

Системные требования.

- На сервере (здесь и далее примем, что “сервером” может быть как отдельный компьютер, так и виртуальная машина, VPS/VDS и т .д.) должна быть установлена операционная система Ubuntu 22.04.
- IP ATC Asterisk должна быть версии не ниже 1.8.
На Asterisk должна быть настроена запись информации о разговорах в таблицы cel, cdr и queue_log, подробную информацию о необходимых настройках Asterisk вы найдете ниже в разделе «Настройки Asterisk».
- Сервер БД MySQL не ниже версии 5.5.

Вводная

- Данная инструкция написана используя FreePBX версии 13.0.191.11 с Asterisk 13.14.0 на борту (но подходит и для новых версий).
При возникновении каких-либо сложностей/вопросов с более ранними/поздними версиями системы, просьба написать нам на support@fetg.uz.
- Для выполнения описанных ниже действий понадобится:
 - умение подключаться к серверу по ssh
 - умение редактировать файлы в текстовом редакторе (nano/vim/emacs/etc)
 - ясная голова и хорошее настроение 😊

Настройка

- Перезагружаем модуль cdr_adaptive_mysql.so

```
[root@localhost asterisk]# asterisk -r
localhost*CLI> module reload cdr_adaptive_odbc.so
Module 'cdr_adaptive_odbc.so' reloaded successfully.
localhost*CLI>
```

1. Далее настроим внесение realsrc и realdst-полей в cdr.

