Ejabberd - настройка собственного jabber-сервера

Опубликовано muff в Вс, 2014-03-30 04:29

Понадобилось как-то организовать защищенный обмен сообщениями между пользователями небольшого офиса и работающими удаленно сотрудниками. В качестве **Jabber**-сервера на этот раз решено использовать **ejabberd** (статья по настройке **Jabber**-сервера Openfire [1]).

Стоит отметить тот факт, что **ejabberd** является лидером среди **Jabber**-серверов. Его используют **Facebook**, **Vkontakte**, **Yandex**, **Google**, etc.

Ключевыми особеностями **ejabberd** являются:

- веб-консоль администрирования и инструменты командной строки обеспечивают простоту настройки
- поддерживается шифрование соединений для обеспечения безопасной связи
- множество встроенных сервисов, в том числе **Jabber User Directory** и **Multi-User Chat**
- стандартный **Jabber**-транспорт позволяет обеспечивать связь с другими сетями мгновенного обмена сообщениями
- автоматический сбор статистики для отчетности и анализа системы
- виртуальный хостинг, что позволяет **ejabberd** поддерживать несколько виртуальных доменов в одном екземпляре **ejabberd**
- общие списки, так что пользователи могут иметь предопределенные контакты в списках контактов
- поддерживаются внешние хранилища
- большое количество поддерживаемых протоколов
- многоязычный интерфейс веб-администрирования
- полная поддержка **IPv6**

ejabberd написан на языке **Erlang** и конфигурационный файл в том числе, однако знаний языка для правки конфигурационного файла не нужно - он интуитивно понятен.

Перед установкой и настройкой **ejabberd**, внесем необходимые изменения в **DNS**, чтобы потом на это не отвлекаться. Для полноценной работы **jabber**-сервера, необходимо внести **A-** и **SRV-**записи. Если используется **DNS**-сервер **BIND9** [2], то записи будут выглядеть примерно так:

```
_xmpp-server._tcp.jabber.example.org. IN SRV 0 0 5269     jabber.example.org
._jabber._tcp.jabber.example.org.      IN SRV 0 0 5269     jabber.example.org
._xmpp-client._tcp.jabber.example.org. IN SRV 0 0 5222     jabber.example.org
.conference.jabber       IN      A       10.227.206.69jabber                IN      A       10.227.206.69
```

Ну и соответственно, нужно проследить, чтобы изменения **DNS** вступили в силу...
Итак, вернемся непосредственно к **ejabberd**. Установку **ejabberd** выполним из системы портов:

```
# cd /usr/ports/net-im/ejabberd && make install && rehash
```

Поскольку всю информацию будем хранить в **MySQL** [3], то среди опций установки необходимо отметить **ODBC** (**Open DataBase Connectivity**):

```
ejabberd-2.1.13


[x] DOCS  Build and/or install documentation[x] ODBC  ODBC backend[ ] PAM   PAM support
```

Во время установки **ejabberd** будет установлена зависимость - язык **Erlang**, среди опций установки также необходимо выбрать поддержку **ODBC**.

По завершению установки, отобразились короткие рекомендации относительно дальнейшей настройки.

```
Congratulations, you have installed ejabberd!  o To enable ejabberd you need to add
following string to your /etc/rc.conf:        ejabberd_enable="YES"  o Config sample
s are located in /usr/local/etc/ejabberd - please, create    working ones before run
ning ejabberd.  o The home directory is /var/spool/ejabberd  o PLEASE NOTE! "ejabber
d" user UID and GID have changed from version 2.0.3    of this port! Users are advis
ed to check all configuration files and data    in /var/spool/ejabberd to ensure eve
rything has the correct owner/group.
```

Ознакомимся с рекомендациями, начиная "с хвоста"...

Согласно месседжа, изменились **UID** и **GID** демона **ejabberd**. Однако, поскольку мы не обновляем **ejabberd**, а выполняем новую установку, то нас это предупреждение не касается. Кстати, в текущей версии 2.1.13 используется **UID** и **GID** с идентификаторами 543.

