# Descripción del ARA

Nos permite poder tener archivos de configuración o poder tener la funcionalidad que nos da los archivos de configuración de una base de datos. Significa que por ejemplo lo que nos brida el “sip.conf” que básicamente es crear anexos zip, así como el “extensions.conf” que es para el plan de marcado, el “queues.conf” para crear colas y así la funcionalidad que nos da varios archivos la podríamos tener en una base de datos. Esto nos permitirá simplificar la parte de comunicación, facilitando en un futuro la creación de un programa que nos facilite esto.

El ARA (Asterisk RealTime Architecture) tiene dos tipos:

* Static realtime: básicamente se ejecuta con un reloj, es decir ponemos la data en una base de datos y mediante un reloj jalamos la información y lo tiene ya el Asterisk en su memoria, sigue el formato del “.conf”.
  + Su formato es el sig.:

[settings]

archivo.file = odbc,asterisk,table (archivo que queramos = driver a usar, conector “rest\_odbc.conf”, tabla que va a interactuar)

* Dinamic realtime: básicamente es cualquier cambio que hagamos en la base de datos ya sea agregar o quitar se va a reflejar de manera automática, por lo cual no se necesita hacer un reload, reiniciar el Asterisk ni nada por el estilo, solamente modificando un archivo de configuración.
  + Su formato es el sig:

funcionalidad = odbc,asterisk,table (archivo que querase indica la funcionaliad = driver a usar, conector “rest\_odbc.conf”, tabla que va a interactuar)

# Colocando el queue\_log en la base de datos.

Vamos a ver el archivo para configurar el “real time” tanto estático como dinámico, el archivo se llama “extconfig.conf”, para acceder a el primero tendremos que situarnos en el directorio “/etc/asterisk” donde usaremos el sig. comando:

nano extconfig.conf

Como siempre lo primero que hacemos es mover el archivo al bak con el sig. comando:

mv extconfig.conf extconfig.conf.bak

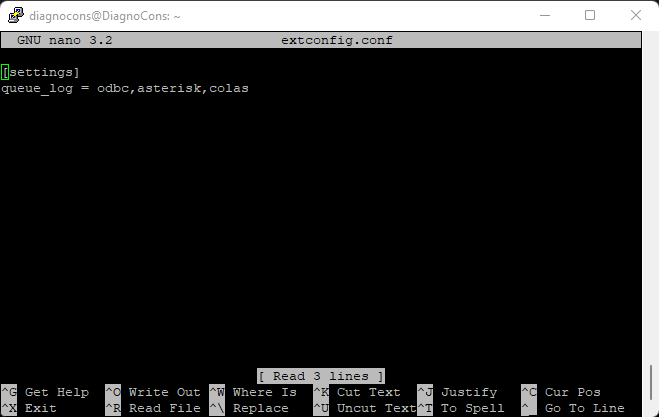
Después entramos de nuevo al “extconfig.conf”

nano extconfig.conf

Donde colocaremos lo sig.:

[settings]

queue\_log = odbc,asterisk,colas



Después guardamos el archivo.

Antes de guardarlo crearemos la tabla “colas” usando el siguiente script:

