



Guía de Instalación y Configuración.

Índice

1	Introducción	3
2	Primeros Pasos	4
2.1	Obtención del SO	4
2.2	Instalación del Sistema Operativo	4
2.3	Obtención de Asterisk	5
2.4	Instalación de Asterisk	6
2.5	Iniciar el Servicio	7
3	Configuración de Asterisk	8
3.1	¿Cómo funciona?	8
3.2	Archivos de configuración	9
3.3	Desarrollo de la configuración	10
4	¿Y funciona?	14
5	Conclusión	16
6	Bibliografía	17
7	Agradecimientos	17

1 .- Introducción

Este es un pequeño tutorial de cómo instalar y configurar Asterisk, la PBX Open Source para Linux.

Se verán distintos aspectos de instalación desde el primer paso, que es instalar el Sistema Operativo, la configuración de este y herramientas anexas que pueda necesitar Asterisk.

Se aclararan algunos términos técnicos propios de VoIP, pero el enfoque principal es el funcionamiento de la PBX.

La versión del software es la 1.0.7, el último release oficial, todos los paquetes adicionales que pueda usar Asterisk, ya sea sonidos o drivers son la misma versión.

El protocolo que se utiliza para tener comunicación es SIP (Session Initiation Protocol) y el cliente (softphone) es el X-Lite de la empresa Xten, en su versión para Windows y Linux.

Asterisk es el mas poderoso, flexible y extenso software de telecomunicaciones disponible. Su nombre viene del símbolo asterisco “*”, que en ambientes UNIX y DOS representa un wildcard. Similarmente la PBX Asterisk está diseñada para conectar cualquier hardware telefónico o cualquier tipo de software de telefonía de manera transparente y consistente.

Tradicionalmente, los productos telefónicos son diseñados para ejecutar una tarea específica en una red. Sin embargo, gran cantidad de aplicaciones de telefonía comparten gran cantidad de tecnología. Asterisk toma ventaja de esta sinergia para crear un solo entorno de desarrollo que puede ser moldeado a cualquier necesidad que el usuario requiera.

Asterisk, además de muchas otras cosas, puede ser usado en cualquiera de estas aplicaciones:

- VoIP Gateway (MGCP, SIP, IAX, H.323)
- Private Branch eXchange (PBX)
- Servidor de voz de respuesta interactiva (IVR)
- Softswitch
- Servidor de Conferencias
- Traductor de números
- Y muchas otras...

2 .- Primeros Pasos.

2.1 .- Obtención del SO.

El sistema operativo utilizado para este proyecto es Slackware Linux en su versión current. Elegí este SO por ser en el que más experiencia tengo y el que más me acomoda, pero Asterisk funciona en cualquier distribución de Linux o *nix en general, ya que los requerimientos son mínimos.

Slackware Linux se puede obtener desde el siguiente link torrent: <http://slackware.com/torrents/slackware-10.1-install-d1.torrent>, la última versión estable es la 10.1.

El SO lo instale en un computador con procesador AMD de 700mhz con 192 Mb en RAM, 10 Gb de espacio en disco duro y una tarjeta de red 3com.

Es el único SO que existe en el servidor, por lo tanto es buena recomendación respaldar y borrar el disco, aunque de todas formas puede convivir con otro Sistema Operativo (este no es el caso).

2.2 .- Instalación del Sistema Operativo

Se asume cierto conocimiento anterior para no entrar en detalles de como instalar Linux, me voy a referir a los espacios que ocupa cada partición y qué es lo que necesita Asterisk para ejecutarlo sin problemas pero, como todo en Linux, esto es a gusto de uno y no es una regla predeterminada a seguir.

