

Installing a basic mail server.

POST MASTER

COLIN MURPHY



Is there some element of your email software that bugs you? Do you get the feeling that you've lost control of how your machine exactly handles your email? Maybe you should consider, or reconsider, running your own email server and taking back control.

When it comes to email so many people, even those who should know better, make do with a monolithic *email program*, software which takes all of the responsibilities of creating, processing and displaying your email. This goes very much against the grain for a UNIX/Linux system, where modularity is the order of the day, where all of the separate processes needed for a particular service (in this case email) are handled by individual programs.

This article will describe how you can set up a very basic email server for a dial-up Linux machine with no other network connections. We will have to assume that you have already configured your system to connect to the Internet. This is the simplest example for the most common of situations, where the basics can be learnt. From here you can further configure to suit your own needs. All this article can really hope to do is give you the impetus to start the ball rolling.

There is an ever increasing range of software that you can use to run as your email server and this variety can be enough to dissuade people from ever starting what can be quite a simple and rewarding

task of running (or should that be tinkering with) your own mail server. It would be pointless for us to suggest which software you should choose because that depends so much on your requirements. These requirements differ from case to case, even if only slightly. We've chosen packages here for demonstration purposes only, you really should do the homework, look at what's available and make your own choices for your own situation. Our choice of packages, Postfix and fetchmail was made mainly on the grounds of availability - we expect most of the boxed set distributions to include them - and a balance between ease of configuration and power. You will also need a package to read and write your email with. For this you might like to reconfigure the software you currently use, or better still, experiment with a new package during a testing phase. We'll be using Kmail as an example here.

Postfix

Postfix is an MTA (a Mail Transport Agent) with responsibility for moving mail around from place to place, most importantly, moving new email that you have created from your machine to the big bad world of the Internet. You may already have an MTA installed, if it is Postfix then all well and good. If it is

some other MTA, Sendmail being a likely candidate, some work will need to be done first. Use RPMs to install Postfix, if there is another MTA installed already, your package manager will complain and you will know what to uninstall beforehand.

Postfix comes with lots of documentation which you should at least look at, but don't be put off if it seems unclear, the most basic configuration which we are dealing with doesn't require very much of it. The upshot of it all is that you need to add some lines to one or more of the Postfix command files to configure it. The most important file is 'main.' which will most often be found in the */etc/postfix/* directory. If it's not there try running *locate main.cf* as a command line to get some clues as to where it might be hiding.

```
relayhost = [mail.ispname.com]
defer_transport = SMTP
disable_dns_lookups = yes
```

with *mail.provider.com* from the first line changed to the address to which you upload your mail at the moment to your ISP. Details of what this is could be found by checking your ISP's support pages or by looking at the configuration details of your current email program. For example in Kmail look under *Settings/ Configuration/ Network/ Sending Mail*. It will be something like *post.demon.co.uk* or *smtp.uklinux.co.uk*

Explaining what all this is:

1. The relayhost is the name of your ISP's mail server, which we are going to take advantage of because, hopefully, your ISP is always connected to the Internet
2. Defer_transport is present because we are not always connected to the Internet, so we will have to take responsibility for when our mail server should try to send its mail
3. disable_dns_lookup because, not being connected to a local network, we are unlikely to

have our own local DNS server running, so looking for it would cause problems

On occasions things go wrong and it is usually better to know about it than to bury your head in the sand. Sometimes things will go wrong with email and the mail servers, your local one or those outside will want to tell someone about it. The mail servers will send email to the postmaster, a special user on the system. Obviously you are not going to want to log in as this special user just to wait for something bad to happen, so arrangements are made for the message for the postmaster to be sent somewhere more convenient, say, to your own login. This is done by setting up an alias in the file */etc/postfix/aliases*, which you again need to edit, changing the postmaster entry from root to your most frequently used login name. We should be safe in assuming that root is not your most frequent user.