Отредактируйте /etc/asterisk/extensions_override_freepbx.conf, внеся в него

```
[ext-did-catchall]
include => ext-did-catchall-custom
exten => _,1,Noop(Catch-All DID Match - Found ${EXTEN} - You probably
want a DID for this.)
exten => _,n,Set(_FROM DID=${EXTEN})
exten => _,n,Set(_VISTEPURREALDST=${FROM DID})
exten => _,n,Set(_VISTEPURREALSRC=${CALLERID(num)})
exten => _,n,Set(CDR(realsrc)=${VISTEPURREALSRC})
exten => _,n,Set(CDR(realdst)=${VISTEPURREALDST})
exten => _,n,Goto(ext-did,s,1)

[sub-record-check]
```

```
include => sub-record-check-custom
exten => s,1,GotoIf($[$LEN(${FROMEXTEN})])?initialized)
exten => s,n,Set(__REC_STATUS=INITIALIZED)
exten => s,n,Set(NOW=${EPOCH})
exten => s,n,Set(__DAY=${STRFTIME(${NOW},,%d)})
exten => s,n,Set(__MONTH=${STRFTIME(${NOW},,%m)})
exten => s,n,Set(__YEAR=${STRFTIME(${NOW},,%Y)})
exten => s,n,Set(__Timestr=${YEAR}${MONTH}${DAY}-
${STRFTIME(${NOW},,%H%M%S)})
exten =>
s,n,Set(__FROMEXTEN=${IF($[$LEN(${AMPUSER})])?${AMPUSER}:${IF($[$LEN(
${REALCALLERIDNUM})])?${REALCALLERIDNUM}:unknown}}})
exten =>
s,n,Set(__MON_FMT=${IF($["${MIXMON_FORMAT}"="wav49"]?WAV:${MIXMON_FORMA
T})})
exten => s,n(initialized),Noop(Recordings initialized)
exten => s,n,ExecIf($![${LEN(${ARG3})}]?Set(ARG3=dontcare))
exten => s,n,Set(REC_POLICY_MODE_SAVE=${REC_POLICY_MODE})
exten => s,n,ExecIf($["${BLINDTRANSFER}${ATTENDEDTRANSFER}" !=
""]?Set(REC_STATUS=NO))
exten => s,n(next),GotoIf($[$LEN(${ARG1})]?checkaction)
exten => s,n(recorderror),Playback(something-terribly-wrong,error)
exten => s,n,Hangup
exten => s,n(checkaction),GotoIf($[$DIALPLAN_EXISTS(sub-record-
check,${ARG1})]?sub-record-check,${ARG1},1)
exten => s,n,Noop(Generic ${ARG1} Recording Check - ${FROMEXTEN}
${ARG2})
exten => s,n,Gosub(recordcheck,1(${ARG3},${ARG1},${ARG2}))
exten => s,n,Return()

exten => recordcheck,1,Noop(Starting recording check against ${ARG1})
exten => recordcheck,n,Goto(${ARG1})
exten => recordcheck,n(dontcare),Return()
exten => recordcheck,n(always),Noop(Detected legacy "always" entry.
Mapping to "force")
exten => recordcheck,n(force),Set(__REC_POLICY_MODE=FORCE)
exten => recordcheck,n,GotoIf($["${REC_STATUS}"!="RECORDING"]?startrec)
exten => recordcheck,n,Return()
exten => recordcheck,n(delayed),Noop(Detected legacy "delayed" entry.
Mapping to "yes")
exten => recordcheck,n(yes),ExecIf($["${REC_POLICY_MODE}" = "NEVER" |
"${REC_POLICY_MODE}" = "NO" | "${REC_STATUS}" = "RECORDING"]?Return())
exten => recordcheck,n,Set(__REC_POLICY_MODE=YES)
exten => recordcheck,n,Goto(startrec)
exten => recordcheck,n(no),Set(__REC_POLICY_MODE=NO)
exten => recordcheck,n,Return()
exten => recordcheck,n(never),Set(__REC_POLICY_MODE=NEVER)
exten => recordcheck,n,Goto(stoprec)
exten => recordcheck,n(startrec),Noop(Starting recording: ${ARG2},
${ARG3})
exten => recordcheck,n,Set(AUDIOHOOK_INHERIT(MixMonitor)=yes)
```

```
exten =>
recordcheck,n,ExecIf($[$LEN(${VISTEPRUREALDST})}]?NoOp():Set(CDR(realdst)=${ARG3}))
exten =>
recordcheck,n,ExecIf($[$LEN(${VISTEPRUREALSRC})}]?NoOp():Set(CDR(realsrc)=${FROMEXTEN}))
