

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 ведет записи 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 релоадем 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.

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)}))

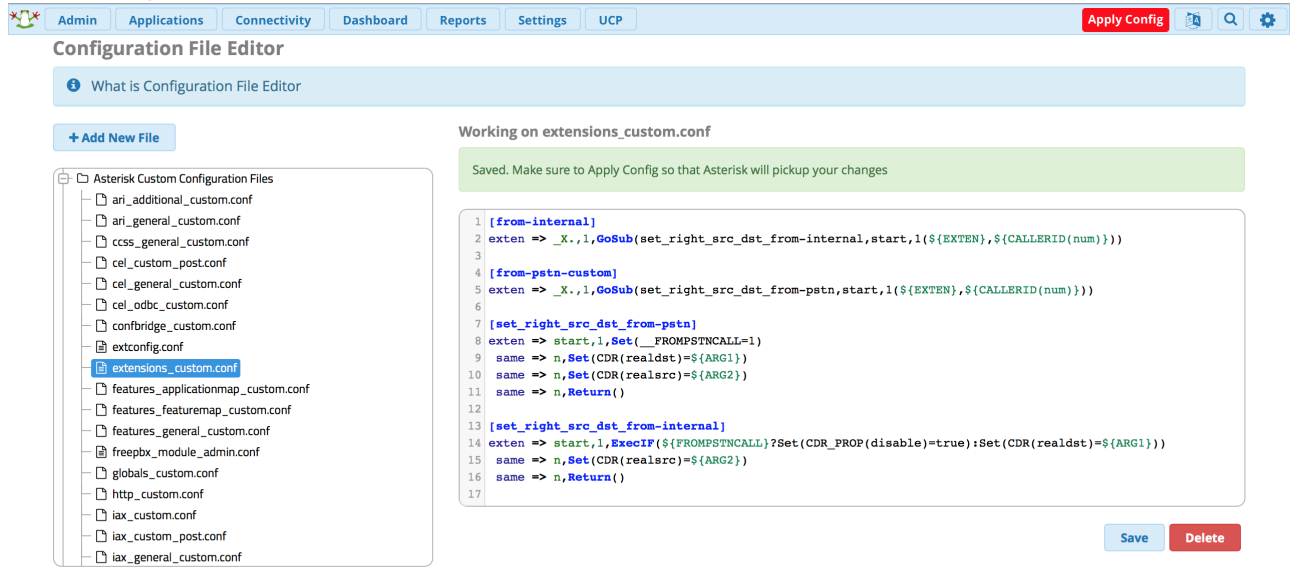
[from-pstn-custom]
exten => _X.,1,GoSub(set_right_src_dst_from-pstn,start,1(${EXTEN},${CALLERID(num)}))

[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(${FROMPSTNCALL}?Set(CDR_PROP(disable)=true):Set(CDR(real
dst)=${ARG1}))
same => n,Set(CDR(realsrc)=${ARG2})
same => n,Return()
```

А вот и скрин, для наглядности



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


7. 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 '',
  `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;
```

8. 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


9. С препарированием нашей АТС по части БД закончили!

Переходим к настройке синхронизации БД и файлов записей разговоров в облако ViStep.RU.

После того, как вы получите ссылку для входа в систему, логин и пароль, вам необходимо авторизоваться и кликнуть меню «Настройки→Основные»

В скринах это будет выглядеть так:  

10. Укажите путь к файлам записей разговоров (в FreePBX это обычно

/var/spool/asterisk/monitor/) и нажмите кнопку «Сохранить», после чего появится возможность скачать архив со скриптом синхронизации 

11. Загрузите на сервер скачанный архив в папку /opt (например, используя scp или FileZilla)

12. Далее просто листинг команд, дабы не перегружать текст

```
[root@localhost ~]# cd /opt/
[root@localhost opt]# mkdir -p stat.vistep.ru
[root@localhost opt]# mv Fb38f1218c6.zip stat.vistep.ru/
[root@localhost opt]# cd stat.vistep.ru/
[root@localhost stat.vistep.ru]# unzip Fb38f1218c6.zip
[root@localhost stat.vistep.ru]# rm -f Fb38f1218c6.zip
```

по окончании у нас должно получиться так 

13. В скрипте **stat.vistep.ru.js** уже практически все учтено, нам остается только вбить данные доступа к БД. Откроем файл **stat.vistep.ru.js** в любимом текстовом редакторе и внесем нужные изменения в строках 393-397

```
// Исправляем все что ниже под себя
"dbhost": "localhost",
"dbuser": "freepbxuser",
"dbpassword": "1230239dkl8c5ef2rf497aeb523f1d117cdb",
"db": "asteriskcdrdb", // <--- имя БД
"timezone": "Europe/Moscow", // <--- часовой пояс
"fileMask": /\.*/ // <--- регулярка, если нужно синкать все файлы

// далее ничего не трогаем
```

*Логин и пароль для доступа к БД вы найдете в файле
/etc/asterisk/res_odbc_additional.conf

14. Мы почти у цели, остался последний штрих.

Скрипт нужно запустить, но не только запустить, а еще и форкнуть его (говоря по-русски нужно чтобы скрипт остался в работе после того, как мы отключимся от сервера, т.е. работал бы как любой другой демон в ОС).

Такой функционал обеспечит замечательный [proccess meneger pm2](#).

Установим pm2 и отдадим в его руки управление скриптом

```
npm install pm2 -g
pm2 start /opt/stat.vistep.ru/stat.vistep.ru.js --name "ViStep.RU stat"
pm2 startup centos6
pm2 save
```

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.vistep.ru/> - Система статистики call центров на IP-ATC Asterisk (FreePBX)

Permanent link:

https://wiki.vistep.ru/doku.php?id=configure_freepbx_for_cloud_version&rev=1505356480

Last update: **2017/09/14 09:34**

