

Создаём почтовый сервер

Полное руководство

Андрей Шетухин

В этой статье будет рассказано, как самостоятельно создать почтовый сервер на основе операционной системы FreeBSD, SMTP-сервера Postfix, POP3/IMAP4-сервера courier, систем фильтрации спама SpamAssassin и антивируса ClamAV. Для хранения учетных записей используется SQL СУБД PostgreSQL.

Установка компонентов

1. Устанавливаем СУБД PostgreSQL.
2. Устанавливаем систему аутентификации Cyrus-SASL.

```
cd /usr/ports/security/cyrus-sasl2
make install clean
```

Options for cyrus-sasl 2.1.22

```
[ ] BDB          Use Berkeley DB
[ ] MYSQL        Use MySQL
[X] PGSQL        Use PostgreSQL
[ ] SQLITE       Use SQLite
[ ] DEV_URANDOM  Use /dev/urandom
[ ] ALWAYSTRUE   Enable the alwaystrue password verifier
[ ] KEEP_DB_OPEN Keep handle to Berkeley DB open
[X] AUTHDAEMON   Enable use of authdaemon
[X] LOGIN        Enable LOGIN authentication
[X] PLAIN         Enable PLAIN authentication
[X] CRAM          Enable CRAM-MD5 authentication
[X] DIGEST        Enable DIGEST-MD5 authentication
[X] OTP           Enable OTP authentication
[X] NTLM          Enable NTLM authentication
```

[OK] Cancel

3. Устанавливаем Postfix.

```
cd /usr/ports/mail/postfix
make install clean
```

Отмечаем в таблице пункты, которые нам необходимы.

Options for postfix 2.5.1_2,1

```
[X] PCRE          Perl Compatible Regular Expressions
[X] SASL2         Cyrus SASLv2 (Simple Auth. and Sec. Layer)
[ ] DOVECOT       Dovecot SASL authentication method
[ ] SASLKR5       If your SASL req. Kerberos select this option
[ ] SASLKR5B5     If your SASL req. Kerberos5 select this option
[ ] SASLKRMIT     If your SASL req. MIT Kerberos5 select this option
[X] TLS           Enable SSL and TLS support
[ ] BDB           Berkeley DB (choose version with WITH_EDB_VER)
[ ] MYSQL         MySQL maps (choose version with WITH_MYSQL_VER)
[X] PGSQL         PostgreSQL maps (choose with DEFAULT_PGSQL_VER)
[ ] OPENLDAP      OpenLDAP maps (choose ver. with WITH_OPENLDAP_VER)
[ ] CDB           CDB maps lookups
[ ] NIS           NIS maps lookups
[X] VDA           VDA (Virtual Delivery Agent)
[X] TEST          SMTP/IMTP test server and generator
```

[OK] Cancel

Во время установки будет задан вопрос, хотим ли мы добавить пользователя postfix и группу maildrop.

```
Added group "postfix".
Added group "maildrop".
Added user "postfix".
You need user "postfix" added to group "mail".
Would you like me to add it [y]? y
```

Отвечаем утвердительно.

После установки отключаем Sendmail, указывая в файле /etc/rc.conf строки:

```
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
```

А в файле /etc/periodic.conf – строки:

```
daily_clean_hoststat_enable="NO"
daily_status_mail_rejects_enable="NO"
daily_status_include_submit_mailq="NO"
daily_submit_queuerun="NO"
```

После этого переходим к установке и настройке POP3/IMAP4 сервера courier.

```
cd /usr/ports/mail/courier-imap/
make install clean
```

Options for courier-imap 4.3.1,2

```
[X] FAM          Build in fam support for IDLE command
[X] DRAC        Build in DRAC support
[X] TRASHQUOTA  Include deleted mails in the quota
[ ] GDBM        Use gdbm db instead of system db
[ ] IPV6        Build with IPv6 support
[ ] AUTH_LDAP   LDAP support
[ ] AUTH_MYSQL  MySQL support
[X] AUTH_PGSQL  PostgreSQL support
[ ] AUTH_USERDB Userdb support
[ ] AUTH_VCHKFW Vpopmail/vchkpw support
```

[OK] Cancel

После установки указываем разрешение на запуск. Для Postfix:

```
postfix_enable="YES"
```

Для courier-imap:

```
courier_authdaemon_enable="YES"
courier_imap_imapd_enable="YES"
courier_imap_pop3d_enable="YES"
```

Если нам необходима поддержка защищенных соединений, включаем дополнительные сервисы:

```
courier_imap_pop3d_ssl_enable="YES"
courier_imap_imapd_ssl_enable="YES"
```

