

Установка ABillS на Debian

- [Установка ABillS на Debian](#)
- [Настройка сопутствующего ПО](#)
- [MySQL](#)
- [Web Server](#)
 - [Apache](#)
 - [Apache 2.4](#)
- [Perl modules](#)
- [FreeRadius](#)
 - [Установка](#)
 - [Проверка PERL](#)
 - [Настройка](#)
- [Дополнительно](#)
- [Постнастройка Abills](#)
 - [Первые шаги](#)
 - [Проверка](#)
- [Дополнительно](#)

Версия для которой писалась документация

```
Debian 8.6.0
Debian 9.5.0
Debian 10
Debian 11
Debian 12
```

Установка ABillS на Debian

При установке использовались параметры:

- SSH Server
- Standart system utilities

Переходим в режим `root`. (Используя пароль, который вводили при установке)

```
su
```

Скачиваем **ABillS**: На момент написания статьи актуальная версия **1.00.05** (Сходите [сюда](#) и проверьте).

```
cd /usr
wget https://netix.dl.sourceforge.net/project/abills/abills/1.00/abills-1.00.05.tgz
tar zxvf abills-1.00.05.tgz
cp /usr/abills/libexec/config.pl.default /usr/abills/libexec/config.pl
```

Изменяем конфигурационный файл `/usr/abills/libexec/config.pl`

```
#DB configuration
$conf{dbhost}='localhost';
$conf{dbname}='abills';
$conf{dbuser}='abills';
$conf{dbpasswd}='sqlpassword'; #
$conf{ADMIN_MAIL}='info@your.domain';
$conf{USERS_MAIL_DOMAIN}="your.domain";
#
$conf{secretkey}="test12345678901234567890";
```

Значение `$conf{secretkey}` рекомендуется заменить с "test12345678901234567890" на случайную длинную строку.

При изменении значения в `$conf{secretkey}` после установки, не забудьте изменить эту строку в `mods-enabled/sql` для **FreeRadius**.

Настройка сопутствующего ПО

Обновляем пакеты и систему:

```
apt-get update -yq && apt-get upgrade -yq  
apt-get install libperl-dev gcc make -yq
```

MySQL

Debian 8.xx

```
apt-get install -yq mariadb-server libmysqlclient-dev
```

Debian 9.xx

```
apt-get install -yq mariadb-server libmariadbclient-dev libmariadbclient-dev-compat
```

Создаём пользователя и базу.

```
mysql --default-character-set=utf8 -u root
```

В консоли mysql делаем :

```
GRANT ALL ON abills.* TO `abills`@localhost IDENTIFIED BY "sqlpassword"; -- sqlpassword $conf{dbpasswd}  
CREATE DATABASE abills DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;  
flush privileges;  
quit;
```

Загружаем таблицы в базу.

```
cd /usr/abills/db/  
mysql --default-character-set=utf8 -D abills < abills.sql
```

Если возникает **Invalid default value for 'registration'** выполняем

```
sed -i -e "1 s/^/SET SQL_MODE='NO_ENGINE_SUBSTITUTION,NO_AUTO_VALUE_ON_ZERO';/" /usr/abills/db/abills.sql  
mysql --default-character-set=utf8 -u root -D abills < /usr/abills/db/abills.sql
```

Web Server

Apache

Установка Apache

```
apt-get install -yq apache2
```

Создаём сертификаты.

```
/usr/abills/misc/certs_create.sh apache
```

Копируем готовый конфиг Apache в папку внешних конфигураций apache.

Apache 2.4

```
cp /usr/abills/misc/apache/abills_httpd.conf /etc/apache2/sites-enabled/
```

Включаем модули для апача:

```
a2enmod ssl
a2enmod rewrite
a2enmod suexec
a2enmod include
a2enmod cgid
a2enmod headers
```

Создаем файл лога ошибок Abills

```
mkdir /var/log/httpd
service apache2 restart
```

Perl modules

Для работы системы нужны модули.

```
cd /usr/abills/misc/ && perl perldeps.pl apt-get -batch
```

Улучшаем себе жизнь в будущем:

```
apt-get install -yq cpanminus
```

Если какие-то модули не установились, можно запустить

```
cpanm <>
```

Например у меня не установился **Digest::SHA1**, запускаю:

```
cpanm Digest::SHA1
```

FreeRadius

Установка

На все предупреждения **./configure** кроме **rlm_perl** и **rlm_sql_mysql** можно не обращать внимания.
На удаленном сервере установите **libmysqlclient** и **libmysqlclient-dev**

