Re: pine program and mail services with FC6 System

Source: http://linux.derkeiler.com/Mailing−Lists/Fedora/2007−02/msg01095.html

- *From*: Tim <ignored_mailbox@xxxxxxxxxxxx>
- *Date*: Wed, 07 Feb 2007 03:14:32 +1030

Tim:

> Back when I started using on FC4, I don't recall having to configure anything, just start the service, the defaults were okay (as far as I can remember). Looking at a later version, it appears everything is commented out, though the file suggests it's showing you the default options.

edwardspl@xxxxxxxxxx:

> Just run "/etc/rc.d/init.d/dovecot", right ?

Something like that. I use the service command, like "service dovecot start", I've haven't messed with init scripts in that manner.

Once you'd got it working fine, you'd then set the script to be started when the system boots, automatically.

> So, would you mind send your old version as a sample for me (reference) ?

Remember this is the FC4 configuration file, only use it for hints. And I only use this for internal mail, behind a firewall and only accessible within a private LAN. I've never bothered configuring *it* to be safe from external access. It was set up almost a couple of years ago, and worked without any further tweaking.

−−
(This PC runs FC4, my others FC5 & FC6, in case that's important to the thread)

Don't send private replies to my address, the mailbox is ignored.
I read messages from the public lists.

```
## Dovecot 1.0 configuration file

# Base directory where to store runtime data.
#base_dir = /var/run/dovecot/

# Protocols we want to be serving:
# imap imaps pop3 pop3s
protocols = imap imaps pop3 pop3s

# IP or host address where to listen in for connections. It's not currently
# possible to specify multiple addresses. "*" listens in all IPv4 interfaces.
# "[::]" listens in all IPv6 interfaces, but may also listen in all IPv4
# interfaces depending on the operating system. You can specify ports with
# "host:port".
imap_listen = [::]
pop3_listen = [::]

# IP or host address where to listen in for SSL connections. Defaults
# to above non−SSL equilevants if not specified.
imaps_listen = [::]
pop3s_listen = [::]

# Disable SSL/TLS support.
#ssl_disable = no

# PEM encoded X.509 SSL/TLS certificate and private key. They're opened before
# dropping root privileges, so keep the key file unreadable by anyone but
# root. Included doc/mkcert.sh can be used to easily generate self−signed
# certificate, just make sure to update the domains in dovecot−openssl.cnf
ssl_cert_file = /etc/pki/dovecot/dovecot.pem
ssl_key_file = /etc/pki/dovecot/private/dovecot.pem

# SSL parameter file. Master process generates this file for login processes.
# It contains Diffie Hellman and RSA parameters.
#ssl_parameters_file = /var/run/dovecot/ssl−parameters.dat

# How often to regenerate the SSL parameters file. Generation is quite CPU
# intensive operation. The value is in hours, 0 disables regeneration
# entirely.
#ssl_parameters_regenerate = 24

# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that 127.*.*.* and
# IPv6 ::1 addresses are considered secure, this setting has no effect if
# you connect from those addresses.
#disable_plaintext_auth = yes

# Use this logfile instead of syslog(). /dev/stderr can be used if you want to
# use stderr for logging (ONLY /dev/stderr − otherwise it is closed).
#log_path =
```

```
# For informational messages, use this logfile instead of the default
#info_log_path =

# Prefix for each line written to log file. % codes are in strftime(3)
# format.
#log_timestamp = "%b %d %H:%M:%S "

##
## Login processes
##

# Directory where authentication process places authentication UNIX sockets
# which login needs to be able to connect to. The sockets are created when
# running as root, so you don't have to worry about permissions. Note that
# everything in this directory is deleted when Dovecot is started.
login_dir = /var/run/dovecot−login

# chroot login process to the login_dir. Only reason not to do this is if you
# wish to run the whole Dovecot without roots.
# http://wiki.dovecot.org/Rootless
#login_chroot = yes


##
## IMAP login process
##

login = imap

# Executable location.
#login_executable = /usr/libexec/dovecot/imap−login