CREATE TABLE colas(

id bigint(255) unsigned NOT NULL AUTO\_INCREMENT,

time varchar(26) NOT NULL DEFAULT '',

callid varchar(40) NOT NULL DEFAULT '',

queuename varchar(20) NOT NULL DEFAULT '',

agent varchar(20) NOT NULL DEFAULT '',

event varchar(20) NOT NULL DEFAULT '',

data varchar(100) NOT NULL DEFAULT '',

data1 varchar(40) NOT NULL DEFAULT '',

data2 varchar(40) NOT NULL DEFAULT '',

data3 varchar(40) NOT NULL DEFAULT '',

data4 varchar(40) NOT NULL DEFAULT '',

data5 varchar(40) NOT NULL DEFAULT '',

created timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

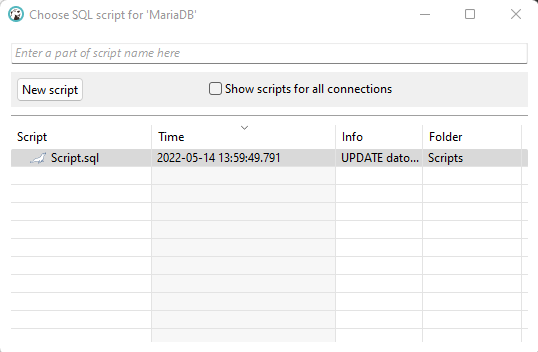
PRIMARY KEY (id),

KEY queue (queuename),

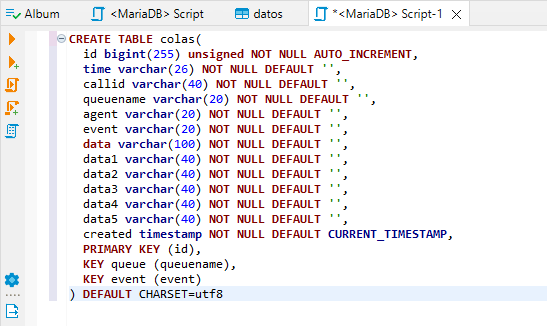
KEY event (event)

) DEFAULT CHARSET=utf8

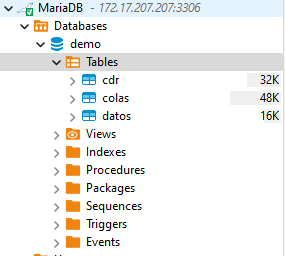
Entramos al dbeaver y en la base de datos le hacemos click derecho, luego click en “Editor SQL” y después en “Script SQL”.



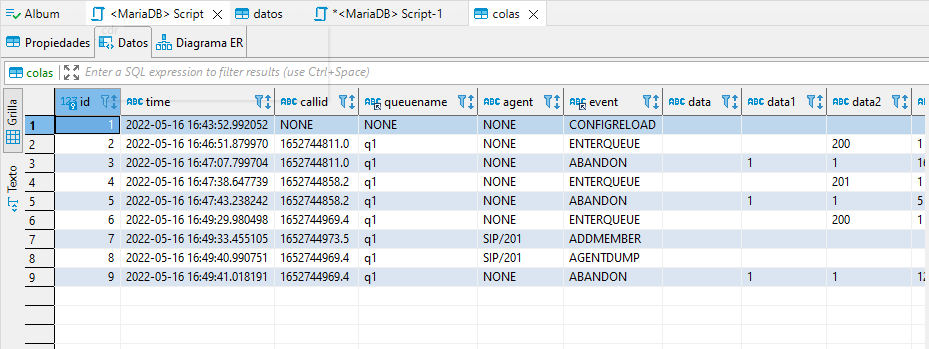
Luego en “New Script” y pegamos el script (no olvidar el USE demo al iniciar).



Luego en la base de datos “demo” en “Tables” le damos click derecho y le hacemos click en “Renovar” y nos aparecerá la nueva tabla “colas”.



Entramos al Asterisk y cargamos la consola con un “reload” y entonces ya debería de poder guardarse esa información en la base de datos. Luego hacemos llamadas a la cola (nuestra extensión es 600 en este caso) y automáticamente debería de guardarse en la base de datos.



# Ejemplo de Static Realtime, colocando queues.conf en la base de datos.

El “Static Realtime” nos permite poner un archivo de configuración en la base de datos, pero solamente va a funcionar como un reloj, es decir cuando pongamos los valores y cuando hagamos reload, recién se van a cargar en el Asterisk.

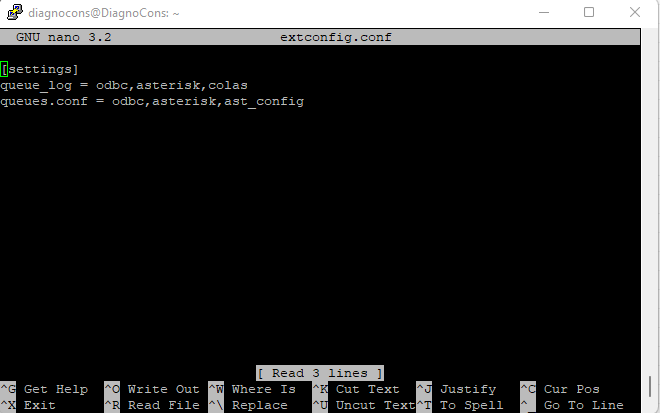
Primero entramos en el “extconfig.conf” con el sig. comando:

nano extenconfig.conf

Una vez ahí, para el tema del estático simplemente colocamos lo sig.:

[settings]

queues.conf = odbc,asterisk,ast\_config



Una vez hecho eso guardamos.