Домашний каталог - /var/spool/ejabberd... Что ж, в случае чего - заглянем.

А вот информация о том, что есть примеры конфигурационных файлов порадовала... Создадим конфигурационные файлы копированием примеров, установим владельцем файлов пользователя **ejabberd**:

```
# cd /usr/local/etc/ejabberd
# cp ejabberd.cfg.example ejabberd.cfg
# cp ejabberdctl.cfg.example ejabberdctl.cfg
# chown ejabberd:ejabberd *.cfg
```

В принципе, можно уже приступать к редактированию конфигурационного файла, но чтобы не возвращаться к нему повторно, выполним несколько предварительных действий.

Создадим БД в **MySQL** и пользователя:

```
mysql> create database ejabberd;
Query OK, 1 row affected (0,01 sec)

mysql> grant all on ejabberd.* to ejabberd@localhost identified by 'password_here';
Query OK, 0 rows affected (0,00 sec)
```

Создаем структуру таблиц, воспользовавшись sql-дампом, который идет в комплекте с установочными файлами ejabberd (там же можно обнаружить варианты для альтернативных поддерживаемых БД):

```
# cd /usr/ports/net-im/ejabberd/work/ejabberd-2.1.13/src/odbc
# mysql -u ejabberd -p ejabberd <  mysql.sql
```

В моем случае (mysql55-server-5.5.36) пришлось дамп немного подредактировать, поскольку попытка заливки дампа завершалась ошибкой:

```
# mysql -u ejabberd -p ejabberd <  mysql.sql
Enter password:
ERROR 1193 (HY000) at line 21: Unknown system variable 'table_type'
```

Как оказалось, проблема была в устаревшем синтаксисе дампа **MySQL**. Проблема в строке 21 - **SET table_type=InnoDB;**. Данная переменная удалена в **MySQL 5.5.3**. Вместо нее необходимо использовать **storage_engine** либо же **default_storage_engine**. В моем случае переменная **default_storage_engine** была установлена в значение **InnoDB**.

```
mysql> show variables like 'default_storage_engine';
+------------------------+--------+| Variable_name          | Value  |+-------------
----------+--------+| default_storage_engine | InnoDB
 |+------------------------+--------+1 row in set (0,00 sec)
```

Соответственно, для решения проблемы достаточно было удалить либо закомментировать содержимое строки 21. После этого дамп БД корректно создал структуру таблиц:

```
mysql> use ejabberd;
Reading table information for completion of table and column namesYou can turn off t
his feature to get a quicker startup with -ADatabase changedmysql> show tables;
+------------------------+| Tables_in_ejabberd      |+------------------------+| i
rc_custom              || last                   || motd                   || muc_
registered          || muc_room               || privacy_default_list    || privacy
_list            || privacy_list_data       || private_storage         || pubsub_ite
m             || pubsub_node             || pubsub_node_option      || pubsub_node_o
wner       || pubsub_state            || pubsub_subscription_opt || roster_version
        || rostergroups            || rosterusers             || spool
     || sr_group               || sr_user                || users
  || vcard                  || vcard_search           || vcard_xupdate          |
+------------------------+25 rows in set (0,00 sec)
```

Продвигаемся дальше... Поскольку планируется работа с поддержкой SSL, займемся сертификатом. Создадим каталог для сертификатов (если такового нету), сгенеруем сертификат и выставим права доступа на файл сертификата:

```
# mkdir /etc/ssl/certs && cd /etc/ssl/certs
# openssl req -new -x509 -nodes -newkey rsa:1024 -days 3650 -keyout privkey.key -out
ejabberd.pem \
-subj "/C=UA/ST=/L=Kiev/O=Company
LLC/OU=/CN=jabber.example.org/emailAddress=email [at] domain [dot] com"
# cat privkey.key >> ejabberd.pem
# rm privkey.key
# chown ejabberd:ejabberd ejabberd.pem
# chmod 600 ejabberd.pem
```