# User to use for the login process. Create a completely new user for this,
# and don't use it anywhere else. The user must also belong to a group where
# only it has access, it's used to control access for authentication process.
# Note that this user is NOT used to access mails.
# http://wiki.dovecot.org/UserIds
#login_user = dovecot

# Set max. process size in megabytes. If you don't use
# login_process_per_connection you might need to grow this.
#login_process_size = 32

# Should each login be processed in it's own process (yes), or should one
# login process be allowed to process multiple connections (no)? Yes is more
# secure, espcially with SSL/TLS enabled. No is faster since there's no need
# to create processes all the time.
#login_process_per_connection = yes

# Number of login processes to create. If login_process_per_user is
# yes, this is the number of extra processes waiting for users to log in.
```

```
#login_processes_count = 3

# Maximum number of extra login processes to create. The extra process count
# usually stays at login_processes_count, but when multiple users start logging
# in at the same time more extra processes are created. To prevent fork−bombing
# we check only once in a second if new processes should be created − if all
# of them are used at the time, we double their amount until limit set by this
# setting is reached. This setting is used only if login_process_per_use is yes.
#login_max_processes_count = 128

# Maximum number of connections allowed in login state. When this limit is
# reached, the oldest connections are dropped. If login_process_per_user
# is no, this is a per−process value, so the absolute maximum number of users
# logging in actually login_processes_count * max_logging_users.
#login_max_logging_users = 256

##
## POP3 login process
##

# Settings default to same as above, so you don't have to set anything
# unless you want to override them.

login = pop3

# Exception to above rule being the executable location.
#login_executable = /usr/libexec/dovecot/pop3−login

##
## Mail processes
##

# Maximum number of running mail processes. When this limit is reached,
# new users aren't allowed to log in.
#max_mail_processes = 1024

# Show more verbose process titles (in ps). Currently shows user name and
# IP address. Useful for seeing who are actually using the IMAP processes
# (eg. shared mailboxes or if same uid is used for multiple accounts).
#verbose_proctitle = no

# Show protocol level SSL errors.
#verbose_ssl = no

# Valid UID range for users, defaults to 500 and above. This is mostly
# to make sure that users can't log in as daemons or other system users.
# Note that denying root logins is hardcoded to dovecot binary and can't
# be done even if first_valid_uid is set to 0.
#first_valid_uid = 500
#last_valid_uid = 0
```

```
# Valid GID range for users, defaults to non−root/wheel. Users having
# non−valid GID as primary group ID aren't allowed to log in. If user
# belongs to supplementary groups with non−valid GIDs, those groups are
# not set.
#first_valid_gid = 1
#last_valid_gid = 0

# Grant access to these extra groups for mail processes. Typical use would be
# to give "mail" group write access to /var/mail to be able to create dotlocks.
#mail_extra_groups =

# ':' separated list of directories under which chrooting is allowed for mail
# processes (ie. /var/mail will allow chrooting to /var/mail/foo/bar too).
# This setting doesn't affect login_chroot or auth_chroot variables.
# WARNING: Never add directories here which local users can modify, that
# may lead to root exploit. Usually this should be done only if you don't
# allow shell access for users. See doc/configuration.txt for more information.
#valid_chroot_dirs =

# Default chroot directory for mail processes. This can be overridden by
# giving /./ in user's home directory (eg. /home/./user chroots into /home).
#mail_chroot =

# Default MAIL environment to use when it's not set. By leaving this empty
# dovecot tries to do some automatic detection as described in
# doc/mail−storages.txt. There's a few special variables you can use:
#
# %u − username
# %n − user part in user@domain, same as %u if there's no domain
# %d − domain part in user@domain, empty if user there's no domain
# %h − home directory
#
# You can also limit a width of string by giving the number of max. characters
# after the '%' character. For example %1u gives the first character of
# username. Some examples:
#
# default_mail_env = maildir:/var/mail/%1u/%u/Maildir
# default_mail_env = mbox:~/mail/:INBOX=/var/mail/%u
# default_mail_env = mbox:/var/mail/%d/%n/:INDEX=/var/indexes/%d/%n
#
#default_mail_env =

