

Системные требования.

- На сервере (здесь и далее примем, что “сервером” может быть как отдельный компьютер, так и виртуальная машина, VPS/VDS и т .д.
Также отметим: docker-контейнер с локальной версией системы статистики может быть запущен как на том же сервере, что и АТС Asterisk, так и на отдельном сервере) должна быть установлена операционная система Linux семейства Debian не ниже 7.0 (Ubuntu 14.04 и выше), CentOS не ниже 6.5 или другие дистрибутивы с актуальной базой репозитория.
- IP АТС 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 5.1.
- Docker версии 1.7.1 и выше (версии ниже нужно проверять дополнительно).
Установка docker в различных ОС подробно расписана в официальной документации: [Debian](#), [Ubuntu](#), [CentOS](#).
Для установки в FreePBX 13 можете воспользоваться [нашей инструкцией](#)

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:

```
mysql> alter table cdr add column filename varchar(120) default 'none';
Query OK, 2 rows affected (0.17 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> DELIMITER $$
mysql> CREATE TRIGGER `before_insert_cdr` BEFORE INSERT ON `cdr` FOR EACH ROW
-> BEGIN
-> SET NEW.filename=NEW.recordingfile;
-> END $$
Query OK, 0 rows affected (0.09 sec)

mysql> DELIMITER ;
mysql>
```

- 3. Совершим вызов и проверим, что поле filename заполняется

```
mysql> select filename from cdr;
+-----+
| filename |
+-----+
| none |
| none |
| in-74951341307-79130751580-20170429-133530-1493462130.4.wav |
| in-74951341307-79130751580-20170429-133530-1493462130.4.wav |
+-----+
4 rows in set (0.00 sec)
```

- 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:

```
[root@localhost asterisk]# cat cdr_adaptive_odbc.conf
[asteriskcdrdb]
connection=asteriskcdrdb
loguniqueid=yes
table=cdr
alias start => calldate
alias realdst => dst
alias realsrc => src
```

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` - внесение правильных `src`- и `dst`-полей.

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

```
[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_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},
${ARG3})
exten => recordcheck,n,Set(AUDIOHOOK_INHERIT(MixMonitor)=yes)
exten =>
recordcheck,n,ExecIF($[${LEN(${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}/${CALLFIL
ENAME}.${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
${ARG2})
exten => out,n,Set(CDR(realsrc)=${FROMEXTEN})
exten => out,n,Set(CDR(realdst)=${ARG2})
exten =>
out,n,Set(RECMODE=${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=${CALLERID(num)})
)
exten => in,n,Set(_VISTEPRUREALDST=${ARG2})
exten => in,n,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
${ARG2})
exten => conf,n,Gosub(reconf,1(${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"?reconf,1(${ARG2},${FRO
MEXTEN},${ARG3}))
exten => page,n,Return()

exten => reconf,1,noop(Setting up recording: ${ARG1}, ${ARG2},
${ARG3})
exten =>
reconf,n,Set(__CALLFILENAME=${IF(${CONFBRIDGE_INFO(parties,${ARG2})}
)?${DB(RECONF/${ARG2})}:${ARG1}-${ARG2}-${ARG3}-${TIMESTR}-
${UNIQUEID}}))
exten =>
reconf,n,ExecIf(!${CONFBRIDGE_INFO(parties,${ARG2})})?Set(DB(RECONF
/${ARG2})=${CALLFILENAME}))
exten =>
reconf,n,Set(CONFBRIDGE(bridge,record_file)=${MIXMON_DIR}${YEAR}/${MON
TH}/${DAY}/${CALLFILENAME}.${MON_FMT})
exten => reconf,n,ExecIf("${ARG3}"!="always"?Return())