Здесь:

- **C** - Country Name (2 letter code)
- **ST** - State or Province Name (full name)
- **L** - Locality Name (eg, city)
- **O** - Organization Name (eg, company)
- **OU** - Organizational Unit Name (eg, section)
- **CN** - Common Name (eg, YOUR name)
- **emailAddress** - Email Address

Кажется пора заняться конфигурационным файлом **ejabberd.cfg**. Наведу все полотно конфигурационного файла - с комментариями все же удобнее. Его синтаксис, такой же как и языка **Erlang**. Основы синтаксиса детально описаны в начале конфигурационного файла.

```
%%%
%%%              ejabberd configuration file
%%%
%%%'

%%% The parameters used in this configuration file are explained in more detail
%%% in the ejabberd Installation and Operation Guide.
%%% Please consult the Guide in case of doubts, it is included with
%%% your copy of ejabberd, and is also available online at
%%% http://www.process-one.net/en/ejabberd/docs/

%%% This configuration file contains Erlang terms.
%%% In case you want to understand the syntax, here are the concepts:
%%%
%%%  - The character to comment a line is %
%%%
%%%  - Each term ends in a dot, for example:
%%%      override_global.
%%%
%%%  - A tuple has a fixed definition, its elements are
%%%    enclosed in {}, and separated with commas:
%%%      {loglevel, 4}.
%%%
%%%  - A list can have as many elements as you want,
%%%    and is enclosed in [], for example:
%%%      [http_poll, web_admin, tls]
%%%
%%%  - A keyword of ejabberd is a word in lowercase.
%%%    Strings are enclosed in "" and can contain spaces, dots, ...
%%%      {language, "en"}.
%%%      {ldap_rootdn, "dc=example,dc=com"}.
%%%
%%%  - This term includes a tuple, a keyword, a list, and two strings:
%%%      {hosts, ["jabber.example.net", "im.example.com"]}.
%%%

%%%.   =======================
%%%'   OVERRIDE STORED OPTIONS

%%
%% Override the old values stored in the database.
%%

%%
%% Override global options (shared by all ejabberd nodes in a cluster).
%%
%%override_global.

%%
%% Override local options (specific for this particular ejabberd node).
%%
%%override_local.

%%
%% Remove the Access Control Lists before new ones are added.
%%
```

```
%%override_acls.

%%%.   ==========
%%%'   DEBUGGING

%%
%% loglevel: Verbosity of log files generated by ejabberd.
%% 0: No ejabberd log at all (not recommended)
%% 1: Critical
%% 2: Error
%% 3: Warning
%% 4: Info
%% 5: Debug
%%
{loglevel, 4}.

%%
%% watchdog_admins: Only useful for developers: if an ejabberd process
%% consumes a lot of memory, send live notifications to these XMPP
%% accounts.
%%
%%{watchdog_admins, ["bob [at] example [dot] com"]}.

%%%.   =================
%%%'   SERVED HOSTNAMES

%%
%% hosts: Domains served by ejabberd.
%% You can define one or several, for example:
%% {hosts, ["example.net", "example.com", "example.org"]}.
%%
{hosts, ["jabber.example.org"]}.

%%
%% route_subdomains: Delegate subdomains to other XMPP servers.
%% For example, if this ejabberd serves example.org and you want
%% to allow communication with an XMPP server called im.example.org.
%%
%%{route_subdomains, s2s}.

%%%.   ================
%%%'   LISTENING PORTS

%%
%% listen: The ports ejabberd will listen on, which service each is handled
%% by and what options to start it with.
%%
{listen,
 [

  {5222, ejabberd_c2s, [

                  %%
                  %% If TLS is compiled in and you installed a SSL
                  %% certificate, specify the full path to the
                  %% file and uncomment this line:
                  %%
                  {certfile, "/etc/ssl/certs/ejabberd.pem"}, starttls,
```

```
                {access, c2s},
                {shaper, c2s_shaper},
                {max_stanza_size, 65536}
               ]},

  %%
  %% To enable the old SSL connection method on port 5223:
  %%

  %%{5223, ejabberd_c2s, [
  %%               {access, c2s},
  %%               {shaper, c2s_shaper},
  %%               {certfile, "/path/to/ssl.pem"}, tls,
  %%               {max_stanza_size, 65536}
  %%              ]},

  {5269, ejabberd_s2s_in, [
                  {shaper, s2s_shaper},
                  {max_stanza_size, 131072}
                 ]},

  %%