# Space−separated list of fields to cache for all mails. Currently these
# fields are allowed followed by a list of commands they speed up:
#
# Envelope − FETCH ENVELOPE and SEARCH FROM, TO, CC, BCC, SUBJECT,
# SENTBEFORE, SENTON, SENTSINCE, HEADER MESSAGE−ID,
# HEADER IN−REPLY−TO
# Body − FETCH BODY
# Bodystructure − FETCH BODY, BODYSTRUCTURE
# MessagePart − FETCH BODY[1.2.3] (ie. body parts), RFC822.SIZE,
```

```
# SEARCH SMALLER, LARGER, also speeds up BODY/BODYSTRUCTURE
# generation. This is always set with mbox mailboxes, and
# also default with Maildir.
#
# Different IMAP clients work in different ways, that's why Dovecot by default
# only caches MessagePart which speeds up most operations. Whenever client
# does something where caching could be used, the field is automatically marked
# to be cached later. For example after FETCH BODY the BODY will be cached
# for all new messages. Normally you should leave this alone, unless you know
# what most of your IMAP clients are. Caching more fields than needed makes
# the index files larger and generate useless I/O.
#
# With maildir there's one extra optimization − if nothing is cached, indexing
# the maildir becomes much faster since it's not opening any of the mail files.
# This could be useful if your IMAP clients access only new mails.

#mail_cache_fields = MessagePart

# Space−separated list of fields that Dovecot should never set to be cached.
# Useful if you want to save disk space at the cost of more I/O when the fields
# needed.
#mail_never_cache_fields =

# Workarounds for various client bugs:
# oe6−fetch−no−newmail:
# Never send EXISTS/RECENT when replying to FETCH command. Outlook Express
# seems to think they are FETCH replies and gives user "Message no longer
# in server" error. Note that OE6 still breaks even with this workaround
# if synchronization is set to "Headers Only".
# outlook−idle:
# Outlook and Outlook Express never abort IDLE command, so if no mail
# arrives in half a hour, Dovecot closes the connection. This is still
# fine, except Outlook doesn't connect back so you don't see if new mail
# arrives.
# outlook−pop3−no−nuls:
# Outlook and Outlook Express hang if mails contain NUL characters.
# This setting replaces them with 0x80 character.
#client_workarounds =

# Dovecot can notify client of new mail in selected mailbox soon after it's
# received. This setting specifies the minimum interval in seconds between
# new mail notifications to client − internally they may be checked more or
# less often. Setting this to 0 disables the checking.
# NOTE: Evolution client breaks with this option when it's trying to APPEND.
#mailbox_check_interval = 0

# Like mailbox_check_interval, but used for IDLE command.
#mailbox_idle_check_interval = 30

# Allow full filesystem access to clients. There's no access checks other than
# what the operating system does for the active UID/GID. It works with both
```

```
# maildir and mboxes, allowing you to prefix mailboxes names with eg. /path/
# or ~user/.
#mail_full_filesystem_access = no

# Maximum allowed length for custom flag name. It's only forced when trying
# to create new flags.
#mail_max_flag_length = 50

# Save mails with CR+LF instead of plain LF. This makes sending those mails
# take less CPU, especially with sendfile() syscall with Linux and FreeBSD.
# But it also creates a bit more disk I/O which may just make it slower.
#mail_save_crlf = no

# Use mmap() instead of read() to read mail files. read() seems to be a bit
# faster with my Linux/x86 and it's better with NFS, so that's the default.
#mail_read_mmaped = no

# By default LIST command returns all entries in maildir beginning with dot.
# Enabling this option makes Dovecot return only entries which are directories.
# This is done by stat()ing each entry, so it causes more disk I/O.
# (For systems setting struct dirent−>d_type, this check is free and it's
# done always regardless of this setting)
#maildir_stat_dirs = no