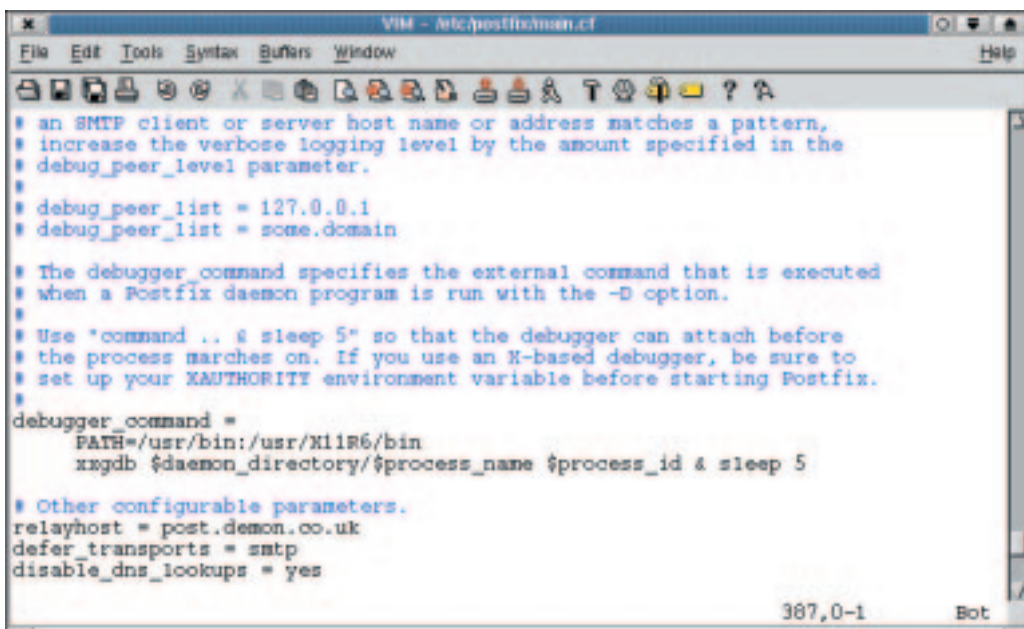
If your ISP supplies the facility to use an unlimited number of email address and if you have taken advantage of this you may want to set up more aliases for those other email addresses. If there is more than one user for your machine you should set up aliases for them as well so that their post will go directly to their login account, unless, by some happy coincidence their email user name, (the bit before the @) matches their login name, then it will happen automatically.

Once you have edited the */etc/postfix/aliases* file with your information, you need to create a database from it by running the command

```
postalias /etc/postfix/aliases
```

This database is required by *postfix*, so, even if you have decided not to make aliases for your users, you still must run this command as part of the configuration process.

To make sure that Postfix looks again at the new configuration you need to restart it with the command



```
VIM - /etc/postfix/main.cf
File Edit Tools Syntax Buffers Window Help
# an SMTP client or server host name or address matches a pattern,
# increase the verbose logging level by the amount specified in the
# debug_peer_level parameter.
#
# debug_peer_list = 127.0.0.1
# debug_peer_list = some.domain
#
# The debugger_command specifies the external command that is executed
# when a Postfix daemon program is run with the -D option.
#
# Use "command .. & sleep 5" so that the debugger can attach before
# the process marches on. If you use an X-based debugger, be sure to
# set up your XAUTHORITY environment variable before starting Postfix.
#
debugger_command =
    PATH=/usr/bin:/usr/X11R6/bin
    xzqdb $daemon_directory/$process_name $process_id & sleep 5
#
# Other configurable parameters.
relayhost = post.demon.co.uk
defer_transports = smtp
disable_dns_lookups = yes
387,0-1 Bot
```

The config file for Postfix. With an editor you need to add the following details to the *main.cf* file, at the end of the file will do fine



```
/etc/rc.d/init.d/postfix reload
```

We just need to set up Kmail to pass any new email to the server that has been installed, so go back to the configuration screen and set Kmail to use `/usr/sbin/sendmail` instead of sending directly to your ISP's mail server directly, that's if you were using Kmail previously. The `/usr/sbin/sendmail` here is actually still part of Postfix, it's just a neat way of allowing Postfix to take over tasks that have been configured for the real Sendmail package. While you are still in Kmail configuration screen you

[right]
Starting to configure

[below]
Initially choose Novice
You will need to configure fetchmail as root, so either log in as root or become a Super User with the `su` command at the terminal prompt. To start `fetchmailconf`, if you have it, just type



should also make sure you have sensible info set up under the identity tag.

Compose an email, either to yourself, or better still to an email echo server, like `echo@tu-Berlin.de`, which, on receipt of your message just sends it back to you, but with all the message headers on display - useful for tracking down any unusual activity. Compose your email and send it from Kmail, but do this while offline.

To get Postfix to actually send the mail, once you have made an online connection, you need to fire off the command

```
/usr/sbin/sendmail -q
```

which is fine for testing purposes, but would become a real pain if you had to type it every time you wanted to send some mail, which is why there is an automatic way. Back in your editor, add to or create the file `/etc/ppp/ip-up.local` with

```
#!/bin/bash
/usr/sbin/sendmail -q
```

and make sure that this file is executable with this command

```
chmod +x /etc/ppp/ip-up.local
```

So now, every time your Internet connection starts up, this script will be run and Postfix will be told to send its stuff. If you can think of any other programs or utilities that you use online you could also add them to this script.

Configuring fetchmail

Postfix and all the other MTAs can look after the movement of emails, in either inward or outward directions using a Simple Mail Transport Protocol - but usually only for machines that have a permanent connection to the Internet. The majority of ISPs expect their dial-up users to retrieve email using a different protocol, usually something called POP3 but others are possible too, which is why we need to call upon the uses of Fetchmail.