El disco duro se particionó de la siguiente forma:

```
Disk /dev/hda: 10.2 GB, 10262568960 bytes
255 heads, 63 sectors/track, 1247 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1		1	37	297171	83	Linux
/dev/hda2		38	87	401625	82	Linux swap
/dev/hda3		88	1247	9317700	5	Extended
/dev/hda5		88	100	104391	83	Linux
/dev/hda6		101	162	497983+	83	Linux
/dev/hda7		163	528	2939863+	83	Linux
/dev/hda8		529	1247	5775336	83	Linux

Filesystem	Size	Mounted on
/dev/hda1	291M	/
/dev/hda2	300M	swap
/dev/hda5	102M	/home
/dev/hda6	487M	/tmp
/dev/hda7	2.9G	/var
/dev/hda8	5.6G	/usr

Como se puede ver, se utilizaron 5 particiones para Slackware más la partición swap. Asterisk en sí no ocupa mucho espacio en disco pero, por ejemplo al tener una cantidad grande de mensajes en espera a ser escuchados, puede ocupar una gran cantidad de espacio, por eso se asignó

tanto espacio de disco para /var. Como la mayoría de los servicios, Asterisk guarda sus cosas en /var, las demás particiones requieren espacio normal. Usé una instalación mínima del sistema operativo, alrededor de 600 Mb, el servidor no requerirá de sistema X-Window, ni de mucho software que ofrece Slackware.

Los requerimientos mínimos para que funcione Asterisk son:

- * NCurses y librerías de desarrollo asociadas
- * OpenSSL y librerías de desarrollo asociadas
- * zlib y librerías de desarrollo asociadas
- * bison y librerías de desarrollo asociadas
- * Kernel Linux en su versión 2.4 (incluidas las fuentes)

Todo esto lo trae Slackware en el CD de instalación, y la mayoría de las otras distribuciones de Linux también, así que solo se debe instalar un sistema con esto más el soporte para tener comunicaciones en red.

Leí en internet que también se puede utilizar el kernel de la versión 2.6, pero hay que hacer unas modificaciones en Asterisk para que funcione.

Así quedo repartido en espacio el SO + Asterisk instalado:

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/hda1	291M	77M	214M	27%	/
/dev/hda5	102M	33M	70M	32%	/home
/dev/hda6	487M	33M	455M	7%	/tmp
/dev/hda7	2.9G	50M	2.8G	2%	/var
/dev/hda8	5.6G	924M	4.7G	17%	/usr

2.3 .- Obtención de Asterisk

Asterisk es un software desarrollado por la empresa Digium bajo licencia GPL, este puede ser descargado de la página principal <http://www.Asterisk.org>, a continuación doy los links para poder bajar la versión que utilicé:

- <http://www.Asterisk.org/html/downloads/Asterisk-1.0.7.tar.gz>
- <http://www.Asterisk.org/html/downloads/Zaptel-1.0.7.tar.gz>
- <http://www.Asterisk.org/html/downloads/Asterisk-sounds-1.0.7.tar.gz>

Antes de proceder con la instalación, debo aclarar un punto:

Para que Asterisk pueda reproducir sonidos, necesita un programa llamado mpg123, Slackware hace algunas versiones eliminó este software de su distribución porque su licencia no es completamente libre, y en su reemplazo puso a su “hermano pobre” mpg321.

Aunque este programa es opcional y no afecta el funcionamiento de la PBX, yo bajé e instale la última versión de mpg123 desde:

- <http://www.mpg123.de/mpg123/mpg123-0.59r.tar.gz>

2.4 .- Instalación de Asterisk

Antes de compilar Asterisk, necesitamos tener arriba ztdummy, este es un modulo para el kernel que provee a Asterisk un timer Zaptel si es que no tenemos ningun hardware Digium instalado. El modulo ztdummy toma el timing desde el modulo del kernel usb-uhci que debe ser cargado antes que ztdummy. El modulo usb-uhci no debe ser compilado dentro del kernel, si este fuera el caso, el timing no funciona. Todo este proceso es solamente para poder realizar conferencias con Asterisk, se necesita el timer para poder realizarlas.

El primer paso entonces es compilar nuestro kernel para dejar como modulo a usb-uhci.

```
USB support --->
  <*> Support for USB
    <M> UHCI (Intel PIIX4, VIA, ...) support
```

Esas son las opciones en el kernel para habilitar el modulo, la manera de compilar, es la misma de siempre y es ajena a este documento. Creado el modulo procedemos a compilar zaptel.

```
bash# tar xzfv zaptel-1.0.7.tar.gz
bash# cd zaptel-1.0.7/
```

En este momento tenemos que editar el Makefile, lo abrimos con el editor preferido, y buscamos la palabra ztdummy, aparecera inmediatamente, y tendra un # antes.

