

TICAR 2015

ASTERISK DEVELOPER



Ing. Fernando A. Gonzalez

Tecnologías





La Presentación

- Uso de Asterisk “puro”
- Administración de herramientas por Consola
- Guía “paso a paso” basada en experiencias.
- Escenario:
 - Archlinux (anfitrión)
 - Linux Deping (virtualizado)

1er Escenario: MySQL + Asterisk

- Objetivo: Registrar el CDR de Asterisk en una BD

MySQL

Instalación:

- `sudo apt-get install mysql-server`

Creación de Usuario

- `mysql -u root -p`
- `CREATE USER 'asterisk'@'%' IDENTIFIED BY 'asterisk';`

Creación de la BD

- `CREATE DATABASE asterisk;`
- `GRANT ALL PRIVILEGES ON asterisk.* TO 'asterisk'@'%';`
- `mysql -u asterisk -p asterisk`

- **Creación de Tabla CDR**

```
CREATE TABLE cdr (  
  calldate datetime NOT NULL default '0000-00-00 00:00:00',  
  clid varchar(80) NOT NULL default "",  
  src varchar(80) NOT NULL default "",  
  dst varchar(80) NOT NULL default "",  
  dcontext varchar(80) NOT NULL default "",  
  channel varchar(80) NOT NULL default "",  
  dstchannel varchar(80) NOT NULL default "",  
  lastapp varchar(80) NOT NULL default "",  
  lastdata varchar(80) NOT NULL default "",  
  start datetime NOT NULL default '0000-00-00 00:00:00',  
  answer datetime NOT NULL default '0000-00-00 00:00:00',  
  end datetime NOT NULL default '0000-00-00 00:00:00',  
  duration int(11) NOT NULL default '0',  
  billsec int(11) NOT NULL default '0',  
  disposition varchar(45) NOT NULL default "",  
  amaflags int(11) NOT NULL default '0',  
  accountcode varchar(20) NOT NULL default "",  
  uniqueid varchar(32) NOT NULL default "",  
  userfield varchar(255) NOT NULL default "",  
  peeraccount varchar(20) NOT NULL default "",  
  linkedid varchar(32) NOT NULL default "",  
  sequence int(11) NOT NULL default '0'  
);
```

