

Инструкция по подключению FreePBX к облачному сервису системы статистики ViStep.RU

Вводная

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

Настройка

В поставке по умолчанию (что называется «из коробки») FreePBX ведет БД asteriskcdrdb, а точнее табличку cdr, не совсем так, как это нужно для правильной работы системы статистики ViStep.RU. Также она совсем не пишет в таблицу queue_log, что является необходимым. Поэтому нам потребуется выполнить несколько шагов настройки, для достижения нужного поведения FreePBX.

1. Начнем с таблицы cdr.

Подключимся к серверу по ssh, а затем к консоли MySQL-сервера

```
[root@localhost ~]# mysql asteriskcdrdb
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1686
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.
```

2. Добавим табличке cdr новые поля id и filename и повесим триггер на INSERT для заполнения filename

```
ALTER TABLE cdr ADD COLUMN filename VARCHAR(120) DEFAULT 'none';
UPDATE cdr SET filename=recordingfile;
ALTER TABLE cdr ADD COLUMN id INT(11) AUTO_INCREMENT PRIMARY KEY;
DELIMITER $$;
CREATE TRIGGER `before_insert_cdr` BEFORE INSERT ON `cdr` FOR EACH ROW
```

```
BEGIN
SET NEW.filename=NEW.recordingfile;
END $$ 
DELIMITER ;
```

При успешном выполнении запросов все должно выглядеть примерно вот так:

3. Совершим вызов и проверим, что поле filename заполняется
4. Теперь нам необходимо изменить логику заполнения полей dst и src в таблице cdr так, чтобы в них всегда попадали те номера, на которые/с которых поступил вызов (например, не номер оператора очереди, а наш городской номер, на который пришел вызов извне для dst или не номер транка, а номер внутреннего абонента для src). Для этого отредактируем файл cdr_adaptive_odbc.conf (FreePBX не перетрет его после рестарта, поэтому можно смело его редактировать), добавив в самом низу строку

```
alias realdst => dst
alias realsrc => src
```

В итоге он примет вид:

5. Дабы наши действия возымели силу, модулю cdr_adaptive_mysql.so нужно подсказать, что его конфиг изменился. Подключаемся к консоли asterisk и релоадим модуль вот так:

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

6. Пришло время последнего, но отнюдь не маловажного шага в разрезе конфигурирования таблицы cdr.
Есть два варианта (первый может работать не везде, но он проще) внесения нужных данных в поля src и dst:

Первый.

В web-интерфейсе FreePBX отправимся по пути «Admin→ Config Edit», где в extensions_custom.conf внесем:

```
[from-internal]
exten => _X.,1,GoSub(set_right_src_dst_from-
internal,start,1(${EXTEN}, ${CALLERID(num)}) )

[from-pstn-custom]
exten => _X.,1,GoSub(set_right_src_dst_from-
pstn,start,1(${EXTEN}, ${CALLERID(num)}) )

[set_right_src_dst_from-pstn]
exten => start,1,Set(_FROMPSTNCALL=1)
same => n,Set(CDR(realdst)=$ARG1)
same => n,Set(CDR(realsrc)=$ARG2)
```

```

same => n,Return()

[set_right_src_dst_from-internal]
exten =>
start,1,ExecIf(${FROMPSTNCALL}?Set(CDR_PROP(disable)=true):Set(CDR(realdst)=${ARG1}))
    same => n,Set(CDR(realsrc)=${ARG2})
    same => n,Return()

```

А вот и скрин, для наглядности

Configuration File Editor

Working on extensions_custom.conf

Saved. Make sure to Apply Config so that Asterisk will pickup your changes

```

1 [from-internal]
2 exten => _X.,1,GoSub(set_right_src_dst_from-internal,start,1(${EXTEN}, ${CALLERID(num)}))
3
4 [from-pstn-custom]
5 exten => _X.,1,GoSub(set_right_src_dst_from-pstn,start,1(${EXTEN}, ${CALLERID(num)}))
6
7 [set_right_src_dst_from-pstn]
8 exten => start,1,Set(_FROMPSTNCALL=1)
9 same => n,Set(CDR(realdst)=${ARG1})
10 same => n,Set(CDR(realsrc)=${ARG2})
11 same => n,Return()
12
13 [set_right_src_dst_from-internal]
14 exten => start,1,ExecIf(${FROMPSTNCALL}?Set(CDR_PROP(disable)=true):Set(CDR(realdst)=${ARG1}))
15 same => n,Set(CDR(realsrc)=${ARG2})
16 same => n,Return()
17

```

Save Delete



FreePBX
let freedom ring™
FreePBX is a registered trademark of
Sangoma Technologies Inc.
FreePBX 13.0.191.11 is licensed under the GPL
Copyright© 2007-2017



В завершении кликнем Save, затем Apply Config, позвоним на АТС, с АТС вовне и проверим корректность заполнения полей dst и src.

Второй.