Также необходимо включить поддержку FAM, в файл /etc/inetd.conf прописываем строку:

```
sgi_fam/1-2      stream rpc/tcp wait root  ␣
                /usr/local/bin/fam  fam
```

Не забываем включить inetd, а также включаем portmapper, который необходим для работы FAM:

```
inetd_enable="YES"
rpcbind_enable="YES"
```

На всякий случай убеждаемся, что в файле /etc/rpc присутствует строка:

```
sgi_fam 391002
```

Финальный шаг установки ПО – создание учётной записи пользователя, под которой будет храниться почта:

```
pw groupadd mailsystem -g 900
pw useradd mailsystem -u 900 -g 900 -d /nonexistent ␣
-s /usr/sbin/nologin
id mailsystem

uid=900(mailsystem) gid=900(mailsystem) ␣
groups=900(mailsystem)
```

На этом сборка ПО заканчивается, и мы можем перейти к процедуре настройки.

Настройка ПО

Начнем настройку с создания базы данных. Как мы условились в предыдущей статье цикла, мы используем СУБД PostgreSQL. Итак, создаем БД. Для этого убеждаемся, что сервер PostgreSQL запущен и работает:

```
ps axw | grep postgres
```

```
1877 ?? Ss   0:02.48 /usr/local/bin/postgres -D /var/db/pgsql/data
1879 ?? Ss   0:04.29 postgres: writer process      (postgres)
1880 ?? Ss   0:02.69 postgres: wal writer process  (postgres)
1881 ?? Ss   0:01.70 postgres: autovacuum launcher process (postgres)
1882 ?? Ss   0:02.38 postgres: stats collector process  (postgres)
```

Заходим в консоль управления СУБД:

```
psql -U postgres template1
```

```
Welcome to psql 8.3.1, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=#
```

Создаем пользователя:

```
CREATE USER billing NOCREATEDB NOCREATEUSER PASSWORD ␣
'verysecret';
CREATE USER mailreader NOCREATEDB NOCREATEUSER PASSWORD ␣
'topsecret';
```

Создаем БД:

```
CREATE DATABASE billing WITH OWNER billing;
```

Подключаемся к БД:

```
\c billing
```

Создаем схему для почтовой части биллинга:

```
CREATE SCHEMA mail;
ALTER SCHEMA mail OWNER TO billing;
GRANT ALL ON SCHEMA mail TO billing;
GRANT USAGE ON SCHEMA mail TO mailreader;
```

Создаем необходимые таблицы:

```
CREATE TABLE mail.mailaliases (
    alias character varying(255) NOT NULL,
    rcpt character varying(255) NOT NULL,
    descr text,
    CONSTRAINT mailaliases_alias_check CHECK
        ((length((alias)::text) > 0)),
    CONSTRAINT mailaliases_rcpt_check CHECK
        ((length((rcpt)::text) > 0))
);

CREATE TABLE mail.mailtransport (
    maildomain character varying(255) NOT NULL,
    transport character varying(255) NOT NULL,
    descr text,
    CONSTRAINT mailtransport_maildomain_check CHECK
        ((length((maildomain)::text) > 0)),
    CONSTRAINT mailtransport_transport_check CHECK
        ((length((transport)::text) > 0))
);

CREATE TABLE mail.mailusers (
    username character varying(255) NOT NULL,
    passwd character varying(255) NOT NULL,
    user_id integer DEFAULT -1 NOT NULL,
    group_id integer DEFAULT -1 NOT NULL,
    userquota integer DEFAULT 0 NOT NULL,
    maildir character varying(255) NOT NULL,
    expires bigint DEFAULT (- 1)::bigint NOT NULL,
    enabled bigint DEFAULT (0)::bigint NOT NULL,
    descr text,
    CONSTRAINT mailusers_maildir_check CHECK
        ((length((maildir)::text) > 0)),
    CONSTRAINT mailusers_passwd_check CHECK
        ((length((passwd)::text) > 0)),
    CONSTRAINT mailusers_username_check CHECK
        ((length((username)::text) > 0))
);

ALTER TABLE mail.mailaliases OWNER TO billing;
ALTER TABLE mail.mailtransport OWNER TO billing;
ALTER TABLE mail.mailusers OWNER TO billing;

GRANT SELECT ON TABLE mail.mailaliases TO mailreader;
GRANT SELECT ON TABLE mail.mailtransport TO mailreader;
GRANT SELECT ON TABLE mail.mailusers TO mailreader;
```

Пара комментариев по проделанным действиям. Мы создали двух пользователей с именами billing и mailreader. Пользователю billing даны полные права на доступ ко всем таблицам, а пользователю mailreader – только на чтение. Таким образом, изменять данные мы будем от пользователя billing, а работать с БД почтовая система будет от пользователя mailreader.

