

Retrieving Emails From Remote Servers With fetchmail (Debian Etch)

By Falko Timme

Published: 2007-05-30 17:18

Retrieving Emails From Remote Servers With fetchmail (Debian Etch)

Version 1.0

Author: Falko Timme <ft [at] falkotimme [dot] com>

Last edited 05/17/2007

Fetchmail is a program for retrieving emails from remote servers. Imagine you have five email accounts on five different servers. Of course, you don't want to connect to each of them to get your emails. This is where fetchmail comes into play. If you have a user account on a Linux server, you can make fetchmail download emails from remote servers and put them into just one mailbox (the one of your Linux user), from where you can retrieve them with your email client (e.g. Thunderbird or Outlook).

Or imagine you have an email account at a provider that doesn't do spam- and virus filtering. In that case you could use fetchmail to download the mails to your own server and pipe them through spam- and virus filters (e.g. SpamAssassin and ClamAV) before you download the mails with your email client.

I do not issue any guarantee that this will work for you!

1 Preliminary Note

You need a Linux server with a system user that can receive emails, which means an MTA such as Postfix or Sendmail must be installed on the system. Otherwise fetchmail won't work, because it tries to pass on the downloaded emails to an MTA (Postfix, Sendmail, ...), and the MTA delivers the mails to the user's mailbox (you can configure the system to include spam- and virus scanning in this process, e.g. with [amavisd-new](#) or [procmil](#), but this isn't covered in this tutorial).

I use a Debian Etch system in this tutorial where two users called *falko* and *till* exist.

2 Install fetchmail

In order to install fetchmail, all we have to do is run

```
apt-get install fetchmail
```

3 Configure fetchmail

There are two ways of configuring fetchmail. We can make it run as a daemon with a global configuration file, or we can create a cron job to run fetchmail together with per-user configuration files. I will describe both methods here.

3.1 Run fetchmail As A Daemon With A Global Configuration File

To make fetchmail run as a daemon, we have to edit `/etc/default/fetchmail` and set `START_DAEMON` to `yes`:

```
vi /etc/default/fetchmail
```

```
# This file will be used to declare some vars for fetchmail
#
# Uncomment the following if you dont want localized log messages
# export LC_ALL=C
# Declare here if we want to start fetchmail. 'yes' or 'no'
START_DAEMON=yes
```

Next we must create the configuration file `/etc/fetchmailrc` because the fetchmail daemon won't start if this file doesn't exist. In this file we can specify how the fetchmail daemon should behave as well as the details fetchmail needs to know to retrieve emails from foreign email accounts.

Let's assume `falko` has two email accounts from which we want to retrieve emails:

- First account: server `pop.someprovider.tld`, protocol `POP3`, username `falko@someprovider.tld` (yes, the username is an email address in this case), password `secret`.
- Second account: server `mail.otherprovider.tld`, protocol `POP3`, username `ftimme`, password `verysecurepassword`.

till has one email account:

- Server *mailin.tillsprovider.tld*, protocol *POP3*, username *tbrehm*, password *iwonttellyou*.

So our file */etc/fetchmailrc* could look like this:

```
vi /etc/fetchmailrc
```

```
# /etc/fetchmailrc for system-wide daemon mode
# This file must be chmod 0600, owner fetchmail

set daemon 300 # Pool every 5 minutes
set syslog # log through syslog facility
set postmaster root

set no bouncemail # avoid loss on 4xx errors
# on the other hand, 5xx errors get
# more dangerous...

#####
# Hosts to pool
#####

# Defaults =====
# Set antis spam to -1, since it is far safer to use that together with
# no bouncemail
defaults:
timeout 300
antis spam -1
batchlimit 100
```

```
poll pop.someprovider.tld protocol POP3 user "falko@someprovider.tld" there with password "secret" is falko here
poll mail.otherprovider.tld protocol POP3 user "ftimme" there with password "verysecurepassword" is falko here fetchall
poll mailin.tillsprovider.tld protocol POP3 user "tbrehm" there with password "iwonttellyou" is till here keep
```

At the beginning of the file we have some global options such as `set daemon 300` (which means fetchmail should retrieve emails every 300 seconds) that control the operation of the program. The meanings of the above options are as follows:

- `set daemon`: Set a background poll interval in seconds.
- `set syslog`: Do error logging through syslog.
- `set postmaster`: Give the name of the last-resort mail recipient (default: user running fetchmail, "postmaster" if run by the root user).
- `set no bouncemail`: Direct error mail to the local postmaster (as per the "postmaster" global option above).