Estructura de conexión

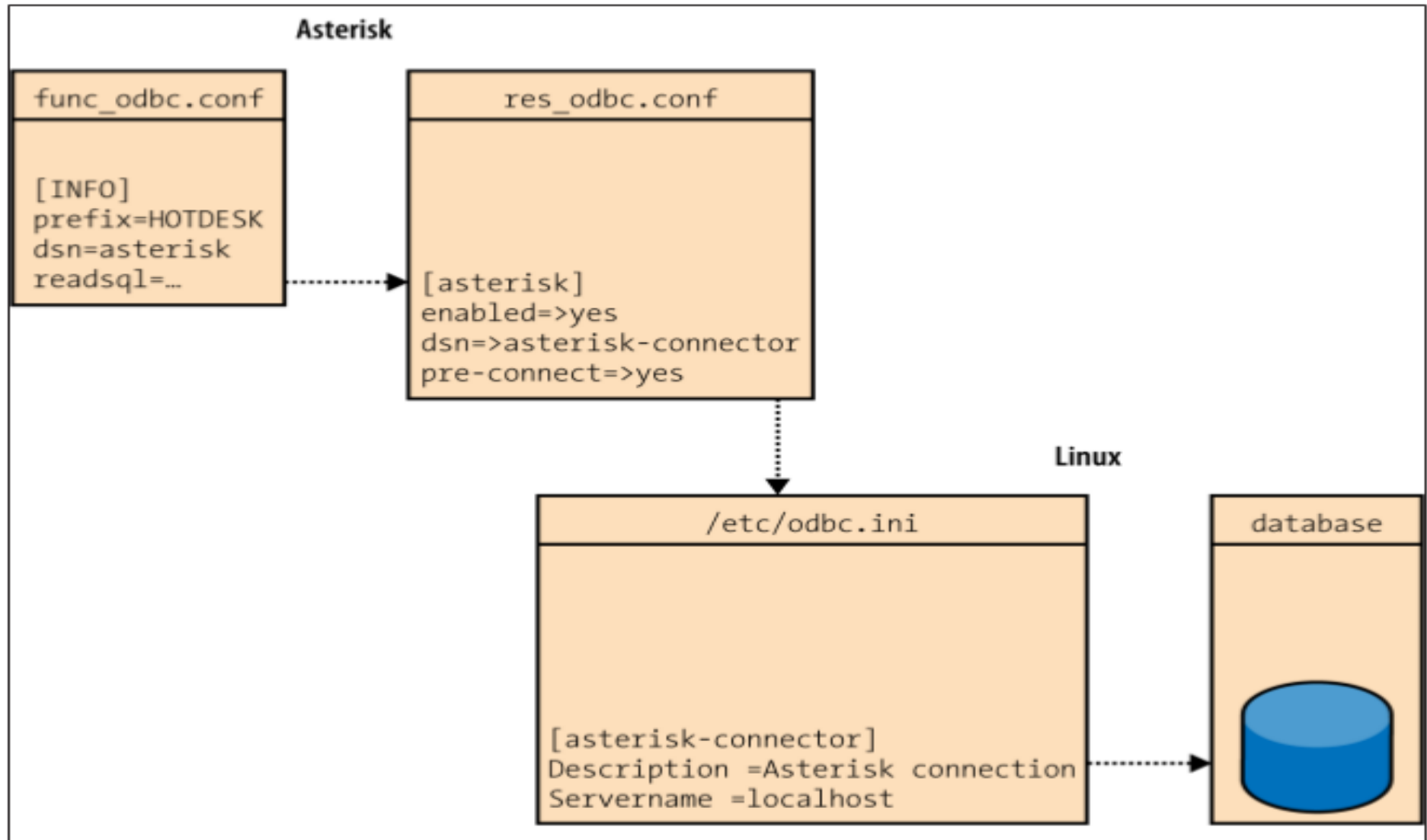


Figure 16-1. Relationships between `func_odbc.conf`, `res_odbc.conf`, `/etc/odbc.ini` (unixODBC), and the database connection

ODBC

Instalar ODBC

- `sudo apt-get install unixODBC unixODBC-dev`

Driver ODBC - MySQL

- `sudo apt-get install libmyodbc`

- **Editar /etc/odbcinst.ini**

```
[MySQL]
```

```
Description<--->= ODBC for MySQL
```

```
Driver<><----->= /usr/lib/x86_64-linux-gnu/odbc/libmyodbc.so
```

```
Setup<-><----->= /usr/lib/x86_64-linux-gnu/odbc/libodbcmyS.so
```

```
FileUsage<----->= 1
```


- **Editar /etc/odbc.ini**

```
[asterisk-connector-mysql]
```

```
Description<---><----->= MySQL connection to 'asterisk' database
```

```
Driver<><-----><----->= MySQL
```

```
Database<-----><----->= asterisk
```

```
Server<><-----><----->= localhost
```

```
Port<--><-----><----->= 3306
```

```
Socket<><-----><----->= /var/run/mysqld/mysqld.sock
```

- **Comprobar:**

```
echo "select 1" | isql -v asterisk-connector-mysql asterisk asterisk
```

Asterisk

- **Editar res_odbc.conf**

[asterisk]

enabled => yes

dsn => asterisk-connector-mysql

username => asterisk

password => asterisk

pooling => no

limit => 1

pre-connect => yes

- **Editar cdr_adaptive_odbc.conf**

```
[adaptive_connection]
```

```
connection=asterisk
```

```
table=cdr
```

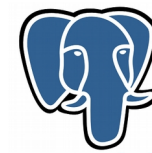
```
usegmtime =no
```

```
alias start => calldate
```

```
alias end => endend
```

2do Escenario: Postgresql + Asterisk

- Objetivo: Interactuar BD en Postgresql + Asterisk



Instalación

- `sudo apt-get install postgresql`

Creación de usuario

- `sudo su - postgres`
- `createuser -P --interactive`

Editar `pg_hba.conf`

```
host<-->all<---><----->asterisk<----->127.0.0.1/32<--><----->md5
```

```
local<->all<---><----->asterisk<-----><-----><-----><----->trust
```

```
host<-->all<---><----->asterisk<----->::1/128<-----><----->md5
```