Настраиваем на работу с созданной БД систему аутентификации courier. Для этого редактируем файл `/usr/local/etc/authlib/authdaemonrc`, указывая в нем тип базы данных, содержащих пользователей:

```
authmodulelist="authpgsql"
```

Ничего более в глобальной конфигурации демона аутентификации изменять не требуется.

После этого прописываем имя базы данных, пользователя и пароль для доступа к БД, хранящей учетные записи пользователей:

```
PGSQL_SERVER      127.0.0.1
PGSQL_USERNAME    mailreader
PGSQL_PASSWORD    topsecret

PGSQL_PORT        5432
PGSQL_DATABASE    billing

PGSQL_USER_TABLE  mail.mailusers
PGSQL_CLEAR_PFIELD passwd
PGSQL_UID_FIELD   user_id
PGSQL_GID_FIELD   group_id
PGSQL_LOGIN_FIELD username
PGSQL_HOME_FIELD  maildir
PGSQL_MAILDIR_FIELD maildir
PGSQL_WHERE_CLAUSE enabled=1
```

Внимание: если вы укажете метод хранения паролей, как `PGSQL_CRYPT_PWFIELD`, вы не сможете использовать тип аутентификации `CRAM-MD5`. Поэтому вы либо храните пароли в хешированном (криптованном) виде, и ваш сервер не поддерживает `CRAM-MD5`, либо пароли находятся внутри БД, как `plaintext`, но доступен более защищенный метод аутентификации.

Создаем тестовый почтовый домен и тестового пользователя и запускаем систему в работу. В консоли PostgreSQL набираем команды:

```
INSERT INTO mail.mailtransport (maildomain, transport, ↓
    descr)
VALUES ('example.com', 'virtual', 'virtual transport');

INSERT INTO mail.mailusers (username, passwd, user_id, ↓
    group_id, userquota, maildir, expires, enabled, ↓
    descr)
VALUES ('test@example.com', 'mailpassword', '900', ↓
    '900', '1024000', ↓
    '/var/spool/mail/example.com/test/', '0', '1', '');

```

Запускаем сервисы:

```
/usr/local/etc/rc.d/courier-authdaemon start
/usr/local/etc/rc.d/courier-imap-pop3d start
/usr/local/etc/rc.d/courier-imap-imapd start
```

Проверяем, что все работает:

```
ps axw | grep courier
```

```

89902 p0 I      0:00.00 /usr/local/sbin/courierlogger -facility=mail
                        -pid=/var/run/authdaemond/pid
                        -start /usr/local/libexec/courier-authli
89903 p0 I      0:00.01 /usr/local/libexec/courier-authlib/authdaemond
89904 p0 I      0:00.00 /usr/local/libexec/courier-authlib/authdaemond
89905 p0 I      0:00.00 /usr/local/libexec/courier-authlib/authdaemond
89906 p0 I      0:00.00 /usr/local/libexec/courier-authlib/authdaemond
89907 p0 I      0:00.00 /usr/local/libexec/courier-authlib/authdaemond
89908 p0 I      0:00.00 /usr/local/libexec/courier-authlib/authdaemond
89933 p0 I      0:00.00 /usr/local/sbin/courierlogger
                        -pid=/var/run/pop3d.pid -start -name=pop3d
                        /usr/local/libexec/courier-imap/couriertcp
89934 p0 I      0:00.00 /usr/local/libexec/courier-imap/couriertcpd
                        -address=0 -maxprocs=40 -maxperip=4 -nodnslookup
                        -noidentlookup 110 /us
89945 p0 I      0:00.00 /usr/local/sbin/courierlogger
                        -pid=/var/run/imapd.pid -start -name=imapd
                        /usr/local/libexec/courier-imap/couriertcp
89946 p0 I      0:00.00 /usr/local/libexec/courier-imap/couriertcpd
                        -address=0 -maxprocs=40 -maxperip=4 -nodnslookup
                        -noidentlookup 143 /us
89956 p0 R+    0:00.00 grep courier

```

Проверяем, что все необходимые порты прослушиваются:

```
sockstat -4 | grep couriertcp
```

```
root      couriertcp 89946 3  tcp4  *:143      *:*
root      couriertcp 89934 3  tcp4  *:110      *:*
```


Все правильно: прослушиваются порты 110 и 143.

