Mail Server

Da Guide@Debianizzati.Org.

VERSIONI COMPATIBILI

DEBIAN ETCH 4.0
DEBIAN LENNY 5.0
DEBIAN SQUEEZE
DEBIAN SID

Indice

- 1 Introduzione
- 2 Postfix
- 3 Procmail
- 4 Server IMAP4
  - 4.1 Creazione directory per ogni utente
  - 4.2 Creazione DB utenze
- 5 Fetchmail
- 6 Mailscanner Spamassasin ClamAV Razor
  - 6.1 Spamassassin
  - 6.2 Clamav
  - 6.3 Mailscanner
- 7 Client IMAP4
- 8 Webmail

Introduzione

L'idea è quella di avere una connessione permanente ad Internet che può ricevere posta dall'esterno e gestire la posta interna alla Lan, quindi, a meno che non abbiate già un dominio, per prima cosa sarà necessario andare su http://www.dyndns.org o un servizio analogo(ad esempio http://www.no-ip.com), registrarsi, scegliersi un dominio (mandare le email a utente@123.231.201.178 non è proprio comodissimo, soprattutto quando il giorno dopo il numero cambia) e associarlo al proprio indirizzo IP. Se avete un IP dinamico, installate sul vostro computer un programmino come `ddclient` (basta installarlo con apt-get e rispondere alle domande) che aggiorna automaticamente l'indirizzo IP associato al dominio ogni volta che vi collegate.

Postfix

Cominciamo con un server MTA, Mail Transport Agent, che riceve la posta (è quello che tiene aperta la porta 25 SMTP in ricezione, per intenderci). Si può scegliere tra exim, postfix, sendmail, qmail e altri. Scegliamo postfix perché è un buon

compromesso in quanto a prestazioni, compatibilità, flessibilità, sicurezza. Per installarlo diamo il solito:

```
# apt-get install postfix
```

Installandolo verrà automaticamente rimosso un altro eventuale MTA già installato con apt, probabilmente exim o exim4 che è quello di default su Debian.

Debconf pone delle domande quando si installa postfix. Grazie a queste possiamo avere un server di posta già praticamente pronto, molto semplice certo, ma funzionante.

Diamo una spiegazione delle varie schermate.

Nella prima vi informa delle scelte possibili: sono previste varie configurazioni; noi sceglieremo *Internet site using smarthost*, che in pratica consiste nell'avere un server che riceve posta e che la invia tutta ad un altro server (quello fornito dal nostro ISP per esempio). Premete ok per passare alla prossima schermata.

Nella schermata successiva scegliamo "Internet with smarthost".

- **Mail name**: sarà quello che appare dopo la chiocciola nell'indirizzo di posta. Ovviamente deve essere il nome valido del vostro server, dal momento che chi vi risponderà vi manderà la posta a quell'indirizzo.
- **SMTP relay host**: qui indichiamo il server di posta a cui facciamo il relay. In parole semplici, quando inviamo una mail al nostro server, esso la spedirà a questo relay che poi la recapiterà. È necessario dato che molti bloccano l'arrivo di email da IP non ritenuti affidabili.
- **Other destinations to accept mail for**: indirizzi che identificano questo server. Quando gli arriva una mail con questa destinazione, capirà che è lui il destinatario. Qua mettete il vostro nome di dominio.
- **Local networks**: quali reti sono abilitate a spedire mail. Non mettendo nulla postfix inserirà tutte le reti connesse. Meglio inserire a mano localhost (127.0.0.0/8) e la nostra LAN (192.168.1.0/24) nel caso la nostra LAN sia 192.168.1.xxx.
- **Use procmail for local delivery**: rispondete SÌ, dato che useremo procmail.
- **Mailbox size limit**: dimensione massima della casella di posta. 0 significa illimitata. Impostatela a vostro piacimento.
- **Where should mail for root go**: questa imposta un alias per l'utente root, dato che non può ricevere posta. Tutto quello che è indirizzato a lui andrà nella casella di un altro utente, non root, che sceglierete voi.

In particolare attenzione a mettere /24 in "local networks" e non /255 o altre robe strane che possono venirvi in mente. Se non sapete cosa vuol dire quel /24, tenete /24. Se sapete cosa vuol dire, non avete bisogno di spiegazioni.

Non preoccupatevi se vi sembra che "local networks" specifichi a postfix di ricevere posta solo da quei computer, in realtà dice a postfix di fare RELAY solo per la posta che arriva da quei computer, la posta che arriverà da internet la riceve lo stesso.