  %% ejabberd_service: Interact with external components (transports, ...)
  %%
  %%{8888, ejabberd_service, [
  %%                  {access, all},
  %%                  {shaper_rule, fast},
  %%                  {ip, {127, 0, 0, 1}},
  %%                  {hosts, ["icq.example.org", "sms.example.org"],
  %%                   [{password, "secret"}]
  %%                  }
  %%                 ]},

  %%
  %% ejabberd_stun: Handles STUN Binding requests
  %%
  %%{{3478, udp}, ejabberd_stun, []},

  {5280, ejabberd_http, [
                  %%{request_handlers,
                  %% [
                  %%  {["pub", "archive"], mod_http_fileserver}
                  %% ]},
                  captcha,
                  http_bind,
                  http_poll,
                  %%register,
                  web_admin
                 ]}

 ].

%%
%% s2s_use_starttls: Enable STARTTLS + Dialback for S2S connections.
%% Allowed values are: false optional required required_trusted
%% You must specify a certificate file.
%%
%%{s2s_use_starttls, optional}.
```

```
%%
%% s2s_certfile: Specify a certificate file.
%%
%%{s2s_certfile, "/path/to/ssl.pem"}.

{s2s_certfile, "/etc/ssl/certs/ejabberd.pem"}.

%%
%% domain_certfile: Specify a different certificate for each served hostname.
%%
%%{domain_certfile, "example.org", "/path/to/example_org.pem"}.
%%{domain_certfile, "example.com", "/path/to/example_com.pem"}.

%%
%% S2S whitelist or blacklist
%%
%% Default s2s policy for undefined hosts.
%%
%%{s2s_default_policy, allow}.

%%
%% Allow or deny communication with specific servers.
%%
%%{{s2s_host, "goodhost.org"}, allow}.
%%{{s2s_host, "badhost.org"}, deny}.

%%
%% Outgoing S2S options
%%
%% Preferred address families (which to try first) and connect timeout
%% in milliseconds.
%%
%%{outgoing_s2s_options, [ipv4, ipv6], 10000}.

%%%.   ===============
%%%'   AUTHENTICATION

%%
%% auth_method: Method used to authenticate the users.
%% The default method is the internal.
%% If you want to use a different method,
%% comment this line and enable the correct ones.
%%
%% {auth_method, internal}.
%%
%% Store the plain passwords or hashed for SCRAM:
%%{auth_password_format, plain}.
%%{auth_password_format, scram}.
%%
%% Define the FQDN if ejabberd doesn't detect it:
%%{fqdn, "server3.example.com"}.

%%
%% Authentication using external script
%% Make sure the script is executable by ejabberd.
%%
%%{auth_method, external}.
%%{extauth_program, "/path/to/authentication/script"}.
```

```
%%
%% Authentication using ODBC
%% Remember to setup a database in the next section.
%%
%%{auth_method, odbc}.

{auth_method, odbc}.

%%
%% Authentication using PAM
%%
%%{auth_method, pam}.
%%{pam_service, "pamservicename"}.

%%
%% Authentication using LDAP
%%
%%{auth_method, ldap}.
%%
%% List of LDAP servers:
%%{ldap_servers, ["localhost"]}.
%%
%% Encryption of connection to LDAP servers:
%%{ldap_encrypt, none}.
%%{ldap_encrypt, tls}.
%%
%% Port to connect to on LDAP servers:
%%{ldap_port, 389}.
%%{ldap_port, 636}.
%%
%% LDAP manager:
%%{ldap_rootdn, "dc=example,dc=com"}.
%%
%% Password of LDAP manager:
%%{ldap_password, "******"}.
%%
%% Search base of LDAP directory:
%%{ldap_base, "dc=example,dc=com"}.
%%
%% LDAP attribute that holds user ID:
%%{ldap_uids, [{"mail", "%u [at] mail [dot] example [dot] org"}]}.
%%
%% LDAP filter:
%%{ldap_filter, "(objectClass=shadowAccount)"}.

%%
%% Anonymous login support:
%%   auth_method: anonymous
%%   anonymous_protocol: sasl_anon | login_anon | both
%%   allow_multiple_connections: true | false
%%
%%{host_config, "public.example.org", [{auth_method, anonymous},
%%                                      {allow_multiple_connections, false},
%%                                      {anonymous_protocol, sasl_anon}]}.
%%
%% To use both anonymous and internal authentication:
%%
%%{host_config, "public.example.org", [{auth_method, [internal, anonymous]}]}.
```

```
%%%.   ===============
%%%'   DATABASE SETUP

%% ejabberd by default uses the internal Mnesia database,
%% so you do not necessarily need this section.
%% This section provides configuration examples in case
%% you want to use other database backends.