Restart postgres

- `Sudo service postgres restart`

- **Creación de BD**

```
createdb --owner=asterisk asterisk
```

```
psql -d template1
```

```
ALTER USER asterisk WITH PASSWORD 'asterisk';
```

ODBC

- **Instalar**

```
sudo apt-get install unixODBC unixODBC-dev
```

```
sudo apt-get install odbc-postgresql
```

- **Editar /etc/odbcinst.ini**

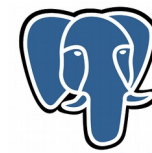
```
[PostgreSQL]
```

```
Description<--->= ODBC for PostgreSQL
```

```
Driver<><----->= psqlodbc.so
```

```
Setup<-><----->= libodbcpsqlS.so
```

```
FileUsage<----->= 1
```



- **Editor /etc/odbc.ini**

```
[asterisk-connector-postgres]
```

```
Description<---><----->= PostgreSQL connection to 'asterisk' database
```

```
Driver<><-----><----->= PostgreSQL
```

```
Database<-----><----->= asterisk
```

```
Servername<----><----->= localhost
```

```
UserName<-----><----->= asterisk
```

```
Password<-----><----->= asterisk
```

```
Port<--><-----><----->= 5432
```

```
Protocol<-----><----->= 8.1
```

```
ReadOnly<-----><----->= No
```

```
RowVersioning<-><----->= No
```

```
ShowSystemTables<----->= No
```

```
ShowOidColum<--><----->= No
```

```
FakeOidIndex<--><----->= No
```

```
ConnSettings<--><----->=
```


- **Comprobar resultados de la conexión:**

```
echo "select 1" | isql -v asterisk-connector asterisk  
some_secret_password
```

- **Crear tabla**

```
CREATE TABLE llamadas_internos  
(  
  id serial NOT NULL,  
  extension text,  
  callerid bigint,  
  date_time timestamp,  
  CONSTRAINT llamadas_internos_id_pk PRIMARY  
  KEY (id)  
)  
WITHOUT OIDS;
```

Asterisk Módulos

- **Compilar módulos relacionados a ODBC en asteriks:**

cdr_odbc

cdr_adaptive_odbc

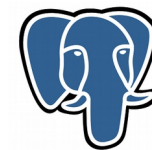
func_odbc

func_realtime

pbx_realtime

res_config_odbc

res_odbc



- **Editar res_odbc**

[asterisk-postgres]

enabled => yes

dsn => asterisk-connector-postgres

username => asterisk

password => asterisk

pooling => no

limit => 1

pre-connect => yes

- **func_odbc**

[LLAMADAS]

prefix=POSTGRESQL

dsn=asterisk-postgres

readsql=INSERT INTO llamadas_internos (extension, callerid, date_time) VALUES ('\${ARG1}', '\${ARG2}', '\${ARG3}');

- **Editar extensions.conf**

```
exten => _5XXX,1,NoOp()
```

```
same => n,Set(timestamp=${STRFTIME(${EPOCH},,%d/  
%m/%C%y %H:%M:%S)})
```

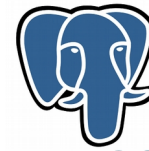
```
same => n,Set(RETURNED_VALUE=${  
{POSTGRESQL_LLAMADAS(${EXTEN},${CALLERID(num)},{  
{timestamp})})})
```

```
same => n,Playback("colocar algún sonido a elección")
```

```
same => n,Hangup()
```

3er Escenario: Postgres + PHP + Asterisk JAVA + Asterisk

- **Objetivo:** Desarrollar aplicaciones PHP o Java en conjunto con Asterisk y sistemas de persistencia de datos.



PHP-AGI

- **Instalar PHP**

```
sudo apt-get install php5
```

```
sudo apt-get install php5-pgsql
```

- **Instalar PHP-AGI**

```
sudo wget
```

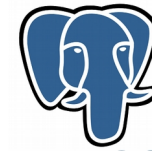
```
http://ufpr.dl.sourceforge.net/project/phpagi/phpagi/2.20/phpagi-2.20.tgz
```

```
tar xzvf phpagi-2.20.tgz -C /var/lib/asterisk/agi-bin
```