A questo punto dobbiamo fare solo dei piccoli ritocchi. Editiamo dunque il file di configurazione principale che si chiama `/etc/postfix/main.cf`. Le voci da sistemare sono:

myhostname: controlliamo che ci sia il nome giusto. Vedi `/etc/hostname`

Poi in `/etc/aliases` come da esempio giriamo tutti i messaggi di sistema all'utente se vogliamo ricevere tutto noi esempio:

```
haldaemon: root
mail: root
news: root
proftpd: root
sync: root
root: nomeutente
```

nomeutenteX è quello che verrà usato da chi vorrà inviare posta, ad esempio all'indirizzo nomeutente1@tuo.dominio.org. Si possono creare utenze che puntano ad altre utenze, ad esempio: root che punta a utente1, e utente1 che punta ad utenteposta. Il vero e proprio database di postfix si chiama `/etc/aliases.db`, che non va editato a mano. Per sincronizzarlo con le nostre modifiche, lanciamo il comando newaliases:

```
# newaliases
```

che non restituisce alcun output se tutto va bene. Nel caso vi spari fuori qualcosa tipo `duplicate entry:`, editate di nuovo `/etc/aliases`, eliminate il duplicato ed eseguite di nuovo il comando `newaliases`. Tenete presente che non è necessario che siano associati direttamente agli utenti di sistema normali.

Questo esempio riporta in parte la configurazione di `/etc/postfix/main.cf` :

```
# appending .domain is the MUA's job.
append_dot_mydomain = no
# Uncomment the next line to generate "delayed mail" warnings
# delay_warning_time = 4h
mydomain = nomeserver
myhostname = nomeserver.nomedominio.it
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = nomeserver, localhost.localdomain, localhost, nomeserver.nomedominio.it
myorigin = $mydomain
relayhost = out.virgilio.it
relay_domains = $mydestination
mynetworks = 127.0.0.0/8 192.168.0.0/24
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
disable_dns_lookups = yes
sender_canonical_maps = hash:/etc/postfix/sender_canonical
# aggiunta per mailscanner
# header_checks = regexp:/etc/postfix/header_checks
# aggiunta per server da DEBIANIZZATI
# mailbox_command = /usr/bin/spamc | procmail -a "&SENDER &RECIPIENT $EXTENSION"
```

Le due parti aggiunte, lasciatele commentate fino alla fine altrimenti senza aver installato procmail, vi daranno errori.

Ora create i file per far corrispondere i vostri indirizzi email ai vostri utenti locali. Si tratta dei file `/etc/postfix /sender_canonical` (che contiene gli indirizzi che verranno inseriti nel campo `from` per ogni utente al posto di utente@vostra.macchina) .

Questo è un esempio `/etc/postfix/sender_canonical`:

```
root root@myserver.it
gino gino.paoli@myserver.it
www-data security@myserver.it
utente1 pierino@myserver.it
```

se modificate il file `/etc/postfix/sender_canonical` date:

```
# postmap /etc/postfix/sender_canonical''
```

Ora da un utente a vostra scelta provate a verificare se il server funziona.

Dovete avere installato *mailutils* , altrimenti

```
# apt-get install mailutils
```

con il comando `mail` esempio:

```
''nomeuser@nomeserver:~$ mail root
(invio)

Subject: test
(invio)
prova
(ctrl-D)
Cc: www-data
(invio)
nomeuser@nomeserver:~$
```

potete inviare anche le mail verso indirizzi esterni.

Se vi arrivano vuol dire che tutto funziona.

Procmail

In questa guida ho preferito una configurazione in un unico file, intanto perché per me è stata più semplice da gestire e poi perché il lavoro che dovevo svolgere io era semplice. Quindi se non volete complicarvi la vita fate come ho fatto io, altrimenti leggetevi qualche guida e potrete capire un po' meglio.

Postfix così sarebbe a posto, ma non possiamo ancora fare una prova, perché manca procmail, un programma che smista la posta.

Postfix ne potrebbe anche fare a meno, ma con procmail si possono fare cose più belle, per esempio filtrare la posta mettendola in varie directory a seconda dell'user (fittizio) o a seconda della sorgente (vedi mailing list).