%% Please consult the ejabberd Guide for details on database creation.

%%
%% MySQL server:
%%
%%{odbc_server, {mysql, "server", "database", "username", "password"}}.
%%
%% If you want to specify the port:
%%{odbc_server, {mysql, "server", 1234, "database", "username", "password"}}.

{odbc_server, {mysql, "localhost", "ejabberd", "ejabberd", "password"}}.

%%
%% PostgreSQL server:
%%
%%{odbc_server, {pgsql, "server", "database", "username", "password"}}.
%%
%% If you want to specify the port:
%%{odbc_server, {pgsql, "server", 1234, "database", "username", "password"}}.
%%
%% If you use PostgreSQL, have a large database, and need a
%% faster but inexact replacement for "select count(*) from users"
%%
%%{pgsql_users_number_estimate, true}.

%%
%% ODBC compatible or MSSQL server:
%%
%%{odbc_server, "DSN=ejabberd;UID=ejabberd;PWD=ejabberd"}.

%%
%% Number of connections to open to the database for each virtual host
%%
%%{odbc_pool_size, 10}.

%%
%% Interval to make a dummy SQL request to keep the connections to the
%% database alive. Specify in seconds: for example 28800 means 8 hours
%%
%%{odbc_keepalive_interval, undefined}.

%%%.   ===============
%%%'   TRAFFIC SHAPERS

%%
%% The "normal" shaper limits traffic speed to 1000 B/s
%%
{shaper, normal, {maxrate, 1000}}.

%%
%% The "fast" shaper limits traffic speed to 50000 B/s
%%
```

```
{shaper, fast, {maxrate, 50000}}.

%%
%% This option specifies the maximum number of elements in the queue
%% of the FSM. Refer to the documentation for details.
%%
{max_fsm_queue, 1000}.

%%%.  =====================
%%%'  ACCESS CONTROL LISTS

%%
%% The 'admin' ACL grants administrative privileges to XMPP accounts.
%% You can put here as many accounts as you want.
%%
%%{acl, admin, {user, "aleksey", "localhost"}}.
%%{acl, admin, {user, "ermine", "example.org"}}.

{acl, admin, {user, "muff", "jabber.example.org"}}.

%%
%% Blocked users
%%
%%{acl, blocked, {user, "baduser", "example.org"}}.
%%{acl, blocked, {user, "test"}}.

%%
%% Local users: don't modify this line.
%%
{acl, local, {user_regexp, ""}}.

%%
%% More examples of ACLs
%%
%%{acl, jabberorg, {server, "jabber.org"}}.
%%{acl, aleksey, {user, "aleksey", "jabber.ru"}}.
%%{acl, test, {user_regexp, "^test"}}.
%%{acl, test, {user_glob, "test*"}}.

%%
%% Define specific ACLs in a virtual host.
%%
%%{host_config, "localhost",
%% [
%%  {acl, admin, {user, "bob-local", "localhost"}}
%% ]
%%}.

%%%.  =============
%%%'  ACCESS RULES

%% Maximum number of simultaneous sessions allowed for a single user:
{access, max_user_sessions, [{10, all}]}.

%% Maximum number of offline messages that users can have:
{access, max_user_offline_messages, [{5000, admin}, {500, all}]}.

%% This rule allows access only for local users:
{access, local, [{allow, local}]}.
```

```
%% Only non-blocked users can use c2s connections:
{access, c2s, [{deny, blocked},
               {allow, all}]}.

%% For C2S connections, all users except admins use the "normal" shaper
{access, c2s_shaper, [{none, admin},
                      {normal, all}]}.

%% All S2S connections use the "fast" shaper
{access, s2s_shaper, [{fast, all}]}.

%% Only admins can send announcement messages:
{access, announce, [{allow, admin}]}.

%% Only admins can use the configuration interface:
{access, configure, [{allow, admin}]}.

%% Admins of this server are also admins of the MUC service:
{access, muc_admin, [{allow, admin}]}.

%% Only accounts of the local ejabberd server can create rooms:
{access, muc_create, [{allow, local}]}.

%% All users are allowed to use the MUC service:
{access, muc, [{allow, all}]}.

%% Only accounts on the local ejabberd server can create Pubsub nodes:
{access, pubsub_createnode, [{allow, local}]}.

%% In-band registration allows registration of any possible username.
%% To disable in-band registration, replace 'allow' with 'deny'.
{access, register, [{allow, all}]}.

%% By default the frequency of account registrations from the same IP
%% is limited to 1 account every 10 minutes. To disable, specify: infinity
%%{registration_timeout, 600}.

%%
%% Define specific Access Rules in a virtual host.
%%
%%{host_config, "localhost",
%% [
%%  {access, c2s, [{allow, admin}, {deny, all}]},