```
MODULES=zaptel tor2 torisa wcusb wcfxo wcfxs \
  ztdynamic ztd-eth wctl1xxp wct4xxp wctellxp # ztdummy
```

Quitamos el #, se guarda el archivo y con esto se habilita ztdummy. Lo que sigue es lo mas sencillo.

```
bash# make
bash# make install
```

Luego solo tenemos que agregar a /etc/rc.d/rc.modules las siguientes lineas:

```
modprobe usb-uhci
modprobe zaptel
modprobe ztdummy
```

Tambien es buena idea cargar los modulos, o reiniciar la maquina. Ahora podemos proceder con la compilacion de Asterisk.

El proceso de instalación es bastante simple y no es distinto a cualquier software para plataformas *nix.

Primero se debe descomprimir el archivo asterisk-1.0.7.tar.gz y luego proceder a su compilación e instalación.

```
bash# tar xzfv asterisk-1.0.7.tar.gz
bash# cd asterisk-1.0.7/
bash# make
```

```
bash# make install
```

Como esta es la primera vez que administro Asterisk, hice lo que se recomienda en su página e instalé las extensiones de demostración que trae la PBX

```
bash# make samples
```

Con estos simples comandos se compila e instala Asterisk.

El proceso de compilación es limpio, no tuve ningún error y por lo tanto, el que tome esta guía tampoco debería tenerlos. Si es que llegan a existir problemas de compilación, ya sea porque se está usando otra distribución Linux o de cualquier otro tipo, recomiendo que lean la lista de mail de Digium.

Para poder comprobar si la PBX está instalada correctamente hay que ejecutar el siguiente comando:

```
bash# asterisk -vvvc
```

Se va a ver una gran cantidad de mensajes, que son los que arroja Asterisk al iniciar, para luego terminar en una terminal como esta.

```
*CLI>
```

Se puede teclear `help` en cualquier momento para la ayuda de Asterisk y sus comandos básicos.

El siguiente paso es instalar los sonidos de Asterisk, los pasos son parecidos al anterior.

```
bash# tar xzfv asterisk-sounds-1.0.7.tar.gz
bash# cd asterisk-sounds-1.0.7/
bash# make install
```

Los distintos sonidos van a ser guardados en `/var/lib/asterisk/sounds` y todos están en inglés, existe un proyecto en desarrollo en el sitio Asteriskspain.org, donde locutores profesionales van a traducir todos los sonidos al español, según la página esto va a estar disponible luego.

2.5 .- Iniciar el Servicio

Para levantar Asterisk cuando el sistema inicia, hay que crear un script que debe ser guardado en `/etc/rc.d/rc.asterisk` y debe tener permisos de ejecución, el archivo es el siguiente:

```
#!/bin/sh
#
# Start/stop/restart Asterisk PBX
#
# Version: 1.0 - Paul Belanger <pabelanger at gmail.com>
```

```

#
# 03.29.2005 - Initial Version
#

Asterisk_start() {
    if [ -x /usr/sbin/Asterisk ]; then
        echo "Starting Asterisk    /usr/sbin/Asterisk"
        /usr/sbin/Asterisk
    fi
}

Asterisk_stop() {
    # If there is no PID file, ignore this request...
    if [ -r /var/run/Asterisk.pid ]; then
        killall Asterisk
    fi
}

Asterisk_restart() {
    Asterisk_stop
    Asterisk_start
}

case "$1" in
'start')
    Asterisk_start
    ;;
'stop')
    Asterisk_stop
    ;;
'restart')
    Asterisk_restart
    ;;
*)
    echo "usage $0 start|stop|restart" ;;
esac

```

El permiso de ejecución se da con `chmod`

```
bash# chmod 755 /etc/rc.d/rc.Asterisk
```

además se debe agregar las siguientes líneas al archivo `/etc/rc.d/rc.local`