Quindi installiamolo.

```
# apt-get install procmail
```

Il suo file di configurazione va messo nella home dell'utente per la posta, ovvero "utenteposta". Creiamolo con i giusti permessi:

```
$ touch /home/utenteposta/.procmailrc
$ chmod 600 /home/utenteposta/.procmailrc
$ chown utenteposta:usergroup /home/utenteposta/.procmailrc
```

A questo punto occorre decidere come farlo, infatti in base alle proprie esigenze e alla complessità del file è possibile seguire due tipi di configurazione del file.

C'è chi crea più file per filtri vari, diversi utenti e altro che poi fanno riferimento al file principale; oppure chi mette tutti i comandi in un unico file.

Per me la scelta del file unico è stata migliore dato che è molto semplice e in un solo file ho tutta la configurazione di procmail.

Quello sotto è un esempio del mio file:

```
# ~/.procmailrc
SHELL=/bin/sh
#log
VERBOSE = yes # impostare a no dopo il debug
LOGABSTRACT = all # produce log MOLTO estesi, impostare a no in seguito
FORMAIL=/usr/bin/formail # path di formail, usato per processare alcune email
SENDMAIL=/usr/sbin/sendmail # path di sendmail
# File di log
#se tutto funziona dopo un po' potete commentarlo
LOGFILE=${HOME}/procmail.log
# Directory della posta
MAILDIR=${HOME}"/.Maildir"
# Cartella di default. Notare lo slash (/) alla fine che indica a Procmail
# di trattare la cartella in formato Maildir (compatibile con il server IMAP
# che configureremo) e non Mbox !!!
DEFAULT=${MAILDIR}/
YEAR=`date +%Y`
MONTH=`date +%m`
## SPAMASSASSIN ##
## Prima di consegnare le mail, le filtriamo tutte con spamassassin
:0fw:
| spamassassin
# Poi salviamo lo spam in una cartella a parte denominata Spam/
# Lo spam identificato da un controllo negli header sul campo
# X-Spam-Status aggiunto da spamassassin quando la mail viene analizzata
:0:
* ^X-Spam-Status: Yes
.Spam/
#utenti
:0:
* ^TO_nomeutente@nomedominio.it
nomeutente/.Maildir/`$FORMAIL -rt -xMessage-Id:`
```

`:0:`: marca l'inizio della sezione;

`* ^TO_` indica l'indirizzo email completo dell'utente.

La successiva riga indica la directory in cui le mail di quell'utente verranno parcheggiate, e con che formato.

Quindi avremo ad esempio per ogni utente la parte prima indicata e correggeremo solo il finale:

```
:0:
* ^TO_nomeutente1@tuo.dominio.com
nomeutente1/.Maildir/`$FORMAIL -rt -xMessage-Id:`

:0:
* ^TO_nomeutente2@tuo.dominio.com
nomeutente2/.Maildir/`$FORMAIL -rt -xMessage-Id:`
```

eccetera.

Nel caso arrivino mail che non soddisfano i filtri precedentemente applicati nel file di configurazione di procmail possiamo scaricarle in una directory precisa aggiungendo queste 3 righe:

```
:0:

* ^TO_

nomeutentex/.Maildir/`$FORMAIL -rt -xMessage-Id:`
```

Precisazione: `` `$FORMAIL -rt -xMessage-Id:` `` serve ovviamente per dare un nome al file.

Server IMAP4

Bene, a questo punto abbiamo un sistema pronto, per quanto semplice, che riceve posta e la smista in directory. Non è ancora funzionante perché le directory non gliele abbiamo ancora create, e prima di farlo dobbiamo scegliere un server IMAP o POP da usare, cioè il server a cui gli utenti si collegheranno per leggere la posta col loro client. In teoria potrebbero leggere la posta direttamente dal filesystem, ma non è molto comodo. Abbiamo varie possibilità, fra cui due sono state prese in considerazione e spiegate: 'courier-imap' e 'dovecot'. Entrambi sono server IMAP, che ci consentono di tenere tutta la posta nel server senza scaricarla nel client. Quale scegliere? Beh, quello che preferite. Ovviamente sono già pronti e pacchettizzati in Debian (dovecot però c'è solo in Debian Sid in questo momento) quindi per installarli usiamo, come al solito, apt-get.

```
# apt-get install courier-imap fam
```

Notare che con courier installiamo anche fam, File Alteration Monitor. Non è un elemento indispensabile, tuttavia courier è già pronto ad usarlo, permettendo che vari utenti possano condividere delle stesse directory e venir immediatamente avvisati dei cambiamenti. Ora che sono installati (o uno o l'altro) abbiamo a disposizione uno strumento che ci permette di finire il lavoro con procmail, cioè creare le directory organizzate correttamente.