# Copy mail to another folders using hard links. This is much faster than
# actually copying the file. This is problematic only if something modifies
# the mail in one folder but doesn't want it modified in the others. I don't
# know any MUA which would modify mail files directly. IMAP protocol also
# requires that the mails don't change, so it would be problematic in any case.
# If you care about performance, enable it.
#maildir_copy_with_hardlinks = no

# Check if mails' content has been changed by external programs. This slows
# down things as extra stat() needs to be called for each file. If changes are
# noticed, the message is treated as a new message, since IMAP protocol
# specifies that existing messages are immutable.
#maildir_check_content_changes = no

# Which locking methods to use for locking mbox. There's three available:
# dotlock: Create <mailbox>.lock file. This is the oldest and most NFS−safe
# solution. If you want to use /var/mail/ like directory, the users
# will need write access to that directory.
# fcntl : Use this if possible. Works with NFS too if lockd is used.
# flock : May not exist in all systems. Doesn't work with NFS.
#
# You can use both fcntl and flock too; if you do the order they're declared
# with is important to avoid deadlocks if other MTAs/MUAs are using both fcntl
# and flock. Some operating systems don't allow using both of them
# simultaneously, eg. BSDs. If dotlock is used, it's always created first.
mbox_locks = fcntl
```

```
# Should we create dotlock file even when we want only a read−lock? Setting
# this to yes hurts the performance when the mailbox is accessed simultaneously
# by multiple processes, but it's needed for reliable reading if no other
# locking methods are available.
#mbox_read_dotlock = no

# Maximum time in seconds to wait for lock (all of them) before aborting.
#mbox_lock_timeout = 300

# If dotlock exists but the mailbox isn't modified in any way, override the
# lock file after this many seconds.
#mbox_dotlock_change_timeout = 30

# umask to use for mail files and directories
#umask = 0077

# Drop all privileges before exec()ing the mail process. This is mostly
# meant for debugging, otherwise you don't get core dumps. Note that setting
# this to yes means that log file is opened as the logged in user, which
# might not work. It could also be a small security risk if you use single UID
# for multiple users, as the users could ptrace() each others processes then.
#mail_drop_priv_before_exec = no

##
## IMAP process
##

# Executable location
#imap_executable = /usr/libexec/dovecot/imap

# Set max. process size in megabytes. Most of the memory goes to mmap()ing
# files, so it shouldn't harm much even if this limit is set pretty high.
#imap_process_size = 256

# Support for dynamically loadable modules.
#imap_use_modules = no
#imap_modules = /usr/lib/dovecot/imap

##
## POP3 process
##

# Executable location
#pop3_executable = /usr/libexec/dovecot/pop3

# Set max. process size in megabytes. Most of the memory goes to mmap()ing
# files, so it shouldn't harm much even if this limit is set pretty high.
#pop3_process_size = 256

# Support for dynamically loadable modules.
#pop3_use_modules = no
```

```
#pop3_modules = /usr/lib/dovecot/pop3

##
## Authentication processes
##

# An Authentication process is a child process used by Dovecot that
# handles the authentication steps. The steps cover an authentication
# mechanism (auth_mechanisms, how the client authenticates in the IMAP or
# POP3 protocol), which password database should be queried (auth_passdb),
# and which user database should be queried (auth_userdb, to obtain
# UID, GID, and location of the user's mailbox/home directory).
#
# You can have multiple processes, though a typical configuration will
# have only one. Each time "auth = xx" is seen, a new process
# definition is started. The point of multiple processes is to be able
# to set stricter permissions. (See auth_user below.)
#
# Just remember that only one Authentication process is asked for the
# password, so you can't have different passwords accessible through
# different process definitions (unless they have different
# auth_mechanisms, and you're ok with having different password for
# each mechanisms).

# Authentication process name.
auth = default