```
# Asterisk, con esto inicio Asterisk
. /etc/rc.d/rc.Asterisk start
```

Con esto último tenemos el Sistema Operativo y Asterisk instalado, ahora hay que pasar a la configuración de la PBX.

3.- Configuración de Asterisk

3.1.- ¿Cómo funciona?

Asterisk funciona sin la necesidad de hardware alguno para conectividad con la PSTN, siendo esta la gran ventaja para este proyecto que esta basado en la Voz sobre IP, el software nos entrega conectividad en tiempo real para redes VoIP y sólo cuento con una tarjeta de red.

Asterisk es mucho mas que una PBX normal y se puede hacer telefonía de nuevas formas. Puede conectar empleados trabajando en casa hacia la PBX en la oficina mediante una conexión de banda ancha, conecta oficinas mediante varias alternativas de VoIP, Internet o una red privada virtual, entrega voicemail, intregrada con una web y algún mail, puede construir aplicaciones interactivas de voz y mucho más. Además incluye cualidades que sólo se encuentran en productos top de la mensajería unificada.

Para poder hacer todo esto, Asterisk funciona mediante canales. Estos canales son drivers para distintos tipos de conexiones para protocolos de VoIP como SIP, IAX, MGCP y H.323.

Telefonos y Softphones conectan a un canal, algunos de ellos se registran (en el proyecto todos se registran) para dar a conocer que estan en linea. Los canales también registran conexiones salientes a otro servidor VoIP, mediante SIP nos podemos conectar a la red Free World Dialup o a proveedores SIP como Nufone , Vonage o Siphone.

En el proyecto uso como protocolo SIP, que es muy parecido a HTTP o a SMTP. El mensaje consiste en una cabecera (header) y un cuerpo. Es un protocolo basado en texto que usa la codificacion UTF-8 y el puerto 5060 para conexiones tcp y udp, y ofrece todas las gamas de posibilidades de la telefonía moderna. Dado que es un protocolo muy flexible es posible agregar funciones y aumentar la operabilidad.

Entender la configuracion de Asterisk es algo así como aprender a programar, son eventos que van sucediendo, y que se tienen que ir organizando paso a paso, para que la PBX sepa que hacer. Ahora en el siguiente paso voy a dar a conocer los archivos de configuración y cuales fueron los que utilice para la PBX.

3.2 .- Archivos de configuración

Todos los archivos de configuración se encuentran en /etc/asterisk y son bastantes, pero para este caso solo utilizaremos algunos, estan muy bien comentados todos.

A continuación voy a dar un volcado de cuantos archivos son.

```
bash# ls /etc/asterisk
adsi.conf          cdr_pgsql.conf    indications.conf  oss.conf          skinny.conf
adtranvoivr.conf  cdr_tds.conf      logger.conf       phone.conf        telcordia-1.adsi
agents.conf        enum.conf         manager.conf      privacy.conf      voicemail.conf
alarmreceiver.conf extconfig.conf    meetme.conf       queues.conf       vpb.conf
alsa.conf          extensions.conf   mgcp.conf         res_config_odbc.conf zapata.conf
asterisk.adsi      features.conf     modem.conf        res_odbc.conf
asterisk.conf      festival.conf     modules.conf      rpt.conf
cdr_manager.conf  iax.conf         musiconhold.conf rtp.conf
cdr_odbc.conf     iaxprov.conf     osp.conf          sip.conf
```

Como se puede ver son una gran cantidad de archivos, pero como dije antes,

para este caso vamos a usar algunos y son los siguientes:

```
asterisk.conf
extensions.conf
meetme.conf
musiconhold.conf
sip.conf
voicemail.conf
zapata.conf
```

El archivo más importante de todos es `extensions.conf`, es la llave del funcionamiento de Asterisk, es el Dial Plan.

3.3 .- Desarrollo de la configuración

Para entender que fue lo que hice, voy a volcar cada archivo de configuración, y comentaré dentro del mismo archivo las partes más importantes de cada uno.

El sistema que desarrolle tiene 3 usuarios que se deben registrar para ser parte de la red, esto quiere decir que cada uno tiene nombre de usuario y password, al registrarse se pueden comunicar a cualquier numero dentro de la red y ocupar cualquiera de los servicios que ofrece la PBX.

El primer archivo es `asterisk.conf`, este indica a Asterisk donde se encuentra todo lo que necesita, este archivo se crea solo y no es buena idea modificarlo a no ser que sepan que estan haciendo.