Теперь необходимо настроить Postfix так, чтобы он доставлял почту в каталоги, указанные в нашей SQL СУБД. Для этого нам следует отредактировать файлы /usr/local/etc/postfix/master.cf и /usr/local/etc/postfix/main.cf. Собственно, правка сводится к указанию файлов конфигурации доступа к СУБД и прописыванию метода аутентификации. В файле /usr/local/etc/postfix/main.cf указываем:

```
# Имя хоста
myhostname = mail.example.com
# Имя домена
mydomain = example.com

# Порядок поиска пользователей
local_recipient_maps = unix:passwd.byname ↵
virtual_mailbox_maps $virtual_mailbox_maps

# Список сетей, с которых мы можем принимать почту
# без аутентификации
mynetworks = 127.0.0.0/8
# Ограничения на транзит почты (разрешаем локальным
# и аутентифицированным, запрещаем всем остальным)
smtpd_recipient_restrictions = permit_mynetworks, ↵
permit_sasl_authenticated, reject_unauth_destination

# Список доменов, которые обрабатываются системой
mydestination = $myhostname, localhost.$mydomain, localhost, ↵
pgsql:/usr/local/etc/postfix/sql-mydestination.cf

# Файлы конфигурации SQL
virtual_mailbox_base = /
virtual_mailbox_maps = pgsql:/usr/local/etc/postfix/ ↵
sql-virtual-maps.cf
virtual_maps = pgsql:/usr/local/etc/postfix/ ↵
sql-virtual.cf
virtual_minimum_uid = 500
virtual_uid_maps = pgsql:/usr/local/etc/postfix/ ↵
sql-virtual-uid.cf
virtual_gid_maps = pgsql:/usr/local/etc/postfix/ ↵
sql-virtual-gid.cf
transport_maps = pgsql:/usr/local/etc/postfix/ ↵
sql-transport.cf

# Квота
virtual_mailbox_limit = 10240000
virtual_mailbox_limit_inbox = yes
virtual_mailbox_limit_maps = pgsql:/usr/local/etc/ ↵
postfix/sql-mailboxsize.cf
virtual_mailbox_limit_override = yes
virtual_maildir_extended = yes
virtual_create_maildirsize = yes

broken_sasl_auth_clients = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_auth_enable = yes
```

Для поддержки SASL-аутентификации создаем файл /usr/local/lib/sasl2/smtpd.conf со следующим содержимым:

```
pwcheck_method: auxprop
auxprop_plugin: sql
sql_engine: pgsql
sql_user: mailreader
sql_passwd: topsecret
sql_hostnames: 127.0.0.1
sql_database: billing
sql_select: select passwd from mail.mailusers ↵
where username = '%u@%r'
sql_verbose: yes
```

Проверяем работу

Попробуем отправить почту локальному пользователю:

```
telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^J'.
220 mail.example.com ESMTP Postfix
HELO stellar
250 mail.example.com
MAIL FROM:<test@mail.com>
250 2.1.0 Ok
RCPT TO:<test@example.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
FROM:<test@mail.com>
TO:<test@example.com>
SUBJECT: test

passed?
.
250 2.0.0 Ok: queued as D0D441E3049
QUIT
221 2.0.0 Bye
```

Проверяем доставку по лог-файлам:

```
Apr 2 17:18:56 homer postfix/smtpd[22459]:
connect from localhost[127.0.0.1]
Apr 2 17:19:08 homer postfix/smtpd[22459]:
D0D441E3049: client=localhost[127.0.0.1]
Apr 2 17:19:16 homer postfix/cleanup[22469]:
D0D441E3049: message-id=<20080402131908.D0D441E3049@mail.example.com>
Apr 2 17:19:16 homer postfix/qmgr[22457]:
D0D441E3049: from=<test@mail.com>, size=353, nrcpt=1 (queue active)
Apr 2 17:19:16 homer postfix/virtual[22508]:
D0D441E3049: to=<test@example.com>, relay=virtual, delay=13,
delays=13/0.01/0/0.05, dsn=2.0.0, status=sent (delivered to maildir)
Apr 2 17:19:16 homer postfix/qmgr[22457]: D0D441E3049: removed
Apr 2 17:19:19 homer postfix/smtpd[22459]:
disconnect from localhost[127.0.0.1]
```

Проверяем, можно ли забрать почту по протоколу POP3:

```
telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^J'.
+OK Hello there.
USER test@example.com
+OK Password required.
PASS mailpassword
+OK logged in.
LIST
+OK POP3 clients that break here, they violate STD53.
1 455
.
QUIT
+OK Bye-bye.
```

В этой части статьи мы разобрали, как провести основную настройку почтового сервера. В следующей части мы коснемся вопросов конфигурации веб-интерфейса для доступа к почтовым ящикам, настройки систем фильтрации спама и вирусов. 