exten => recordcheck,n,Set(__CALLFILENAME=${ARG2}-${ARG3}-${FROMEXTEN}-${TimestR}-${UNIQUEID})
exten =>
recordcheck,n,MixMonitor(${MIXMON_DIR}${YEAR}/${MONTH}/${DAY}/${CALLFILENAME}.${MON_FMT},abi(LOCAL_MIXMON_ID)${MIXMON_BEEP},${MIXMON_POST})
exten => recordcheck,n,Set(__MIXMON_ID=${LOCAL_MIXMON_ID})
exten => recordcheck,n,Set(__RECORD_ID=${CHANNEL(name)})
exten => recordcheck,n,Set(__REC_STATUS=RECORDING)
exten =>
recordcheck,n,Set(CDR(recordingfile)='${CALLFILENAME}.${MON_FMT}')
exten => recordcheck,n,Return()
exten => recordcheck,n(stoprec),Noop(Stopping recording: ${ARG2}, ${ARG3})
exten => recordcheck,n,Set(__REC_STATUS=STOPPED)
exten => recordcheck,n,System(${AMPBIN}/stoprecording.php "${CHANNEL(name)}")
exten => recordcheck,n,Return()

exten => out,1,Noop(Outbound Recording Check from ${FROMEXTEN} to ${ARG2})
exten => out,n,Set(CDR(realsrc)=${FROMEXTEN})
exten => out,n,Set(CDR(realdst)=${ARG2})
exten => out,n,Set(CUTTEDCHANNEL=${CUT(CHANNEL,,1)})
exten => out,n,Set(CUTTEDCHANNEL=${CUT(CUTTEDCHANNEL,,2)})
exten => out,n,ExecIf($["${FROMEXTEN}" =
"${ARG2}"]?Set(CDR(realsrc)=${CUTTEDCHANNEL}))
exten =>
out,n,Set(RECMODE=${DB(AMPUSE/${FROMEXTEN}/recording/out/external)})
exten => out,n,ExecIf($![${LEN(${RECMODE})}] | "${RECMODE}" =
"dontcare"]?Goto(routewins))
exten => out,n,ExecIf($["${ARG3}" = "never" | "${ARG3}" =
"force"]?Goto(routewins))
exten => out,n(extenwins),Gosub(recordcheck,1(${RECMODE},out,${ARG2}))
exten => out,n,Return()
exten => out,n(routewins),Gosub(recordcheck,1(${ARG3},out,${ARG2}))
exten => out,n,Return()

exten => in,1,Noop(Inbound Recording Check to ${ARG2})
exten => in,n,Set(FROMEXTEN=unknown)
exten =>
in,n,ExecIf($[${LEN(${FROM_DID})}]?Set(_VISTEPRUREALDST=${FROM_DID}))
exten =>
in,n,ExecIf($[${LEN(${CALLERID(num)})}]?Set(FROMEXTEN=${CALLERID(num)}))
exten => in,n,ExecIf($[${LEN(${VISTEPRUREALDST})}]?NoOp("==
```

```
VISTEPRUREALDST has already been set
====") :Set(_VISTEPRUREALDST=${ARG2})) 
exten => in,n,ExecIf($[$LEN(${VISTEPRUREALSRC})}]?NoOp("===
VISTEPRUREALSRC has already been set
====") :Set(_VISTEPRUREALSRC=${FROMEXTEN})) 
exten => in,n,Set(CDR(realsrc)=$VISTEPRUREALSRC)
exten => in,n,Set(CDR(realdst)=$VISTEPRUREALDST)
exten => in,n,Gosub(recordcheck,1(${ARG3},in,$${ARG2})) 
exten => in,n,Return()

exten => exten,1,Noop(Exten Recording Check between ${FROMEXTEN} and
${ARG2})
exten =>
exten,n,ExecIF($[$LEN(${VISTEPRUREALDST})}]?NoOP():Set(CDR(realdst)=$
${ARG2})) 
exten =>
exten,n,ExecIF($[$LEN(${VISTEPRUREALSRC})}]?NoOP():Set(CDR(realsrc)=$
${FROMEXTEN})) 
exten => exten,n,Set(CDR(cnum)=$FROMEXTEN)
exten =>
exten,n,Set(CALLTYPE=${IF($[$LEN(${FROM_DID})}]?external:internal)}) 
exten =>
exten,n,ExecIf(${LEN(${CALLTYPE_OVERRIDE})}?Set(CALLTYPE=${CALLTYPE_OVE
RRIDE})) 
exten =>
exten,n,Set(CALLEE=${DB(AMPUSER/${ARG2}/recording/in/${CALLTYPE}))}
exten => exten,n,ExecIf($![${LEN(${CALLEE})}]?Set(CALLEE=dontcare)) 
exten => exten,n,GotoIf($["${CALLTYPE}"=="external"]?callee)
exten => exten,n,GotoIf($["${CALLEE}"=="dontcare"]?caller)
exten =>
exten,n,ExecIf($[$LEN(${DB(AMPUSER/${FROMEXTEN}/recording/priority)})}]