```
;  
; asterisk.conf  
; los comentarios son con un ;  
; este es el archivo que indica los directorios que utiliza Asterisk  
;  
  
[directories]  
astetcdir => /etc/asterisk  
astmoddir => /usr/lib/asterisk/modules  
astvarlibdir => /var/lib/asterisk  
astagidir => /var/lib/asterisk/agi-bin  
astspooldir => /var/spool/asterisk  
astrundir => /var/run  
astlogdir => /var/log/asterisk
```

El siguiente paso es configurar el protocolo, para esto tenemos que editar el archivo `sip.conf` que como su nombre lo indica, es el encargado de manejar las características de SIP. Los clientes deben ser configurados en este archivo antes de poder recibir o hacer llamadas.

El archivo es leído desde arriba hacia abajo. La primera sección es para opciones generales del servidor, como la dirección IP o el puerto. Las secciones siguientes definen parámetros para los clientes, tales como el username, password y la IP por defecto para clientes no registrados.

Las secciones son destacadas por brackets [] , a continuación doy el contenido de `sip.conf`:

```

;
; sip.conf
;
;
[general]                ; Opciones Generales.

port = 5060              ; Defino el puerto (SIP usa 5060)
bindaddr = 0.0.0.0      ; La direccion IP a usar (todas las posibles que existan)
allow=all                ; Permite todo tipo de codecs
context = bogon-calls   ; Envia las llamadas SIP que no conocemos aqui

[2000]                   ; Defino el primer usuario

type=friend              ; Defino el tipo de conexion, en este caso.. AMIGO
username=2000            ; Nombre de usuario
secret=slackware        ; Password :P
host=dynamic             ; El host no siempre tiene la misma IP
context=slack-sip       ; Las llamadas entrantes van a slack-sip
mailbox=100             ; Activa la luz de mensaje en espera si es que
                        ; existe algo en voicemailbox

[2001]                   ; Es un duplicado de 2000, con diferente login

type=friend
username=2001
secret=slackware
host=dynamic
context=slack-sip
mailbox=101

[2002]                   ; Otro duplicado de 2000

type=friend
username=2002
secret=slackware
host=dynamic
context=slack-sip
mailbox=102

```

Con esta configuración el protocolo SIP esta listo para funcionar, ahora el siguiente paso es configurar extensions.conf.

Como ya habia mencionado, extensions.conf es el corazón del funcionamiento de Asterisk, acá se define como se deben manejar las llamadas. Consiste en una lista de instrucciones que Asterisk debe seguir, que son iniciados por dígitos recibidos por un canal o alguna aplicación.

Antes de ver el archivo voy a explicar un poco el contenido, al igual que SIP las secciones se definen en [] y dentro de esta se encuentran las extensiones. Un ejemplo de estas ultimas es algo así:

```

exten => 555,1,Dial(Zap/1,20)
exten => 555,2,Voicemail(u555)

```

El "exten =>" indica al Dialplan que lo próximo será un comando.

El "555" es el dígito actual recibido.

El "1" y el "2" representan la prioridad, que determinan en que orden se tomarán los comandos de esa extensión.

Aclarado esto, pasamos a ver el contenido de extensions.conf.

```

;
; extensions.conf
;

[general]

static=yes          ; Estas dos lineas previenen que desde la linea de
writeprotect=yes   ; comandos se pueda sobrescribir el archivo de configuracion

[bogon-calls]

;
; Toma las llamadas desconocidas que encontraron
; el sistema, y les envia una orden de tono.
; El string "_" borra cualquier secuencia, con esto
; todas las llamadas resiviran el tono de ocupado.
; Eventualmente se aburriran y colgaran.
;

exten => _,1,Congestion

[slack-sip]        ; aqui defino la seccion slack-sip

;
; Si el numero marcado por el que llama es "2000", entonces
; llama al usuario "2000" mediante el canal SIP. Deja que el numero
; suene durante 20 segundos, y si no hay respuesta, procede a la prioridad 2.
; Si el numero retorna un resultado "busy", entonces salta a la prioridad 102
;

exten => 2000,1,Dial(SIP/2000,20)