%%  {access, register, [{deny, all}]}
%% ]
%%}.

%%%.  =================
%%%'  DEFAULT LANGUAGE

%%
%% language: Default language used for server messages.
%%
{language, "en"}.

%%
%% Set a different default language in a virtual host.
%%
```

```
%%{host_config, "localhost",
%% [{language, "ru"}]
%%}.

%%%.  ========
%%%'  CAPTCHA

%%
%% Full path to a script that generates the image.
%%
%%{captcha_cmd, "/lib/ejabberd/priv/bin/captcha.sh"}.

%%
%% Host for the URL and port where ejabberd listens for CAPTCHA requests.
%%
%%{captcha_host, "example.org:5280"}.

%%
%% Limit CAPTCHA calls per minute for JID/IP to avoid DoS.
%%
%%{captcha_limit, 5}.

%%%.  ========
%%%'  MODULES

%%
%% Modules enabled in all ejabberd virtual hosts.
%%
{modules,
 [
  {mod_adhoc,    []},
  {mod_announce, [{access, announce}]}, % recommends mod_adhoc
  {mod_blocking,[]}, % requires mod_privacy
  {mod_caps,     []},
  {mod_configure,[]}, % requires mod_adhoc
  {mod_disco,    []},
  %%{mod_echo,   [{host, "echo.localhost"}]},
  {mod_irc,      []},
  {mod_http_bind, []},
  %%{mod_http_fileserver, [
  %%                  {docroot, "/var/www"},
  %%                  {accesslog, "/var/log/ejabberd/access.log"}
  %%                 ]},
  {mod_last_odbc,     []},
  {mod_muc_odbc,      [
              {host, "conference.jabber.example.org"},
              {access, muc},
              {access_create, muc_create},
              {access_persistent, muc_create},
              {access_admin, muc_admin},

              {history_size, 1000}
             ]},
  %%{mod_muc_log,[]},
  {mod_offline_odbc, [{access_max_user_messages, max_user_offline_messages}]},
  {mod_ping,     []},
  %%{mod_pres_counter,[{count, 5}, {interval, 60}]},
  {mod_privacy_odbc, []},
  {mod_private_odbc, []},
```

```
  %%{mod_proxy65,[]},
  {mod_pubsub,   [
             {access_createnode, pubsub_createnode},
             {ignore_pep_from_offline, true}, % reduces resource comsumption, but XEP incompliant
             %%{ignore_pep_from_offline, false},  % XEP compliant, but increases resource
comsumption
             {last_item_cache, false},
             {plugins, ["flat", "hometree", "pep"]}  % pep requires mod_caps
            ]},
  {mod_register, [

             %%
             %% Protect In-Band account registrations with CAPTCHA.
             %%
             %%{captcha_protected, true},

             %%
             %% Set the minimum informational entropy for passwords.
             %%
             %%{password_strength, 32},

             %%
             %% After successful registration, the user receives
             %% a message with this subject and body.
             %%
             {welcome_message, {"Welcome!",
                                "Hi.\nWelcome to this XMPP server."}},

             %%
             %% When a user registers, send a notification to
             %% these XMPP accounts.
             %%
             %%{registration_watchers, ["admin1 [at] example [dot] org"]},

             %%
             %% Only clients in the server machine can register accounts
             %%
             {ip_access, [{allow, "127.0.0.0/8"},
                          {deny, "0.0.0.0/0"}]},

             %%
             %% Local c2s or remote s2s users cannot register accounts
             %%
             %%{access_from, deny},

             {access, register}
            ]},
  %%{mod_register_web, [

             %%
             %% When a user registers, send a notification to
             %% these XMPP accounts.
             %%
             %%{registration_watchers, ["admin1 [at] example [dot] org"]}
  %%           ]},
  {mod_roster_odbc,   []},
  %%{mod_service_log,[]},
  {mod_shared_roster,[]},
  {mod_stats,    []},
```

```
 {mod_time,     []},
 {mod_vcard_odbc,    []},
 {mod_version,  []}
]}.

%%
%% Enable modules with custom options in a specific virtual host
%%
%%{host_config, "localhost",
%% [{{add, modules},
%%   [
%%    {mod_echo, [{host, "mirror.localhost"}]}
%%   ]
%%  }
%% ]}.

%%%.
%%%'

%%% $Id$

%%% Local Variables:
%%% mode: erlang
%%% End:
%%% vim: set filetype=erlang tabstop=8 foldmarker=%%%',%%%. foldmethod=marker:
```