Скопируйте содержимое диалплана из /etc/asterisk/extensions_additional.conf со строки

```
[sub-record-check]
```

до строки

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

включительно, вставьте скопированный код в /etc/asterisk/extensions_override_freespbx.conf и отредактируйте, добавив установку полей realsrc (src) и realdst (dst), например, так:

```

[sub-record-check]
include => sub-record-check-custom
exten => s,1,GotoIf(${LEN(${FROMEXTEN})}>0)?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:${MIXMON_FORMAT})})
exten => s,n(initialized),Noop(Recordings initialized)
exten => s,n,ExecIf(${!${LEN(${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},
${ARG3}))
exten => recordcheck,n,Set(AUDIOHOOK_INHERIT(MixMonitor)=yes)
exten => recordcheck,n,Set(__Callfilename=${ARG2}-${ARG3}-${Fromexten}-
${Timestr}-${Uniqueid})
exten =>
recordcheck,n,MixMonitor(${Mixmon_Dir}${YEAR}/${Month}/${Day}/${Callfil-
ename}.${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
${ARG2})
exten =>
out,n,Set(RECMODE=${DB(AMPUSER/${FROMEXTEN}/recording/out/external)})
exten => out,n,ExecIf(${!${LEN(${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(${CALLERID(num)})})?Set(FROMEXTEN=${CALLERID(num)})
)
exten => in,n,Gosub(recordcheck,1(${ARG3},in,${ARG2}))
exten => in,n,Return()

exten => exten,1,Noop(Exten Recording Check between ${FROMEXTEN} and
${ARG2})
exten =>
exten,n,ExecIf(${FROMPSTNCALL}?Set(CDR_PROP(disable)=true):Set(CDR(real
dst)=${ARG2}))
exten => exten,n,Set(CDR(realsrc)=${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)
```

Last update:

2018/07/27 configure_freepbx_for_cloud_version https://wiki.fetg.uz/doku.php?id=configure_freepbx_for_cloud_version&rev=1532663873
08:57

```
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},${FR0
MEXTEN},${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}/${MON
TH}/${DAY}/${CALLTYPE}.${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,${ARG2})}]?${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 => recq,1,Noop(Setting up recording: ${ARG1}, ${ARG2}, ${ARG3})
exten => recq,n,Set(AUDIOHOOK_INHERIT(MixMonitor)=yes)
exten =>
recq,n,Set(MONITOR_FILENAME=${MIXMON_DIR}${YEAR}/${MONTH}/${DAY}/${CALL
FILENAME})
exten =>
recq,n,MixMonitor(${MONITOR_FILENAME}.${MON_FMT}, ${MONITOR_OPTIONS}${MI
XMON_BEEP}, ${MIXMON_POST})
exten => recq,n,Set(__REC_STATUS=RECORDING)
exten => recq,n,Set(CDR(recordingfile)=${CALLFILENAME}.${MON_FMT})
exten => recq,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] ---;

```

после чего выполните команду

```
dialplan reload
```

в консоли Asterisk

Не копируйте наш код, используйте тот, что находится в вашей FreePBX

Если сомневаетесь **где именно** вам нужно вставить команды установки полей - напишите нам на support@vistep.ru.

1. С cdr разобрались, дело за queue_log. Эта таблица нужна нам для того, чтобы понимать какие события происходили в очередях Asterisk. Перво-наперво создадим ее

```

CREATE TABLE `queue_log` (
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  `time` TIMESTAMP NULL DEFAULT '0000-00-00 00:00:00',
  `callid` VARCHAR(32) NOT NULL DEFAULT '',
  `queuename` VARCHAR(32) NOT NULL DEFAULT '',
  `agent` VARCHAR(32) NOT NULL DEFAULT '',
  `event` VARCHAR(32) NOT NULL DEFAULT '';

```

Last update:

2018/07/27 configure_freepbx_for_cloud_version https://wiki.fetg.uz/doku.php?id=configure_freepbx_for_cloud_version&rev=1532663873
08:57

```
`data1` VARCHAR(100) NOT NULL DEFAULT '' ,  
`data2` VARCHAR(100) NOT NULL DEFAULT '' ,  
`data3` VARCHAR(100) NOT NULL DEFAULT '' ,  
`data4` VARCHAR(100) NOT NULL DEFAULT '' ,  
`data5` VARCHAR(100) NOT NULL DEFAULT '' ,  
PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

2. Научим FreePBX(читай Asterisk) писать туда все то, что нам нужно.

Отправляемся в web-интерфейс по знакомой дорожке 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: *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 *на запрос в скрине не ориентируйтесь, там я специально добавил условие, чтобы сократить выхлоп

3. С препарированием нашей АТС по части БД закончили!

Переходим к настройке синхронизации БД и файлов записей разговоров в облако ViStep.RU. Для этого воспользуйтесь инструкцией - https://wiki.vistep.ru/doku.php?id=cloud_daemon_sync

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

На этом все настройки завершены.

Если вы все сделали правильно, то через какое-то время (зависит от размера БД) сможете воспользоваться всеми отчетами системы статистики ViStep.RU в своем личном кабинете.

Повторюсь, если у вас остались вопросы и/или есть предложения/замечания, пишите нам на support@vistep.ru.

From:

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

Permanent link:

https://wiki.fetg.uz/doku.php?id=configure_freepbx_for_cloud_version&rev=1532663873

Last update: 2018/07/27 08:57