;
; La prioridad 2 envia la llamada al voicemail, y da el mensaje "unavailable
; para el usuario 2000. La unica forma de salir del voicemail
; en esta instancia es colgando.
;

exten => 2000,2,Voicemail(u2000)

;
; Si el numero marcado en la prioridad 1 devuelve un estado
; "busy", entonces el Dial saltara a 101 + (prioridad actual)
; que en este caso seria 101+1=102. Este +101 es construido
; dentro de Asterisk y no necesita ser definido.
;

exten => 2000,102,Voicemail(b2000)
exten => 2000,103,Hangup

;
; Ahora, que pasa si el numero marcado es "2001" o "2002"?
;

exten => 2001,1,Dial(SIP/2001,20)
exten => 2001,2,Voicemail(u2001)
exten => 2001,102,Voicemail(b2001)
exten => 2001,103,Hangup

exten => 2002,1,Dial(SIP/2002,20)
exten => 2002,2,Voicemail(u2002)
exten => 2002,102,Voicemail(b2002)
exten => 2002,103,Hangup

;
; Ahora defino un numero donde los usuarios puedan alcanzar
; el voicemail. Llamo a la aplicacion VoicemailMain con el
; numero del que llama pasado como variable, asi
; que lo unico que se necesita hacer es teclear el password.

```

```

;
exten => 2999,1,VoicemailMain(${CALLERIDNUM})

;
; Defino un numero para escuchar el Music on Hold
;
exten => 6601,1,WaitMusicOnHold(30)

;
; Con esto puedo incluir las secciones dentro de slack-sip
;

include => help
include => meetme

;
; Esta seccion esta definida en el archivo por defecto,
; es un numero que entrega informacion acerca de Asterisk,
; en el archivo original el numero es una 's'
;

[help]
exten => 666,1,Wait,1 ; Espera un segundo
exten => 666,2,Answer ; Responde la linea
exten => 666,3,DigitTimeout,5 ; Setea el tiempo de digitar en 5 seg.
exten => 666,4,ResponseTimeout,10 ; Setea el tiempo de respuesta en 10 seg.
exten => 666,5,BackGround(demo-congrats) ; Reproduce un mensaje de felicitaciones
exten => 666,6,BackGround(demo-instruct) ; Reproduce algunas instrucciones

exten => 2,1,BackGround(demo-moreinfo) ; Entrega mas informacion.
exten => 2,2,Goto(s,6)

exten => 500,1,Playback(demo-abouttotry); Me deja saber que esta pasando
exten => 500,2,Dial(IAX2/guest@misery.digium.com/s@default) ; Llama el demo de Asterisk
exten => 500,3,Playback(demo-nogo) ; No pudo conectar al demo
exten => 500,4,Goto(666,6) ; Retorna al principio del mensaje

;
; Defino el numero "100" para la conferencia "4000", se usa MeetMe
; y en meetme.conf esta definida la conferencia "4000"
;

[meetme]
;exten => 100,1,Playback,thereare
;exten => 100,2,Playback,callersin
exten => 100,1,MeetMe,4000

```

Ahora solo falta definir las casillas de voz o voicemail, para esto tenemos que editar el archivo voicemail.conf. Este es el encargado de configurar los parámetros para el sistema de buzón de voz, guarda la información en el mailbox, etc.

El archivo esta dividido en dos secciones, la sección general, que contiene distintos parametros, pero en este caso solo utilizaremos el que indica en que formato se guardan los mensajes recibidos. La segunda sección indica la configuración individual de cada casilla.

```

;
; voicemail.conf
;

[general]
format=wav ; formato wav para guardar mensajes

```

```
[local]

;
; formato: password, nombre, direccion de mail para atachar los mensajes de voz
;

2000 => 4321,Hanamichi Sakuragi,H.Sakuragi@slamdunk.org
2001 => 8383,Inkubot,inkubot@slackware.cl
2002 => 1234,Slackware dot CL,contacto@slackware.cl
```

Con esto se puede utilizar el servidor en una red de area local y comunicarse con tres usuarios sin problemas mediante un softphone.

Solo falta la configuracion del Music on Hold y de la conferencia.

Para habilitar una sala de conferencia tenemos que editar meetme.conf, en extensions.conf defino el numero 100 y este llama a la sala 4000, así que solo hay que definir una sala 4000 en meetme.conf:

```
;
; meetme.conf
;

[rooms]

conf => 4000          ; sala 4000
```

No se necesita password, ni siquiera algún tipo de identificación, el que llame al numero 100, será agregado a la conferencia.

Y por último queda habilitar el Music on Hold y para esto en extensions.conf defino el número 6601 para escuchar la música que reproduce Asterisk para este servicio.

Para habilitarlo hay que descomentar lo siguiente en zapata.conf.