# Specifies how the client authenticates in the IMAP protocol.
# Space separated list of permitted authentication mechanisms:
# anonymous plain digest−md5 cram−md5
#
# anonymous − No authentication required.
# plain − The password is sent as plain text. All IMAP/POP3 clients
# support this, and the password can be encrypted by Dovecot to match
# any of the encryption schemes used in password databases.
# digest−md5 and cram−md5 − both encrypt the password so it is more
# secure in transit, but are not well supported by clients, and
# require that the password database use a matching encryption
# scheme (or be in plaintext).
#
# See auth.txt for more details.
#
# If you are using SSL there is less benefit to digest−md5 and
# cram−md5 as the communication is already encrypted.
auth_mechanisms = plain


# Space separated list of realms for SASL authentication mechanisms that need
# them. You can leave it empty if you don't want to support multiple realms.
# Many clients simply use the first one listed here, so keep the default realm
# first.
```

```
#auth_realms =

# Default realm/domain to use if none was specified. This is used for both
# SASL realms and appending @domain to username in plaintext logins.
#auth_default_realm =

# User database specifies where mails are located and what user/group IDs
# own them. For single−UID configuration use "static".
# http://wiki.dovecot.org/Authentication
# http://wiki.dovecot.org/VirtualUsers
# passwd: /etc/passwd or similiar, using getpwnam()
# passwd−file <path>: passwd−like file with specified location
# static uid=<uid> gid=<gid> home=<dir template>: static settings
# vpopmail: vpopmail library
# ldap <config path>: LDAP, see doc/dovecot−ldap.conf
# pgsql <config path>: a PostgreSQL database, see doc/dovecot−pgsql.conf
auth_userdb = passwd

# Password database specifies only the passwords for users.
# http://wiki.dovecot.org/Authentication
# passwd: /etc/passwd or similiar, using getpwnam()
# shadow: /etc/shadow or similiar, using getspnam()
# pam [<service> | *]: PAM authentication
# passwd−file <path>: passwd−like file with specified location
# vpopmail: vpopmail authentication
# ldap <config path>: LDAP, see doc/dovecot−ldap.conf
# pgsql <config path>: a PostgreSQL database, see doc/dovecot−pgsql.conf
auth_passdb = pam

#auth_executable = /usr/libexec/dovecot/dovecot−auth

# Set max. process size in megabytes.
#auth_process_size = 256

# User to use for the process. This user needs access to only user and
# password databases, nothing else. Only shadow and pam authentication
# requires roots, so use something else if possible. Note that passwd
# authentication with BSDs internally accesses shadow files, which also
# requires roots. Note that this user is NOT used to access mails.
# That user is specified by auth_userdb above.
auth_user = root

# Directory where to chroot the process. Most authentication backends don't
# work if this is set, and there's no point chrooting if auth_user is root.
#auth_chroot =

# Number of authentication processes to create
#auth_count = 1

# List of allowed characters in username. If the user−given username contains
# a character not listed in here, the login automatically fails. This is just
```

```
# an extra check to make sure user can't exploit any potential quote escaping
# vulnerabilities with SQL/LDAP databases. If you want to allow all characters,
# set this value to empty.
#auth_username_chars =
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890.-_@

# Username to use for users logging in with ANONYMOUS SASL mechanism
#auth_anonymous_username = anonymous

# More verbose logging. Useful for figuring out why authentication isn't
# working.
#auth_verbose = no

# Even more verbose logging for debugging purposes. Shows for example SQL
# queries.
#auth_debug = no

# digest-md5 authentication process. It requires special MD5 passwords which
# /etc/shadow and PAM doesn't support, so we never need roots to handle it.
# Note that the passwd-file is opened before chrooting and dropping root
# privileges, so it may be 0600-root owned file.

#auth = digest_md5
#auth_mechanisms = digest-md5
#auth_realms =
#auth_userdb = passwd-file /etc/passwd.imap
#auth_passdb = passwd-file /etc/passwd.imap
#auth_user = imapauth
#auth_chroot =

# if you plan to use only passwd-file, you don't need the two auth processes,
# simply set "auth_methods = plain digest-md5"
--
fedora-list mailing list
fedora-list@xxxxxxxxxx
To unsubscribe: https://www.redhat.com/mailman/listinfo/fedora-list
```