Creazione directory per ogni utente

Creiamo una directory per ogni utente che abbiamo specificato. Entriamo nella directory home dell'utente designato come root per la gestione delle email (`/home/utenteposta`) e cominciamo.

```
$ maildirmake.courier /home/utenteposta/.Maildir
$ chown -R utenteposta:usergroup /home/utenteposta/.Maildir

$ maildirmake.courier /home/utenteposta2/.Maildir
$ chown -R utenteposta2:usergroup /home/utenteposta2/.Maildir
```

Ripetete quelle 2 righe di comando per ogni utente che avete aggiunto al file `/etc/aliases`, o comunque ogni utente che deve avere una casella di posta distinta.

Notate che impostiamo l'utente di tutto a utenteposta e i permessi a 700, così utenteposta sarà l'unico oltre a root a poter leggere la posta direttamente dal disco.

A questo punto si può già provare a vedere se il server riceve la posta e la mette nel posto giusto.

Ricordatevi di aprire la porta 25 sul vostro firewall se ne state usando uno, altrimenti non potrete ricevere posta dall'esterno:

```
iptables -I INPUT -p tcp --dport 25 -m state --state NEW -j ACCEPT
```

Aprite il vostro client di posta abituale, la vostra webmail preferita (o chiedete a qualche amico collegato a internet di farlo, così vedete se la posta arriva anche da fuori) e mandate un'email a `nomeutente1@tuo.dominio.com`. Se tutto va bene dentro a `/home/utenteposta/nomeutente1/new/` trovate i files di testo corrispondenti a ciascuna email.

Creazione DB utenze

Come prima, dobbiamo creare il database degli utenti. Oltre a indicare chi si può loggare a IMAP, e guardare la posta, dobbiamo dirgli con quale password e qual è la sua posta. Per fare questo creiamo un file, di tipo database GDBM o DB, che si chiama `/etc/courier/userdb`:

```
# touch /etc/courier/userdb
# chmod 600 /etc/courier/userdb
# chown root:root /etc/courier/userdb
```

Notate che non deve avere permessi alcuni per gli appartenenti del gruppo o per gli altri. Ora recuperiamo l'UID e il GID dell'utente utenteposta, andando a leggere `/etc/passwd`

```
$ grep utenteposta /etc/passwd

utenteposta:x:107:65534::/home/utenteposta:/bin/bash
```

Bene, ora possiamo cominciare a inserire a mano i vari utenti. Usiamo un programma del pacchetto courier, che si chiama userdb.

```
# userdb "john@example.com" set home=/home/utenteposta/ mail=/home/utenteposta/.Maildir/ uid=107 gid=65534
```

Ripetiamo l'operazione per ogni utente.

Manca da inserire la password, però: lo facciamo con un altro programma che si chiama userdbpw, che ci chiede la password criptandola in md5sum:

```
# userdbpw -md5 | userdb "john@example.com" set imappw
```

Ci chiederà di inserire la password due volte, senza farci vedere l'echo sul terminale, e la inserirà nel file userdb.

Notate che in questo modo le password salvate sul disco sono criptate, ma non quelle mandate dal client al server per l'autenticazione.

In quel caso bisogna usare un sistema diverso, di tipo CRAM-MD5 per esempio.

Compiliamo il database con il comando

```
# makeuserdb
```

che creerà due file: `userdb.dat` contenente le informazioni tranne le password e `userdbshadow.dat` che conterrà le password.

L'ultima cosa da fare abilitare l'uso di userdb: aprite il file `/etc/courier/authdaemonrc` e modificate:

```
authmodulelist="authpam"
```

in:

```
authmodulelist="authuserdb"
```

Ora possiamo passare a configurare il demone di autenticazione di courier.

Esso verrà chiamato ogni volta che qualcuno tenta una connessione. Il suo file di configurazione si chiama `/etc/courier/authdaemonrc` ed è molto ben commentato, l'unica opzione da cambiare è `authmodulelist`.

Qua indichiamo quale o quali moduli di autenticazione verranno usati, ovvero in che modo il demone deve recuperare la lista degli utenti e delle password.