- **Crear programa**

```
touch consulta.php
```

```
chmod 775 consulta.php
```

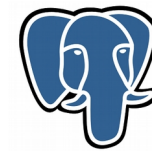


```
#!/usr/bin/php -q  
<?php
```

```
//Estructura básica del framework php-agi  
set_time_limit(30);  
ob_implicit_flush(false);  
require ('phpagi-2.20/phpagi.php') ;  
error_reporting(E_ALL);
```

```
//Consiguiendo variables y mostrando en pantalla.  
$agi = new AGI();  
$agi->answer();  
$agi->set_variable($var1,$agi_channel);  
$agi->channel_status($agi_channel);  
$status = $agi->channel_status($agi_channel);
```

```
//Consiguiendo Caller ID y contando cantidad de Caracteres  
$callid = $agi->request['agi_callerid'];  
$agi->noop("My CallerID: ".$callid);  
$str = strlen($callid);  
$agi->noop("Cantidad de digitos: ".$str);  
//$agi->exec('Playback',"hello-world");
```



```
// Conectando y seleccionado la base de datos..
$dbconn = pg_connect("host=localhost dbname=asterisk user=asterisk password=asterisk")
    or die('No se ha podido conectar: ' . pg_last_error());

// Realizando la consulta SQL
//$ani = $agi->request['agi_dnid'];
$result= pg_query($dbconn,"SELECT * FROM llamadas WHERE id = 1");

$nume2 = pg_num_rows($result);
$row = pg_fetch_array($result);

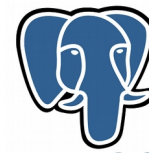
// Comprobando resultados de la consulta
if ( $nume2 == 0) {
    $agi->verbose("No esta en la lista");
    $agi->exec('Playback',"hello-world");
    // $agi->exec('Hangup');
} else {
    $agi->verbose("Esta en la lista");
    $agi->verbose("$row[1] - $row[2]");
    $agi->exec('Playback',"pbx-invalid");
}
// Colgar
$agi->exec('Hangup');

// Liberando el conjunto de resultados
pg_free_result($result);

// Cerrando la conexión
pg_close($dbconn);

sleep(2);
?>
```


Java-AGI



- **Instalar JAVA**

```
sudo apt-get install openjdk-7-jre openjdk-7-jdk
```

- **Descargar:**

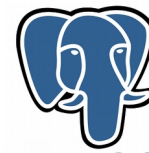
```
sudo wget
```

```
https://nexus.reucon.com/content/repositories/opensource/  
org/asteriskjava/asterisk-java/1.0.0-m2/asterisk-java-1.0.0-  
m2-bin.zip
```

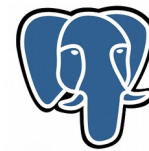
```
sudo unzip asterisk-java-1.0.0-m2-bin.zip
```

- **Crear Programa**

```
sudo touch HelloAgiScript.java
```



```
import org.asteriskjava.fastagi.*;
public class HelloAgiScript2 extends BaseAgiScript {
    private static int counter = 0;
    public void service(AgiRequest request, AgiChannel channel)
        throws AgiException {
    try {
        setVariable("myvar", "Hello World!");
        // Answer the channel...
        answer();
        // ...say hello...
        // Reproducir un sonido al atender
        streamFile("tt-monkeys");
        channel.exec("Verbose", "2, Hola Mundo");
        verbose("hola mundo", 2);
        System.out.println("call count :" + counter + channel.getChannelStatus());
    } catch (org.asteriskjava.fastagi.AgiHangupException e) {
        System.out.println("Ha colgado el cabrón!!");
        setVariable("myvar", "Ha colgado el cabrón!!");
    }
    counter++;
}
}
```



- Editar fastagi-mapping.properties

```
hello.agi = HelloAgiScript
```

- Compilar Programa

```
javac -cp asterisk-java.jar HelloAgiScript.java
```

- Levantar servidor

```
java -jar asterisk-java.jar
```

- Editar /etc/asterisk/extensions.conf

```
exten =>
```

```
1300,1,Agi(agi://localhost/usr/src/java_agi/hello.agi)
```

Conclusiones

- Conocer herramientas disponibles para el Desarrollo de aplicaciones en conjunto con Asterisk.
- Ampliar los recursos de nuestros sistemas ya existentes.
- Crear nuevos sistemas a partir de nuevas ideas y necesidades.

TICAR 2015

ASTERISK DEVELOPER



Ing. Fernando A. Gonzalez - fagonzalez.nt@gmail.com

Muchas Gracias!