Проверка PERL

Проверка версии Perl

```
apt-get install libperl-dev -yq
find /usr/lib/ | grep libperl.so
/usr/lib/x86_64-linux-gnu/libperl.so.5.22
/usr/lib/x86_64-linux-gnu/libperl.so.5.22.1
```

Делаем симлинк для того, чтобы библиотека подгрузилась во время компиляции

```
ln -s /usr/lib/x86_64-linux-gnu/libperl.so.5.22 /usr/lib/x86_64-linux-gnu/libperl.so
```

```
apt-get install -yq gcc make libtalloc-dev
cd /tmp
wget ftp://ftp.freeradius.org/pub/freeradius/freeradius-server-3.0.17.tar.gz
tar zxvf freeradius-server-3.0.17.tar.gz
cd freeradius-server-3.0.17
./configure --prefix=/usr/local/freeradius --with-rlm-perl-lib-dir=/usr/lib/x86_64-linux-gnu/ \
--with-rlm-sql-mysql-lib-dir=/var/lib/mysql/ --with-dhcp=yes --with-openssl=no > /dev/null
make && make install
```

Настройка

```
cd /usr/abills
rm /usr/local/freeradius/etc/raddb/sites-enabled/*
cp misc/freeradius/v3/mods-enabled/perl /usr/local/freeradius/etc/raddb/mods-enabled/perl
cp misc/freeradius/v3/mods-enabled/sql /usr/local/freeradius/etc/raddb/mods-enabled/sql
cp misc/freeradius/v3/sites-enabled/abills_default /usr/local/freeradius/etc/raddb/sites-enabled/
abills_default
cp misc/freeradius/v3/sites-enabled/control-socket /usr/local/freeradius/etc/raddb/sites-enabled/control-
socket
cp misc/freeradius/v3/users /usr/local/freeradius/etc/raddb/users
ln -s /usr/local/freeradius/bin/* /usr/bin/
ln -s /usr/local/freeradius/sbin/* /usr/sbin/
```

Файл для автозапуска , - создаем и вставляем в него следующий текст

```
nano /etc/init.d/radiusd
```

```

#!/bin/sh
# Start/stop the FreeRADIUS daemon.

### BEGIN INIT INFO
# Provides:          radiusd
# Required-Start:    $remote_fs $network $syslog
# Should-Start:      $time mysql slapd postgresql samba krb5-kdc
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Radius Daemon
# Description:       Extensible, configurable radius daemon
### END INIT INFO

set -e

. /lib/lsb/init-functions

PROG="radiusd"
PROGRAM="/usr/sbin/radiusd"
PIDFILE="/usr/local/freeradius/var/run/radiusd/radiusd.pid"
DESCR="FreeRADIUS daemon"

test -f $PROGRAM || exit 0

# /var/run may be a tmpfs
if [ ! -d /var/run/radiusd ]; then
    mkdir -p /var/run/radiusd
    chown freerad:freerad /var/run/radiusd
fi

export PATH="${PATH:+$PATH:}/usr/sbin:/sbin"

ret=0

case "$1" in
    start)
        log_daemon_msg "Starting $DESCR" "$PROG"
        start-stop-daemon --start --quiet --pidfile $PIDFILE --exec $PROGRAM || ret=$?
        log_end_msg $ret
        exit $ret
        ;;
    stop)
        log_daemon_msg "Stopping $DESCR" "$PROG"
        if [ -f "$PIDFILE" ] ; then
            start-stop-daemon --stop --retry=TERM/30/KILL/5 --quiet --pidfile $PIDFILE || ret=$?
            log_end_msg $ret
        else
            log_action_cont_msg "$PIDFILE not found"
            log_end_msg 0
        fi
        ;;
    restart|force-reload)
        $0 stop
        $0 start
        ;;
    *)
        echo "Usage: $0 start|stop|restart|force-reload"
        exit 1
        ;;
esac

exit 0

```

```
chmod +x /etc/init.d/radiusd
update-rc.d radiusd defaults
update-rc.d radiusd enable
```

Меняем в конфиге радиуса **/usr/local/freeradius/etc/raddb/radiusd.conf**

```
nano /usr/local/freeradius/etc/raddb/radiusd.conf

prefix = /usr/local/freeradius

user = freerad
group = freerad
```

Благодаря модулю **sql**, можно хранить сервера доступа в базе данных (всё равно необходим перезапуск радиуса после добавления /изменения серверов, но не надо править **clients.conf**).