Dal momento che abbiamo preparato un database DB, carichiamo il modulo `authuserdb`. Ora che abbiamo sistemato la parte di autenticazione, possiamo far partire il demone che se ne occupa, quindi:

```
# /etc/init.d/courier-authdaemon start
```

Il file di configurazione di courier-imap si chiama `/etc/courier/imapd`. Contiene già tutto quello che serve al demone per funzionare, quindi non lo modificheremo. L'unica opzione che val la pena guardare è `ADDRESS`: essa indica quali indirizzi deve ascoltare.

Non so come inserire un range di indirizzi, quindi lasciamo `0` (prego qualcuno che lo sa di spiegarlo). Avviamo quindi il servizio.

```
# /etc/init.d/courier-imap start
```

Fetchmail

Fetchmail è un programma che diventa molto comodo quando si hanno altre caselle di posta e si vuole concentrare tutto su una.

Ha due modalità di funzionamento, come demone e come normale programma ma noi lo useremo come demone per controllare ad intervalli prefissati la presenza di posta.

Se c'è, verrà scaricata e inoltrata al nostro server che provvederà a smistarla e a recapitarla nella casella locale.

Procediamo ad installarlo:

```
# apt-get install fetchmail
```

Il file di configurazione lo dobbiamo creare in `/etc`, si chiamerà `fetchmailrc`.

```
# touch /etc/fetchmailrc
# chown fetchmail /etc/fetchmailrc
# chmod 600 /etc/fetchmailrc''
```

La cosa più semplice da fare è un copia/incolla di questo file di esempio, che comunque è quello che uso io nel mio server.

Nei commenti c'è la spiegazione delle impostazioni (`man fetchmailrc` per informazioni ulteriori).

```
# Intervallo di tempo che passa fra ogni controllo. 300 secondi = 5 minuti
# 180 = 3 minuti
#
set daemon 300
# Il log delle operazioni viene fatto tramite syslog
set syslog
# Utente a cui viene recapitata la posta se non ce ne sono altri disponibili
#
set postmaster "discarica@example.com"
# Evita di perdere le mail se succede un errore 4xx. Dall'altro lato,
# però, gli errori 5xx diventano più pericolosi
#
set no bouncemail
# Non manda insulti a chi manda spam
#
set no spambounce
# Ignora le stringhe; potrebbero essere usate da script
#
set properties ""
# I default seguenti sono usati nelle connessioni ai vari server, ma
# possono venire sovrascritti dalle impostazioni locali
#
defaults:
# Aggiunge un header di debug
#
tracepolls
# Usa POP3 come protocollo di default
#
protocol POP3
# Ignora gli errori antispam di postfix, dato che è molto lontano
# dalla sicurezza usarli assieme all'opzione bouncemail
#
antispam -1
# Massimo numero di email da forwardare in un colpo
#
batchlimit 100
# Scarica tutte le email, anche quelle marcate come lette
#
fetchall
# Caselle di posta da controllare
#
poll mail.provider.it
username "user" password "pass"
is "john@example.com" here
#
poll in.virgilio.it timeout 60 with protocol imap
username "utente" there with password "xyzxyz"
is "nomeutentex@example.com" here options keep
```

Come potete vedere la sintassi del poll è molto varia. Nel primo esempio abbiamo una casella di posta che verrà controllata e le email che contiene verranno girate all'utente locale `john@example.com`. Nel secondo esempio vediamo la possibilità di aggiungere altre opzioni, ad esempio un timeout oltre al quale fetchmail desiste dal contattare un server, oppure indicare esplicitamente un protocollo, e infine la possibilità di lasciare le email nel server (keep).

Ce ne sono molte altre, la cosa migliore da fare è leggere le pagine di manuale.

La configurazione di fetchmail si esaurisce qui. Facciamo partire il servizio:

```
# /etc/init.d/fetchmail start
```

andando a modificare anche in `/etc/default/fetchmail` mettendo:

```
start_daemon=yes
```

Adesso bisogna aggiungere alla configurazione di procmail 3 righe per dirgli dove mettere le email che sono state spedite all'utente `utente@virgilio.it`, altrimenti le scarterà. Per fare questo andiamo ad editare il file che descrive gli utenti di procmail, `/home/utenteposta/.pm/utenti.rc`, e aggiungiamo:

```
:0:
* ^TO_utente@virgilio.it
nomeutentex/Maildir/`$FORMAIL -rt -xMessage-Id:`
```

In questo modo le mail prelevate da quella casella di posta saranno recapitate nella casella locale dell'utente nomeutentex. Le modifiche a questo file si possono fare al volo senza riavviare procmail, e verranno recepite immediatamente.

