcinst -q -d
```

```
sudo nano /etc/odbc.ini
```

```
[asterisk-connector]
Description = MySQL connection to 'asterisk' database
Driver = MySQL ODBC 8.0 Unicode Driver

Database = asterisk
Server = localhost
Port = 3306
Socket = /var/run/mysqld/mysqld.sock
```

```
/etc/asterisk/res_odbc.conf
```

```
[asterisk]
enabled => yes
dsn => Asterisk-connector
username => asterisk
password => 1234
logging => yes
pre-connect => yes
```

Para comprobar que todo ha sido instalado correctamente, podemos abrir la consola de Asterisk e introducir el comando:

```
CLI> odbc show
```

Si hemos seguido todos los pasos, Asterisk debería reconocer la conexión con la base de datos.

Agentes

Para los agentes hemos seguido el mismo método anterior para añadir más clientes, en este caso el cliente es un agente con usuario 10001/2

Definimos los agentes en el archivo `/etc/asterisk/agents.conf`

```
; Define a template called my-agents:
[my-agents]()
autologoff=15
```

```

ackcall=yes
acceptdtmf=#
musiconhold=default
custom_beep=beep
;
; Define agent 1001 using the my-agents template:
[1001](my-agents)
fullname=Mark Spencer
;
; Define agent 1002 using the my-agents template:
[1002](my-agents)
fullname=Will Meadows

```

Ahora definimos la cola en `/etc/asterisk/queues.conf`

```

[queue]
music=default
wrapuptime=0
joinempty=yes
strategy=ringall
timeout=15
retry=5
announce-frequency=15
announce-holdtime=yes

member => Local/1001@lrss,0,Mark Spencer,Agent:1001
member => Local/1002@lrss,0,Will Meadows,Agent:1002

```

En último lugar, agregamos las extensiones de los agentes y la cola donde entrarán los clientes

a `/etc/asterisk/extensions.conf`

```

; Agentes
[agents]
exten => 1001,hint,PJSIP/1001
exten => 1002,hint,PJSIP/1002

[lrss]
exten => 1001,1,Answer()
same => n,AgentLogin(1001)
same => n,Hangup()

exten => 1002,1,Answer()
same => n,AgentLogin(1002)
same => n,Hangup()

exten => 2001,1,Answer()
same => n,Queue(queue)
same => n,Hangup()

```

Como podemos ver en la CLI de Asterisk, nuestros agentes al llamar a su extensión correspondiente quedarán en espera de una llamada

```

osboxes*CLI> agent show all
Agent-ID Name           State      Channel           Talking with
1001      Mark Spencer          NOT_INUSE   PJSIP/1001-0000000d
1002      Will Meadows          UNAVAILABLE
Defined agents: 2, Logged in: 1, Talking: 0

```

Interconexión de centralitas

Servidor 1:

Añadimos la configuración del otro servidor en `/etc/asterisk/pjsip.conf`

```
[udptransport]
type=transport
protocol=udp
bind=0.0.0.0

[SERVER02]
type=registration
outbound_auth=SERVER02
server_uri=sip:SERVER01@192.168.13.206:5060
client_uri=sip:SERVER01@192.168.13.206:5060

[SERVER02]
type=auth
auth_type=userpass
password=1234
username=SERVER01

[SERVER02]
type=aor
contact=sip:192.168.13.206:5060
max_contacts=10

[SERVER02]
type=endpoint
context=lrss
disallow=all
allow=ulaw,alaw,gsm
transport=udptransport
outbound_auth=SERVER02
aors=SERVER02

[SERVER02]
type=identify
endpoint=SERVER02
match=192.168.13.206
```

Y la extensión correspondiente en `/etc/asterisk/extensions.conf`

```
; Trunk
[lrss]
exten => 9001,1,Answer()
exten => 9001,n,NoOp(Out Trunk Call)
exten => 9001,n,Dial(PJSIP/9001@SERVER02,60)
exten => 9001,n,Congestion(3)
exten => _91XX,1,Answer()
exten => _91XX,n,NoOp(Out Trunk Call)
exten => _91XX,n,Dial(PJSIP/${EXTEN:1}@SERVER02,60)
exten => _91XX,n,Congestion(3)
```

Servidor 2:

Añadimos primero la configuración de `/etc/asterisk/pjsip.conf`:

```
GNU nano 4.8
[transport-udp-nat]
type=transport
protocol=udp
bind=0.0.0.0

[SERVER01]
type=registration
outbound_auth=SERVER01
server_uri=sip:SERVER02@192.168.13.17:5060
client_uri=sip:SERVER02@192.168.13.17:5060

[SERVER01]
type=auth
auth_type=userpass
password=1234
username=SERVER02

[SERVER01]
type=aor
contact=sip:192.168.13.17:5060
max_contacts=10

[SERVER01]
type=endpoint
context=llamadas
disallow=all
allow=ulaw,alaw,gsm
transport=transport-udp-nat
outbound_auth=SERVER01
aors=SERVER01

[SERVER01]
type=identify
endpoint=SERVER01
match=192.168.13.17
```

Y a continuación `/etc/Asterisk/extensions.conf`:

```
[endpoints]
exten => _2XX,1,Answer()
exten => _2XX,n,Dial(PJSIP/${EXTEN})
exten => _2XX,n,Congestion()
[llamadas]
include => endpoints
exten => 9001,1,Answer()
exten => 9001,n,NoOp(In Trunk Call)
exten => 9001,n,Playback(demo-congrats)
exten => 9001,n,Dial(PJSIP/2001,60)
exten => 9001,n,Congestion(3)
```

Conclusiones

Tras seguir estos pasos hemos conseguido crear una centralita que, mediante Asterisk, provee servicios básicos de VoIP. Además, se han añadido funcionalidades extra que permiten implementar música de espera, buzón de voz, menús con redirección de llamadas, llamadas grupales e incluso interconexión y llamadas entre distintas centralitas Asterisk.

Por último para ejecutar las distintas funcionalidades, dejamos las extensiones de interés;

- 30x: Buzon de voz del cliente 10x
- 200: Demo congrats
- 220: Demo música en espera
- *21: Demo redireccionamiento llamada
- 777: Llamada grupal
- 1000: Menu IVR
- 100x: Agente 100x hará login
- 2001: Cliente llamará a la cola
- 9XXX: Llamadas entre centrales (9104 para llamar al usuario 104).