```

```

exten => reconf,n,Set(CONFBRIDGE(bridge,record_conference)=yes)
exten => reconf,n,Set(__REC_STATUS=RECORDING)
exten =>
reconf,n,Set(CDR(recordingfile)=${IF(${CONFBRIDGE_INFO(parties,${ARG
2})}}?${CALLFILENAME}.${MON_FMT}:${CALLFILENAME}.${MON_FMT}))
exten => reconf,n,Noop(${MIXMONITOR_FILENAME})
exten => reconf,n,Set(CHANNEL(hangup_handler_push)=sub-record-hh-
check,s,1)
exten => reconf,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

Код проверен на FreePBX 13.0.191.11.

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

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 '',

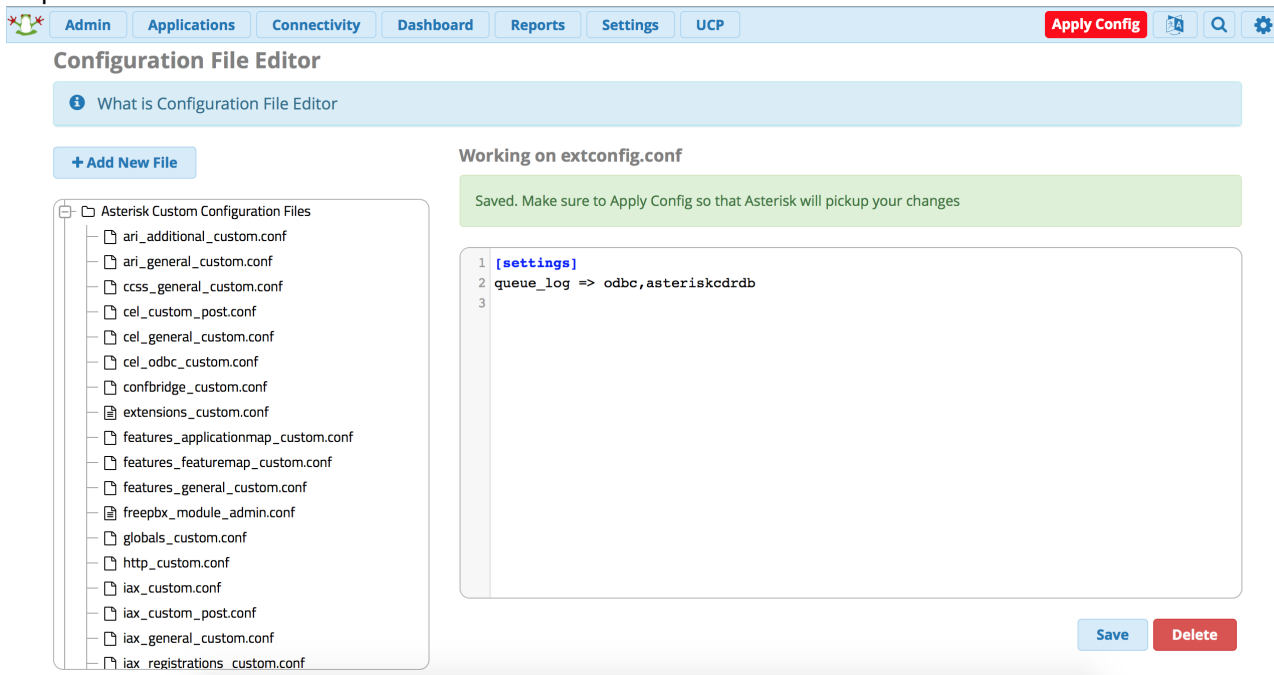
```

```
`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:

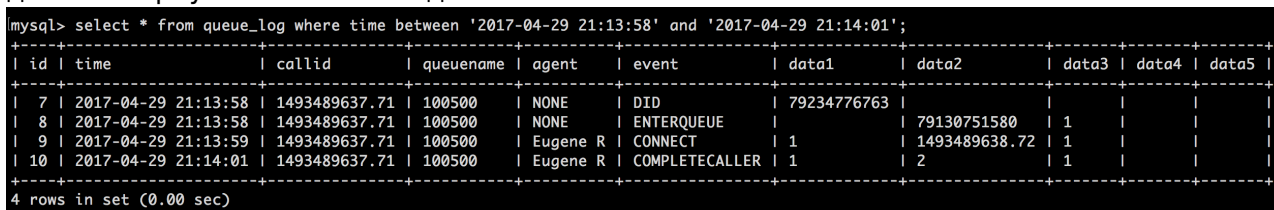


*не забываем про Save и Apply Configs

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

Установка, настройка и запуск контейнера.

1. Для начала нужно провести первичную подготовку MySQL-базы данных Asterisk. Зайдите на сервер MySQL и создайте нового пользователя с полными правами, например вот так:

```
GRANT ALL PRIVILEGES ON *.* TO 'vistep.ru'@'%' \
IDENTIFIED BY 'MySuperPassword111' WITH GRANT OPTION;
FLUSH privileges;
quit
```

Этот шаг необходим для работы мастера настройки БД, речь о котором пойдет несколько позже.

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`, тогда `mysql`-сервер будет слушать на всех IP-адресах сервера.

2. Теперь зайдите на сервер, где предполагаете запускать `docker`-контейнер, по `ssh`. Скачайте архив контейнера по предоставленной ссылке, например (это пример!!!!, корректная ссылка высылается на почту после оплаты):

```
wget https://vistep.ru/stat.vistep.ru_local_1.0.3.tgz
```

Если в вашей системе нет `wget`, то воспользуйтесь вот этой командой:

```
curl -o stat.vistep.ru_local_1.0.3.tgz
https://vistep.ru/stat.vistep.ru_local_1.0.3.tgz
```

3. Далее перейдите в папку с архивом контейнера (здесь и далее это `/root/`) и загрузите его в `docker`

```
docker load -i stat.vistep.ru_local_1.0.3.tgz
```

Проверьте все ли в порядке, команда

```
docker images
```

должна вернуть нечто подобное

```
docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
stat.vistep.ru_local_1.0.3  latest      179e06b24473     56 minutes ago  618 MB
```

4. Следующим пунктом будет настройка возможности прослушивать и скачивать файлы разговоров в интерфейсе статистики. Как и было описано ранее, вы можете запустить `docker`-контейнер с ПО как на сервере с АТС, так и на отдельном сервере. Для запуска `docker` там же, где и Asterisk, вам нужно выполнить такую команду:

```
docker run --name=stat.vistep.ru_local_1.0.3 -p 8768:80 -v
```

```
/var/spool/asterisk/monitor:/var/www/html/public/user/УНИКАЛЬНЫЙ_КАТАЛОГ_ПОЛЬЗОВАТЕЛЯ/some/path/to/files/ --restart=always -d stat.vistep.ru_local_1.0.3
```

В консоле должно отобразиться подобное:

```
root@debian-stat:~# docker run --name=stat.vistep.ru_local_1.0.3 -p 80:80 -v /root/config.php:/var/www/html/app/config.php stat.vistep.ru_lo
cal_1.0.3
2017-04-21 06:38:48,183 CRIT Set uid to user 0
2017-04-21 06:38:48,213 INFO RPC interface 'supervisor' initialized
2017-04-21 06:38:48,213 CRIT Server 'unix_http_server' running without any HTTP authentication checking
2017-04-21 06:38:48,213 INFO supervisord started with pid 7
2017-04-21 06:38:49,216 INFO spawned: 'get_geo_redis.php' with pid 10
2017-04-21 06:38:49,217 INFO spawned: 'php-fpm5' with pid 11
2017-04-21 06:38:49,218 INFO spawned: 'redis' with pid 12
2017-04-21 06:38:49,219 INFO spawned: 'nginx' with pid 13
2017-04-21 06:38:50,229 INFO success: get_geo_redis.php entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
2017-04-21 06:38:50,229 INFO success: php-fpm5 entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
2017-04-21 06:38:50,229 INFO success: redis entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
2017-04-21 06:38:50,229 INFO success: nginx entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
2017-04-21 06:38:51,296 INFO exited: get_geo_redis.php (exit status 0; expected)
```

УНИКАЛЬНЫЙ_КАТАЛОГ_ПОЛЬЗОВАТЕЛЯ - будет выслан Вам вместе с корректной ссылкой для скачивания контейнера

Если же запуск осуществляется на отдельном сервере, то сначала примонтируйте папку с файлами с сервера АТС на сервер с docker по NFS. Вот пример такой конфигурации: файл **/etc/exports** на сервере с Asterisk:

```
/records/mp3/
192.168.0.0/255.255.255.0(rw,insecure,nohide,all_squash,anonuid=1001,anongid=1001,no_subtree_check)
```

а на сервере с docker, в **/etc/fstab**