Изменения в конфигурации (выделенные красным) минимальные...

Включим сервис, добавив необходимую строку в **/etc/rc.conf**:

```
# echo '# Jabber server' >> /etc/rc.conf
# echo 'ejabberd_enable="YES"' >> /etc/rc.conf
```

После этого стартуем сервис:

```
# sh /usr/local/etc/rc.d/ejabberd start
```

Проверяем, все ли в порядке...
В логах обнаружил следующие записи:

```
# cat /var/log/ejabberd/ejabberd.log


=ERROR REPORT==== 2014-03-25 12:25:31 ===
C(<0.37.0>:ejabberd_check:63) : ejabberd is configured to use 'mysql', but the following Erlang
modules are not installed: '[mysql,
                                                                mysql_auth,
                                                                mysql_conn,
                                                                mysql_recv]'


=INFO REPORT==== 2014-03-25 12:25:31 ===
    application: ejabberd
    exited: {bad_return,{{ejabberd_app,start,[normal,[]]},
                         {'EXIT',database_module_missing}}}
    type: temporary
```

Ага... Кажется **Erlang** еще "не научился" работать с **mysql**. Нужно научить :)
Остановим **ejabberd** и установим необходимое расширение **erlang**:

```
# cd /usr/ports/databases/erlang-mysql && make install clean && rehash
```