Очищаем список серверов доступа из **/usr/local/freeradius/etc/raddb/clients.conf** (чтоб не было дубликатов из-за записей в базе)

```
echo '' > /usr/local/freeradius/etc/raddb/clients.conf
```

Заполняем нужные нам параметры соединения с БД

```
nano /usr/local/freeradius/etc/raddb/mods-enabled/sql
```

```
sql {
    database = "mysql"
    driver = "rlm_sql_${database}"
    server = "localhost"
    #port = 3306
    login = "abills"
    password = "sqlpassword" # sqlpassword      $conf{dbpasswd}
    radius_db = "abills"

#
    '%secretkey%' , $conf{secretkey}
```

Создаём user и group. (если freeradius ранее не был установлен то без этого не стартует)

```
groupadd freerad
useradd -g freerad -s /bash/bash freerad
chown -R freerad:freerad /usr/local/freeradius/etc/raddb
```

Запуск radius в режиме отладки

```
/usr/sbin/radiusd -X
```

или

```
radiusd -X
```

Проверяем через другую консоль:

```
radtest test 123456 127.0.0.1:1812 0 secretpass 0 127.0.0.1
```

[Дополнительная настройка и тестирование RADIUS](#)

На всякий случай оставлю здесь: [Кажется настроил, но никто не может подключиться](#)

Если ошибок нет, запускаем

```
service radiusd start
```

Дополнительно

Открытие фаервола для RADIUS пакетов

```
iptables -A ufw-user-input -p udp --dport 1812 -j ACCEPT
iptables -A ufw-user-input -p udp --dport 1813 -j ACCEPT
```

Постнастройка Abills

Вносим периодические процессы

/etc/crontab

```
* /5 * * * * root /usr/abills/libexec/billd -all
1 0 * * * root /usr/abills/libexec/periodic daily
1 1 * * * root /usr/abills/libexec/periodic monthly
```

и даем доступ на ведение лога SQL ошибок

```
mkdir -p /usr/abills/var/log
touch /usr/abills/var/log/sql_errors && chmod 666 /usr/abills/var/log/sql_errors
```

Добавляем линки на **gzip** и **mysqldump** для создания бэкапов базы:

```
ln -s /bin/gzip /usr/bin/gzip
ln -s /usr/bin/mysqldump /usr/local/bin/mysqldump
```

Права на файлы и каталоги

```
chown -Rf www-data /usr/abills/cgi-bin /usr/abills/Abills/templates /usr/abills/backup /usr/abills/language
```

Веб интерфейс администратора: (IP-адрес можно узнать с помощью терминала – ip a)

<https://your.host:9443/admin/>

Логин администратора по умолчанию **abills** пароль **abills**

Веб интерфейс для пользователей:

<https://your.host:9443/>

Первые шаги

В интерфейсе администратора прежде всего надо сконфигурировать сервера доступа NAS (Network Access Server).

Переходим в меню

Настройка > Сервера доступа

Параметры

IP	IP адрес NAS сервера
Название	Название
Radius NAS-Identifier	Идентификатор сервера (можно не вписывать)
Описание	Описание сервера
Тип	Тип сервера. В зависимости от типа по разному обрабатываются запросы на авторизацию
Alive	Период отправки Alive пакетов
Disable	Отключить
:Manage:	Секция менеджмента NAS сервера
IP:PORT	IP адрес и порт для контроля соединения. Например, для отключения пользователя из веб-интерфейса
User	Пользователь для контроля
Password	Пароль
RADIUS Parameters	Дополнительные параметры которые передаются NAS серверу после успешной авторизации

После заведения сервера доступа добавьте ему пул адресов IP POOLs.

Первый IP	Первый адрес в пуле
Количество	Количество адресов

Одному серверу доступа может принадлежать несколько пулов адресов.

Создание тарифного плана

Меню

Настройка>Интернет>Тарифные планы

Регистрация пользователя

Клиенты>Логины>Добавить пользователя

Заведение сервиса Internet на пользователя.

Клиенты>Логины>Сервисы>Интернет

Проверка

Для проверки правильно ли настроен сервис нужно запустить утилиту radtest указав логин и пароль существующего пользователя.

Логин: test Пароль: 123456

```
/usr/local/freeradius/bin/radtest test 123456 127.0.0.1:1812 0 secretpass 0 127.0.0.1
```

Если всё правильно настроено, в журнале ошибок **Отчёт>Интернет+>Последнее подключение** должна появиться строка

```
2014-12-23 12:55:55 LOG_INFO: AUTH [test] NAS: 1 (xxx.xxx.xxx.xxx) GT: 0.03799
```

Если Вы увидите другие ошибки смотрите в [список ошибок](#).

Если журнал ошибок пуст, значит неправильно настроено взаимодействие с **RADIUS** сервером (еще раз пройдитесь по секции **FreeRadius**).

Дополнительно

- [Настройка NAS accel-ppp](#) (PPPoE/IPoE)
- [Настройка NAS Mikrotik](#)