?Set(CALLER_PRI=${DB(AMPUSER/${FROMEXTEN}/recording/priority)}):Set(CA
LLER_PRI=0)) 
exten =>
exten,n,ExecIf($[$LEN(${DB(AMPUSER/${ARG2}/recording/priority)})}]?Set
(CALLEE_PRI=${DB(AMPUSER/${ARG2}/recording/priority)}):Set(CALLEE_PRI=0
))
exten =>
exten,n,GotoIf($["${CALLER_PRI}"=="${CALLEE_PRI}"]?${REC_POLICY}:${IF($[
${CALLER_PRI}>${CALLEE_PRI}]?caller:callee)}) 
exten =>
exten,n(callee),Gosub(recordcheck,1(${CALLEE}, ${CALLTYPE}, ${ARG2})) 
exten => exten,n,Return()
exten =>
exten,n(caller),Set(RECMODE=${DB(AMPUSER/${FROMEXTEN}/recording/out/int
ernal)}) 
exten => exten,n,ExecIf($![${LEN(${RECMODE})}]?Set(RECMODE=dontcare)) 
exten =>
exten,n,ExecIf($["${RECMODE}"=="dontcare"]?Set(RECMODE=${CALLEE})) 
exten => exten,n,Gosub(recordcheck,1(${RECMODE}, ${CALLTYPE}, ${ARG2})) 
exten => exten,n,Return()
```

```
exten => conf,1,Noop(Conference Recording Check ${FROMEXTEN} to ${ARG2})
exten => conf,n,Gosub(recconf,1(${ARG2},${ARG2},${ARG3}))
exten => conf,n,Return()

exten => page,1,Noop(Paging Recording Check ${FROMEXTEN} to ${ARG2})
exten =>
page,n,GosubIf($["${REC_POLICY_MODE}"=="always"]?recconf,1(${ARG2},${FROMEXTEN},${ARG3}))
exten => page,n,Return()

exten => recconf,1,Noop(Setting up recording: ${ARG1}, ${ARG2},
${ARG3})
exten =>
recconf,n,Set(__CALLFILENAME=${IF($[$CONFBRIDGE_INFO(parties,${ARG2})]
]?${DB(RECCONF/${ARG2})}:${ARG1}-${ARG2}-${ARG3}-${TimestR}-
${UNIQUEID}))}
exten =>
recconf,n,ExecIf($![${CONFBRIDGE_INFO(parties,${ARG2})}]?Set(DB(RECCONF
/${ARG2})=${CALLFILENAME}))}
exten =>
recconf,n,Set(CONFBRIDGE(bridge,record_file)={`${MIXMON_DIR}${YEAR}/${MONTH}/
${DAY}/${CALLFILENAME}.${MON_FMT}`})
exten => recconf,n,ExecIf($["${ARG3}"!="always"]?Return())
exten => recconf,n,Set(CONFBRIDGE(bridge,record_conference)=yes)
exten => recconf,n,Set(__REC_STATUS=RECORDING)
exten =>
recconf,n,Set(CDR(recordingfile)=$[ ${IF($[$CONFBRIDGE_INFO(parties,${ARG
2})]}?${CALLFILENAME}.${MON_FMT}: ${CALLFILENAME}.${MON_FMT})])
exten => recconf,n,Noop(${MIXMONITOR_FILENAME})
exten => recconf,n,Set(CHANNEL(hangup_handler_push)=sub-record-hh-
check,s,1)
exten => recconf,n,Return()

exten => req,1,Noop(Setting up recording: ${ARG1}, ${ARG2}, ${ARG3})
exten => req,n,Set(AUDIOHOOK_INHERIT(MixMonitor)=yes)
exten =>
req,n,Set(MONITOR_FILENAME=${MIXMON_DIR}${YEAR}/${MONTH}/${DAY}/${CALL
FILENAME})
exten =>
req,n,MixMonitor(${MONITOR_FILENAME}.${MON_FMT}, ${MONITOR_OPTIONS}${MIX
MON_BEEP}, ${MIXMON_POST})
exten => req,n,Set(__REC_STATUS=RECORDING)
exten => req,n,Set(CDR(recordingfile)=$[ ${CALLFILENAME}.${MON_FMT})}
exten => req,n,Return()

exten => parking,1,Noop(User ${ARG2} picked up a parked call)
exten => parking,n,Set(USER=${ARG2})
exten => parking,n,ExecIf($![${LEN(${ARG2})}]?Set(USER=unknown))
exten =>
parking,n,Set(RECMODE=${DB(AMPUSER/${ARG2}/recording/out/internal)})
```

```

exten => parking,n,ExecIf($![${LEN(${RECMODE})}])?Set(RECMODE=dontcare)
exten => parking,n,Gosub(recordcheck,1(${RECMODE},parked,${USER}))
exten => parking,n,Return()

;--- end of [sub-record-check] ---; ---