После этого повторно запускаем **ejabberd**:

```
# sh /usr/local/etc/rc.d/ejabberd start
```

После этого сервис корректно запустился в штатном режиме, что можно проверить следующей командой:

```
# /usr/local/etc/rc.d/ejabberd status
ejabberd is running.
```

Кстати, еще один момент... При тестировании обнаружил, что **ejabberd 2.1.13** не работает с **MySQL 5.6**. При этом в логи сыпались следующие ошибки:

```
=ERROR REPORT==== 2014-03-28 01:39:26 ===
Error in process <0.32058.2> on node 'ejabberd@localhost' with exit value: {{badmatch,<<0 bytes
>>},[{mysql_conn,greeting,2,[{file,"mysql_conn.erl"},{line,344}]},{mysql_conn,mysql_init,5,[{file,
"mysql_conn.erl"},{line,315}]},{mysql_conn,init,7,[{file,"mysql_conn.erl"},{line,240}]}]}

=ERROR REPORT==== 2014-03-28 01:39:26 ===
Error in process <0.32054.2> on node 'ejabberd@localhost' with exit value: {{badmatch,<<0 bytes
>>},[{mysql_conn,greeting,2,[{file,"mysql_conn.erl"},{line,344}]},{mysql_conn,mysql_init,5,[{file,
"mysql_conn.erl"},{line,315}]},{mysql_conn,init,7,[{file,"mysql_conn.erl"},{line,240}]}]}

=ERROR REPORT==== 2014-03-28 01:39:26 ===
Error in process <0.32063.2> on node 'ejabberd@localhost' with exit value: {{badmatch,<<0 bytes
>>},[{mysql_conn,greeting,2,[{file,"mysql_conn.erl"},{line,344}]},{mysql_conn,mysql_init,5,[{file,
"mysql_conn.erl"},{line,315}]},{mysql_conn,init,7,[{file,"mysql_conn.erl"},{line,240}]}]}

=INFO REPORT==== 2014-03-28 01:39:31 ===
I(<0.2181.0>:ejabberd_odbc:226) : mysql connection failed:
** Reason: "timed out"
** Retry after: 30 seconds

=INFO REPORT==== 2014-03-28 01:39:31 ===
I(<0.2175.0>:ejabberd_odbc:226) : mysql connection failed:
** Reason: "timed out"
** Retry after: 30 seconds

=INFO REPORT==== 2014-03-28 01:39:31 ===
I(<0.2172.0>:ejabberd_odbc:226) : mysql connection failed:
** Reason: "timed out"
** Retry after: 30 seconds
```

Разбираться более детально в чем проблема не стал, проверил только на совместимость **ejabberd 2.1.13** с **MySQL 5.0**, **MySQL 5.1** и **MySQL 5.5**. Возможно это пофиксят в следующих релизах.

Итак, продолжим... Создадим пользователя, которого мы в конфигурационном файле указали администратором:

```
# ejabberdctl register muff jabber.example.org password_here
```

После этого можно логиниться в web-админку, которая доступна по адресу **http://jabber.example.org:5280/admin/** .

Выполнив авторизацию в админке, остальных пользователей можно будет добавлять уже прямо в админ-панели.

Скриншот админ-панели:



[4]

Сервер работает, подключены конференции, общие контакты... Остальное уже допиливается под определенные задачи...

Источник (получено *2026-11-06 02:33*):
http://muff.kiev.ua/content/ejabberd-nastroika-sobstvennogo-jabber-servera

Ссылки:
[1] http://muff.kiev.ua/content/openfire-nastroika-sobstvennogo-jabber-servera-obmena-soobshcheniyami
[2] http://muff.kiev.ua/content/bind-9-nastroika-dns-servera
[3] http://muff.kiev.ua/content/mysql-ustanovka-i-bazovaya-nastroika
[4] http://muff.kiev.ua/files/imagepicker/1/ejabberd_01.jpg