Antes de guardarlo crearemos la tabla “ast\_config” usando el siguiente script:

create table ast\_config (

id int(8) primary key auto\_increment,

cat\_metric int(8), (Es lo que está entre corchetes, a una menor métrica va en la parte superior)

var\_metric int(8), (Es el valor de la métrica a una menor métrica va en la parte superior)

filename varchar(128), (Nombre del archivo)

category varchar(128), (Nombre de la categoría)

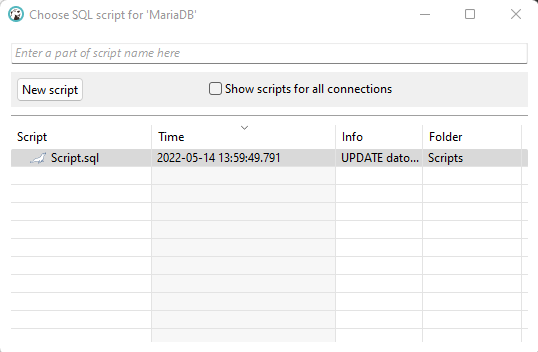
var\_name varchar(128), (Nombre de la variable)

var\_val varchar(128), (Valor de la variable)

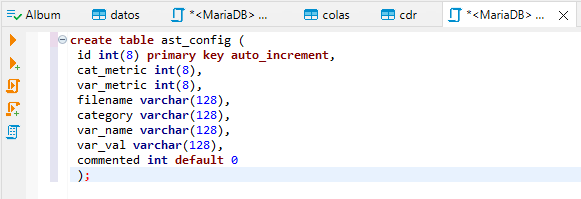
commented int default 0

);

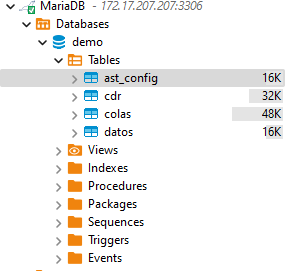
Entramos al dbeaver y en la base de datos le hacemos click derecho, luego click en “Editor SQL” y después en “Script SQL”.



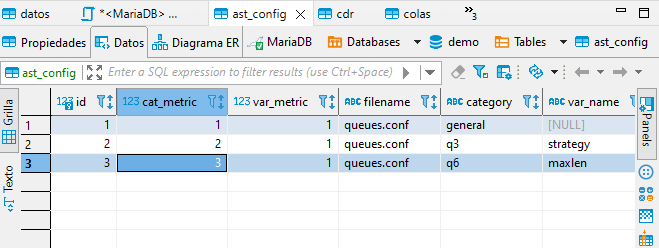
Luego en “New Script” y pegamos el script (no olvidar el USE demo al iniciar).



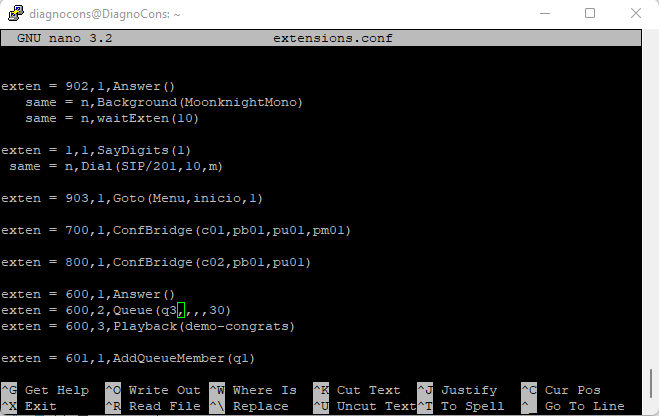
Luego en la base de datos “demo” en “Tables” le damos click derecho y le hacemos click en “Renovar” y nos aparecerá la nueva tabla “ast\_config”.



Una vez hecho eso si queremos agregar o quitar un valor, después de haberlo modificado en la base de datos, hay que hacer un “reload” en la consola Asterisk para que tenga efecto cada vez que hagamos cambios, no será de manera automática.



Antes de probar se tiene que hacer la configuración del nuevo queue, ya que si no dará error.



En el Asterisk ya muestra la nueva configuración queue que hicimos en la tabla, prácticamente esta ignorando el archivo “queues.conf” que se encuentra localmente. Para ver esto usaremos el comando:

queue show “TAB”