```
ASTERISK_IP_ADDRESS:/records/mp3 /root/records nfs
rsize=8192,wsizе=8192 0 0
```

*здесь /records/mp3 - это путь к файлам записей на АТС
Затем выполните запуск уже знакомой командой

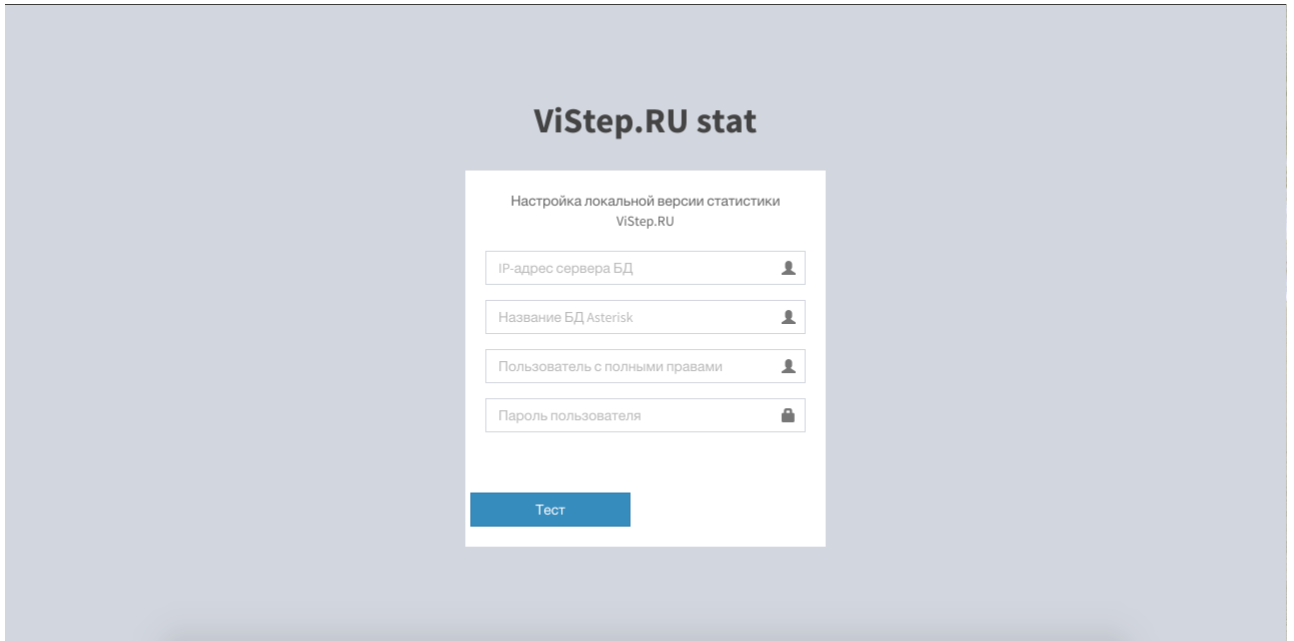
```
docker run --name=stat.vistep.ru_local_1.0.3 -p 8768:80 -v /root/records:/var/www/html/public/user/УНИКАЛЬНЫЙ_КАТАЛОГ_ПОЛЬЗОВАТЕЛЯ/some/path/to/files/ -d stat.vistep.ru_local_1.0.3
```

Ваш УНИКАЛЬНЫЙ_КАТАЛОГ_ПОЛЬЗОВАТЕЛЯ уточняйте у тех. поддержки ViStep.RU

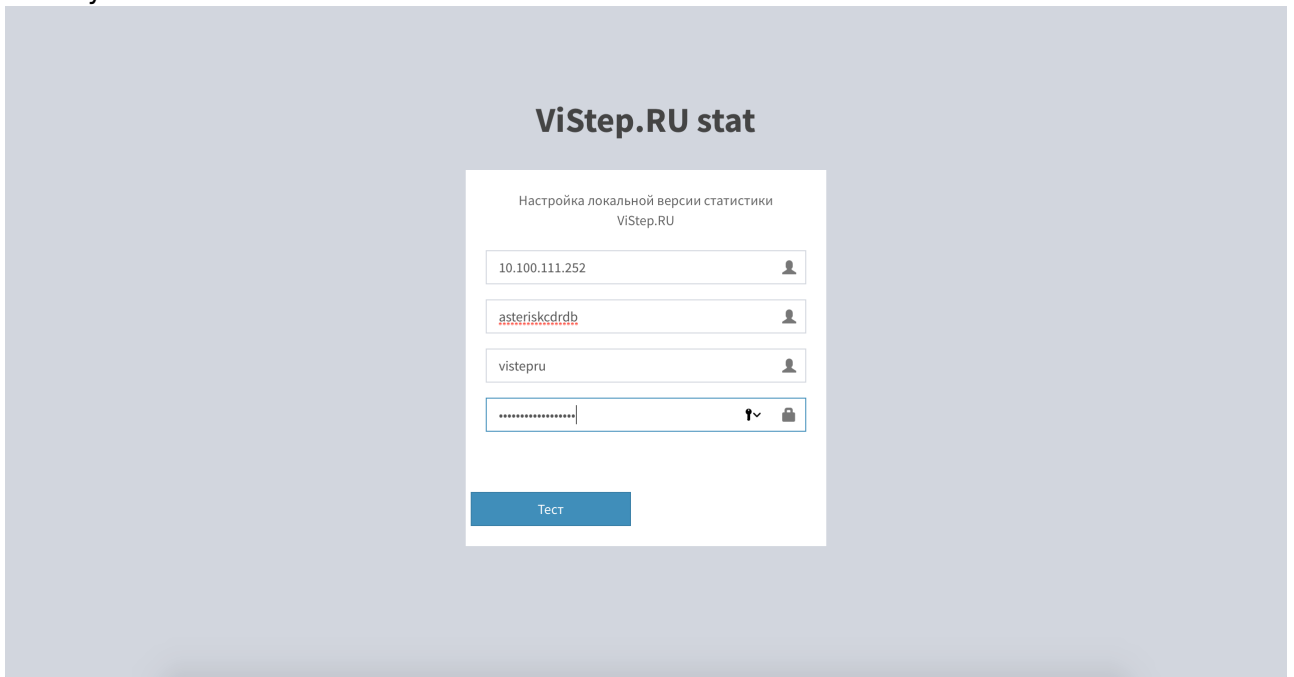
5. Далее откройте web-браузер и перейдите по ссылке