```
: musiconhold=default
```

Y queda de esta forma:

```
musiconhold=default
```

Luego el archivo musiconhold.conf se debe editar y dejar de la siguiente manera:

```
;
; musiconhold.conf
; aqui defino las clases para music on hold

[classes]
default => quietmp3:/var/lib/Asterisk/mohmp3
;
; existen otros tipos de clases
;
;loud => mp3:/var/lib/Asterisk/mohmp3
;random => quietmp3:/var/lib/Asterisk/mohmp3,-z
;unbuffered => mp3nb:/var/lib/Asterisk/mohmp3
;quietunbuf => quietmp3nb:/var/lib/Asterisk/mohmp3
;manual => custom:/var/lib/Asterisk/mohmp3,/usr/bin/mpg123 -q -r 8000 -f 8192 -b 2048 -mono -s
```

Para poder usar alguna otra clases, debo descomentar alguna y habilitar en zapata.conf en la variable musiconhold, por ejemplo:

```
musiconhold=loud
```

Con esto habilito la clase loud.

Solo con estos pasos, podemos habilitar el servidor VoIP, ahora solo hace falta probar el funcionamiento del sistema.

4 .- ¿Y funciona?

Aqui solo voy a poner un screenshot del X-Lite para Linux, haciendo una llamada al número 666 [help], en la imagen se ve que la conexión está establecida y funciona de manera correcta.



5 .- Conclusión

Es impresionante la cantidad de funciones que Asterisk proporciona, para este proyecto solo utilize las mas basicas, pero al investigar sobre el tema, se puede observar que tiene todo lo necesario para ser una PBX (o muchas otras cosas) de alto rendimiento.

Asterisk crece de manera rapida, y cada vez mas se van agregando funciones a travez del CVS, los distintos desarrolladores a travez del mundo agregan nuevo codigo o parches cada dia.

Esto hace que sea el entorno ideal para empresas de todo tipo, que quieran una solucion optima para telecomunicaciones, ya que puede trabajar desde un simple servidor de Voz sobre IP hasta una compleja PBX conectada a la PSTN.

Es un software complejo, su configuracion no es de las mas simples, pero luego de tratar de entender lo mas basico, no se hace tan complicado. Esto se debe a la gran cantidad de informacion que se encuentra en Internet, el proyecto asterisk consta de una excelente documentacion en el sitio <http://www.asteriskdocs.org>, tambien existen las preguntas frecuentes (FAQ), y textos escritos por terceros.

Tambien note la gran portabilidad del software, en la pagina de Asterisk, se dice que es la PBX para Linux, pero tambien funciona en la mayoria de las plataformas Unix, por ejemplo, FreeBSD o NetBSD.

Otra excelente caracteristica es la gran cantidad de protocolos que soporta, y la variedad de usos que se le puede dar, con esto se puede amoldar a cualquier situacion real dentro de una empresa o institucion.

Todo este conjunto de cualidades, ya sean tecnicas, de documentacion o portabilidad, a mi gusto hacen de Asterisk una aplicacion de excelente calidad, y una alternativa real para soluciones de alto nivel.

6 .- Bibliografia

Toda la informacion que lei esta en internet, aqui solo pondre de los links que mas visite, para recolectar informacion.

- <http://www.asterisk.org>
Sitio oficial de Asterisk
- <http://www.asteriskdocs.org>
Sitio oficial de la documentacion de Asterisk
- <http://www.voip-info.org/wiki-Asterisk>
Wiki sobre Asterisk

Nota: Y una cantidad de paginas infinita.

7 .- Agradecimientos

Gracias a Alvaro Oliver, por la edicion de este documento.