```

и выполните команду

```
dialplan reload
```

в консоли Asterisk

Код проверен на FreePBX 13.0.191.11 и 14.0.2.10 .

Если мажорная (13 в данном примере) версия FreePBX у вас отличается, то контекст [sub-record-check] может быть немного другим. Напишите нам на support@fetg.uz и мы поможем с написанием диалплана данного контекста.

2. Теперь научим FreePBX писать табличку queue_log.

Отправляемся в web-интерфейс и создаем новый файл под именем **extconfig.conf**, кликнув Add New File.

Внести в него нужно лишь одну строчку

```

[settings]
queue_log => odbc,asteriskcdrdb

```

Скрин:

The screenshot shows the Asterisk Configuration File Editor interface. At the top, there's a navigation bar with tabs: Admin, Applications, Connectivity, Dashboard, Reports, Settings, UCP, and a red 'Apply Config' button. Below the navigation bar, the title 'Configuration File Editor' is displayed. On the left, there's a sidebar titled 'Asterisk Custom Configuration Files' containing a list of existing files like ari_additional_custom.conf, ari_general_custom.conf, etc. In the main area, there's a blue button '+ Add New File'. To its right, a text editor window titled 'Working on extconfig.conf' shows the configuration code. A green message bar above the editor says 'Saved. Make sure to Apply Config so that Asterisk will pickup your changes'. At the bottom right of the editor window, there are 'Save' and 'Delete' buttons.

*не забываем про Save и Apply Configs

После выполнения данных действий и совершения звонков в очереди, запрос из консоли mysql

```
SELECT * FROM queue_log;
```

должен вернуть вам нечто подобное

```
mysql> select * from queue_log where time between '2017-04-29 21:13:58' and '2017-04-29 21:14:01';
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | time           | callid | queuename | agent   | event    | data1  | data2  | data3  | data4  | data5  |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 7  | 2017-04-29 21:13:58 | 1493489637.71 | 100500  | NONE    | DID      | 79234776763 |        |        |        |        |
| 8  | 2017-04-29 21:13:58 | 1493489637.71 | 100500  | NONE    | ENTERQUEUE |          | 79130751580 | 1       |        |        |
| 9  | 2017-04-29 21:13:59 | 1493489637.71 | 100500  | Eugene R | CONNECT   | 1       | 1493489638.72 | 1       |        |        |
| 10 | 2017-04-29 21:14:01 | 1493489637.71 | 100500  | Eugene R | COMPLETECALLER | 1       | 2       | 1       |        |        |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
```

4 rows in set (0.00 sec)

*на запрос в скрине не ориентируйтесь, там мы специально добавили условие, чтобы сократить выхлоп

3. Для входа в систему статистики используйте данные авторизации (Email-адрес/Пароль для авторизации) из регистрационного письма.

На этом настройка завершена, приятного пользования!

По любым вопросам/неполадкам/пр. пишите нам на support@fetg.uz или прямо из интерфейса, в меню Help Desk.

Некоторые замечания по части MySQL.

1. В ходе своей работы установщик подключается к БД Asterisk, для чего ему необходимы имя пользователя и пароль. Создать их можно, например, вот так:

```
mysql -uroot -p
Enter password:
Welcome TO the MariaDB monitor.  Commands END WITH ; OR \g.
Your MariaDB connection id IS 3359
Server version: 5.5.60-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab AND others.

TYPE 'help;' OR '\h' FOR help. TYPE '\c' TO clear the CURRENT INPUT
statement.

MariaDB [(NONE)]> GRANT ALL PRIVILEGES ON *.* TO 'fetguz'@'%' \
IDENTIFIED BY 'MySuperPassword111' WITH GRANT OPTION;
FLUSH privileges;
quit
```

2. Также проверьте, что MySQL слушает не только 127.0.0.1, но и IP-адрес, по которому к нему будет обращаться система статистики.
За это в настройках MySQL отвечает параметр **bind-address**, проще всего закомментировать эту строку в my.cnf и перезагрузить демон MySQL, тогда сервис будет слушать все IP-адреса сервера.

From:

<https://wiki.fetg.uz/> - Система статистики call центров на IP-ATC Asterisk (FreePBX)

Permanent link:

https://wiki.fetg.uz/doku.php?id=install_local_version_fpbx&rev=1684213626

Last update: 2023/05/16 10:07