Then we have the server and the user options options. These go together into the lines beginning with `poll`; if there are options that are the same for each `poll` line, we can as well specify them before the `poll` lines in a section that begins with `defaults`: (such as `timeout`, `antispam`, and `batchlimit` in our example).

- `timeout`: Server inactivity timeout in seconds (default 300).
- `antispam`: Specify what SMTP returns are interpreted as spam-policy blocks.
- `batchlimit`: Specify the maximum number of messages that will be shipped to an SMTP listener before the connection is deliberately torn down and rebuilt (defaults to 0, meaning no limit).

The poll lines are self-explanatory; as you see fetchmail retrieves emails from both of `falko`'s external email accounts and puts them into one account.

You will notice that the poll lines have different endings (e.g. `nofetchall` (default), `fetchall`, `keep`, `nokeep`). The meanings are as follows:

- `nofetchall`: Retrieve only new messages (default). If nothing else is specified (e.g. `fetchall`, `keep`), this means `nofetchall`.
- `fetchall`: Fetch all messages whether seen or not.
- `keep`: Don't delete seen messages from server.
- `nokeep`: Delete seen messages from server.

To learn more about all available configuration settings, take a look at

```
man fetchmail
```

`/etc/fetchmailrc` must have 600 permissions and must be owned by the user `fetchmail`, so we do the following:

```
chmod 600 /etc/fetchmailrc  
  
chown fetchmail /etc/fetchmailrc
```

Finally, we can start fetchmail:

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

Fetchmail should now download emails and put them into `falko`'s and `till`'s mailboxes (using the MTA). It will repeat this every `set daemon` seconds.

3.2 Use Per-User Configuration Files And Run fetchmail Via Cron

Instead of using a global configuration file as shown in chapter 3.1, we can use per-user configuration files. These must have the name `.fetchmailrc` and must be located in the user's homedir.

We want to create such a file for the user `falko` now.

Then we do this:

```
cd ~/
  
vi .fetchmailrc
```

```
set postmaster falko  
set bouncemail
```

```
poll pop.someprovider.tld protocol POP3 user "falko@someprovider.tld" there with password "secret"  
poll mail.otherprovider.tld protocol POP3 user "ftimme" there with password "verysecurepassword" fetchall
```

The file looks very similar to the file `/etc/fetchmailrc` from chapter 3.1, however you will notice that I don't use the phrase `is falko here` anymore (as `.fetchmailrc` is in `falko`'s homedir, fetchmail knows that the mails should be delivered to `falko`). Of course, you can still use `is falko here`, so the file could look like this as well:

```
set postmaster falko  
set bouncemail  
  
poll pop.someprovider.tld protocol POP3 user "falko@someprovider.tld" there with password "secret" is falko here  
poll mail.otherprovider.tld protocol POP3 user "ftimme" there with password "verysecurepassword" is falko here fetchall
```

To learn more about all available configuration settings, take a look at

```
man fetchmail
```

`.fetchmailrc` must have 600 permissions, so that only `falko` can read from/write to it:

```
chmod 600 ~/.fetchmailrc
```

That's it. Now `falko` can start the retrieval process by running

```
fetchmail
```

or

```
fetchmail -v
```

which shows what's going on.

Of course, *falko* doesn't want to start the retrieval manually every few minutes, so we create a cron job for him. Still as the user *falko*, we run

```
crontab -e
```

and create a cron job like this one (which would start fetchmail every five minutes):

```
*/5 * * * * /usr/bin/fetchmail &> /dev/null
```

4 Links

- Debian: <http://www.debian.org>
- Fetchmail: <http://fetchmail.berlios.de>