```
http://IP_ВАШЕЙ_АТС:8768/install.html
```

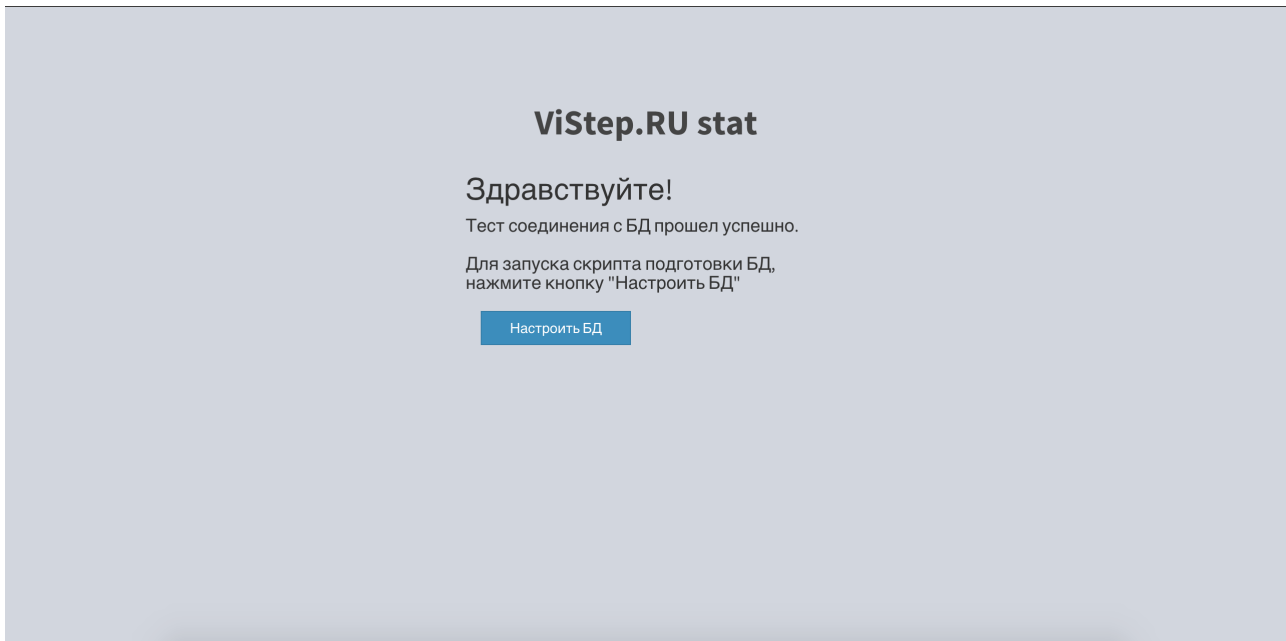
Вас встретит мастер настройки



6. Заполните все поля (на шаге номер 1 мы создали нужного пользователя) и нажмите кнопку **Тест**



7. При успешном прохождении теста на подключение к БД вы увидите

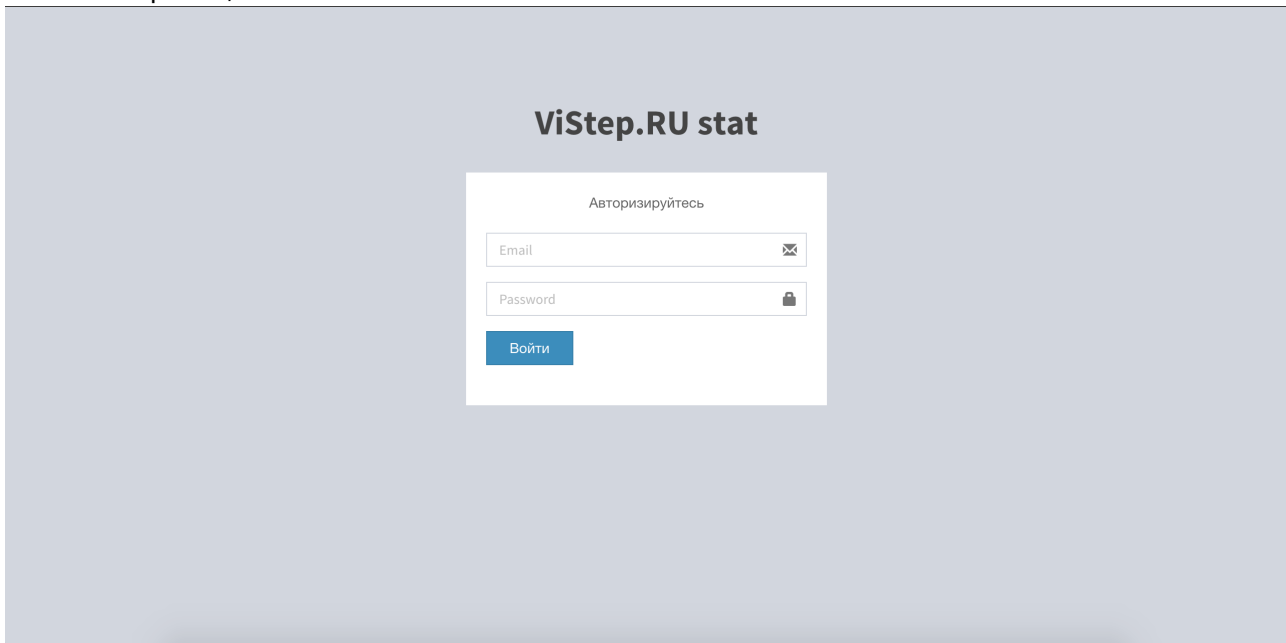


Если все так, жмите Настроить БД

8. Здесь мастер сообщит, что все прошло успешно и вы можете перейти по ссылке

`http://IP_ВАШЕЙ_АТС:8768`

т.е. к авторизации в системе



9. 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@vistep.ru или прямо из интерфейса, в меню Help Desk.

From:

<https://wiki.vistep.ru/> - Система статистики call центров на IP-АТС Asterisk (FreePBX)

Permanent link:

https://wiki.vistep.ru/doku.php?id=install_local_version_fpbx&rev=1550047751

Last update: **2019/02/13 15:49**