Mailscanner Spamassasin ClamAV Razor

A questo punto installiamo con:

```
# apt-get install mailscanner spamassassin clamav razor
```

Spamassassin

Riguardo la configurazione di spamassassin io ho usato webmin, anche se non c'è molto da fare.

Per settaggi particolari di spamassassin vi consiglio di dare un'occhiata al file `/etc/spamassassin/local.cf<code>` `oppure consultare il sito web http://www.yrex.com/spam/spamconfig.php che vi consente di creare un file di configurazione personalizzato rispondendo alle varie domande.`

Abilitiamo spamassassin modificando in <code>/etc/default/spamassassin

```
ENABLE=1
```

Clamav

Fa tutto da solo.

Mailscanner

Dalla guida http://www.mailscanner.info/postfix.html.

Stop Postfix usando il comando:

```
# /etc/init.d/postfix stop
```

Nel file di configurazione di Postfix `/etc/postfix/main.cf` aggiungete questa linea:

```
header_checks = regexp:/etc/postfix/header_checks
```

create il file:

```
# touch /etc/postfix/header_checks
# chmod 644 /etc/postfix/header_checks
```

adesso inserite dentro al file creato la seguente linea:

```
/^Received:/ HOLD
```

Nel vostro `MailScanner.conf` file (probabilmente in `/etc/MailScanner`), avrete 5 settaggi da modificare:

```
Run As User = postfix
Run As Group = postfix
Incoming Queue Dir = /var/spool/postfix/hold
Outgoing Queue Dir = /var/spool/postfix/incoming
MTA = postfix
# per avere i rapporti in italiano
%report-dir% = /etc/MailScanner/reports/it
```

Dovrete anche essere sicuri che postfix possa scrivere in alcune cartelle, andremo ora a modificare i permessi:

```
# chown postfix.postfix /var/spool/MailScanner/incoming
# chown postfix.postfix /var/spool/MailScanner/quarantine
# chown postfix.postfix /var/run/mailscanner
# chown postfix.postfix /var/spool/MailScanner/
# chown postfix.postfix /var/lib/Mailscanner/
# chown postfix.postfix /var/lock/subsys/Mailscanner/
```

A questo punto rilanciamo i servizi:

```
# /etc/rc.d/init.d/
# ./postfix restart
# ./mailscanner restart
```

Client IMAP4

Potete usare il client che più vi aggrada, da Thunderbird su Windows a Evolution su Linux. Per poter inviare la posta in locale dovrete usare:

```
nomeutente@nomeserver.nomedominio
```

esempio:

```
prova@server.pasticcio.it
```

Webmail

Dovete avere i servizi *apache2*, *mysql*, *php4* o *php5*.

A questo punto per rendere più sicuro il sistema ho preferito non aprire la porta 143 (IMAP).

Io subito ho installato `ilohamail`, disponibile come pacchetto che si installa con apt-get ma io ho scelto di installarlo manualmente, dato che avevo avuto problemi con la configurazione, ho copiato l'intero contenuto del file tar.gz nella cartella HTML.

Dopo le giuste configurazioni indirizzate il vostro browser su http://vostrosito/cartella ilohamail dovreste vedere la finestra iniziale di accesso.

Ma dato che Ilohamail mi sembra un progetto abbandonato da qualche annetto ho preferito passare ad altro.

Esistono altri programmi come http://openwebmail.org/ e altri ancora, questa è una vostra scelta.

Adesso sto usando *group office* che sembra essere il migliore in circolazione.

Ottimo il servizio di posta con IMAP, una vera suite per ufficio con calendario, rubrica con vcard, gestione progetti, note e altro ancora il tutto collegato da un database MySQL, davvero un bel prodotto.

L'installazione è semplice, basta seguire le informazioni all'interno del file.

Si scompatta nella cartella web e in pochi passi sarete sbalorditi dalla grafica e dalla velocità.

Il sito ufficiale è http://www.group-office.com/ ma questa è la versione full a pagamento,la potete provare in versione demo on-line visitando il sito, ma esiste anche una versione sviluppata dalla comunità.

Il forum della versione free si trova all'indirizzo: http://www.group-office.com/forum/ mentre qui (http://sourceforge.net /projects/group-office/) potete scaricarlo.

Questa come soluzione mi è sembrata più elegante e completa a confronto del client di posta.

--Mm-barabba 09:16, 3 Apr 2008 (CDT)

Estratto da "http://guide.debianizzati.org/index.php/Mail_Server"

Categoria: Mail server