Make sure you have fetchmail installed and, if you have it available, the stand alone graphical configuration tool for fetchmail, called `fetchmailconf` which takes what little pain there might have been in configuring away completely.

At its most basic, fetchmail needs to know where to collect your mail from, so you will need to know the address of the mail server from which you will download your mail, details of which will be found on your ISPs support pages, or can be plucked from the configuration details of your current email program. For example, in Netscape you would look at `Edit/ Preferences/ Mail Servers/ Incoming Mail Server`. It also needs to know your user name and password that you use to log into your ISP.

```
fetchmailconf &
```

at the command prompt of your terminal. Choose the Novice Configuration option and enter the name of your ISP's mail server, then your user name at your ISP, then your password. You may also want to check the box to *Suppress deletion of messages after reading* initially, until you are confident that all is working, minimising the chances of losing any of your email. OK all of this information and save the configuration file. Go online and hit the *Test fetchmail* button and wait a little while. It will take a few moments for fetchmail to talk to your ISP's mail server, so the output in the *fetchmail run window* won't appear until your email has downloaded. Hopefully, you will see some output which has *fetchmail: normal termination, status 0* near the end, meaning all went well, or *status 1* meaning that you don't have any email to download, so send yourself some and try again. Anything else means you have a problem - look the status number up in the man page for fetchmail for clues.

```
poll pop.ispname.net protocol pop3 username Z
>YourUserName" password "YourPassword"
```

changing the ispname, YourUserName and YourPassword parameters to your details. Save it and then, at a command prompt do

```
chmod 600 /root/.fetchmailrc
```

which will stop anyone other than the root user from looking at the file and seeing your password.

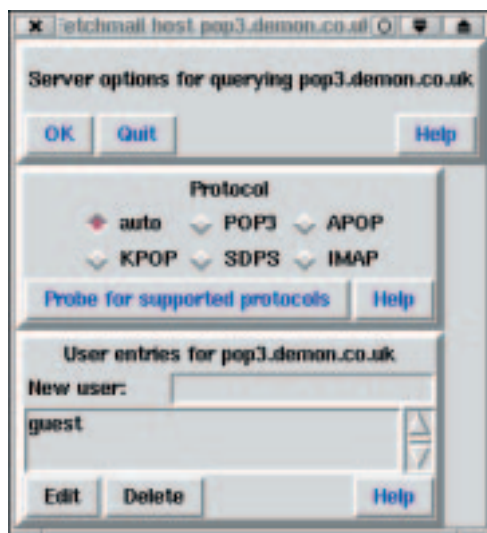
To test it, go online and at a root command prompt enter

```
fetchmail -d0 -v --nosyslog
```

Just like with Postfix, you won't want to be messing around with running fetchmail from the command line every time you want to see if you have mail, so you need to add a line to the */etc/ppp/ip-up.local* file

```
fetchmail -d [600]
```

will poll your mail server every 10 minutes to check and download any new mail.



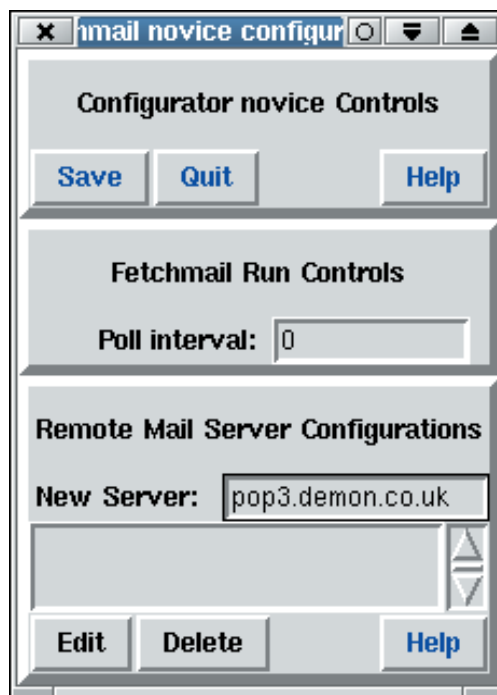
Info

Postfix <http://www.postfix.org/>

Fetchmail <http://www.tuxedo.org/~esr/fetchmail#>

Alternative documentation <http://www.redhat.com/support/docs/faqs/RH-postfix-FAQ/book1.html>

<http://www.mandrakeuser.org/docs/connect/cmail.html>



Enter the ISP details

You should also add to the file */etc/ppp/ip-down.local*

```
fetchmail --quit
```

to stop fetchmail when you log off, otherwise it will start complaining about not being able to find a DNS server.



[far left]
Choose the protocol.

[left]
Is your password safe?
To configure fetchmail without the use of the graphical configuration tool, you will need to edit a file called */root/.fetchmailrc* while logged in as a root user so that it reads