

Mastering Linux, part 17

So you've never used Linux or Unix before? *Jarrod Spiga* says think again... Chances are you have if you've ever dealt with web, FTP or email servers.

JARROD SPIGA

Jarrod Spiga is an infrastructure and network engineer who specialises in getting Linux and Windows systems happily coexisting with each other. He currently holds CCNA and numerous MCP certifications, and dabbles in web application development in his spare time.



SKILL LEVEL

- Advanced

TIME TO COMPLETE

- Three hours

REQUIREMENTS

- An installation of Linux (Fedora Core 4 and Red Hat Enterprise Linux ES 4.0 instances were used in the writing of this Workshop);
- Domain Name and Static public IP address (A dynamic public IP and a Dynamic DNS client may also be suitable).

IN THIS SERIES

- **Part 16 - March '06**
Network file services.
- **Part 17 - April '06**
Internet services.
- **Part 18 - May '06**
Backups and troubleshooting.

Fast, flexible and free internet services are one of the reasons Linux is becoming more popular. Even if they're unaware of it, the first exposure that many people have to Linux and Unix is when they're working with a web, FTP or email server. An estimated two-thirds of web and mail servers online run on Linux and Unix systems — so chances are you've been using Linux and Unix from the first time you connected to the net.

SERVING THE WEB

By some margin, the Apache HTTP Server is the most popular web server in use. It's powerful, flexible, stable and serves up HTML data quickly; all the features you'd want from a web server.

While version 2.2 of the Apache HTTP Server has recently been released, not many of the add-in modules are compatible with this latest version. The 2.0.53 release is the latest stable release (at the time of writing). These packages can be installed (or upgraded) by issuing the `yum` command:

```
yum install httpd httpd-devel
```

This will install the Apache HTTPD server, its development tools and any pre-requisite packages. If your Linux distro doesn't support `yum`, you can download and compile the Apache HTTP Server using the instructions found at <http://httpd.apache.org/docs/2.0/install.html>.

Once installed, Apache's runstate can be controlled like any other service running on your system — the start, stop, restart, reload and status arguments (among others) can be added to the `service httpd` command. However, after

the installation, Apache isn't running, nor is it set to automatically start upon boot. The following commands allow us to change this:

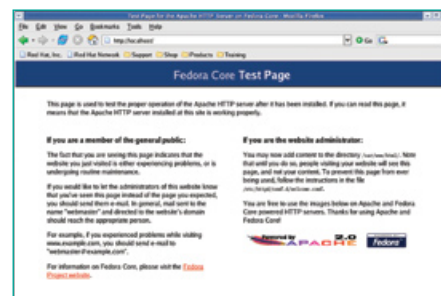
```
/sbin/service httpd start
/sbin/chkconfig httpd --levels 2345 on
```

To test that everything is working well, load up your browser and point it to <http://localhost>. With any luck, a welcome screen should appear.

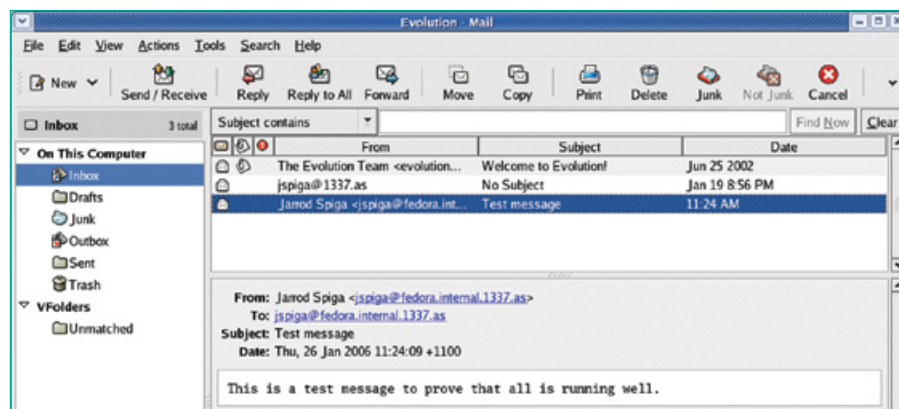
WEB CONFIGURATION

The default Apache configuration is suitable at first for most people. Once started, the server will bind across all network interfaces on port 80, treat the `/var/www/html` directory as the root directory of the server, and set the server admin's email address to `root@hostname` (where `hostname` is the host name of your system).

In order to publish your web site, simply copy your site data to the `/var/www/html` directory.



Testing, testing: once the Apache HTTP Server is installed, a default home page will appear until you upload your site to `/var/www/html`.



Two birds with one stone: by sending an email message to yourself via a POP/IMAP/SMTP email client, you can test that all of these services are running.

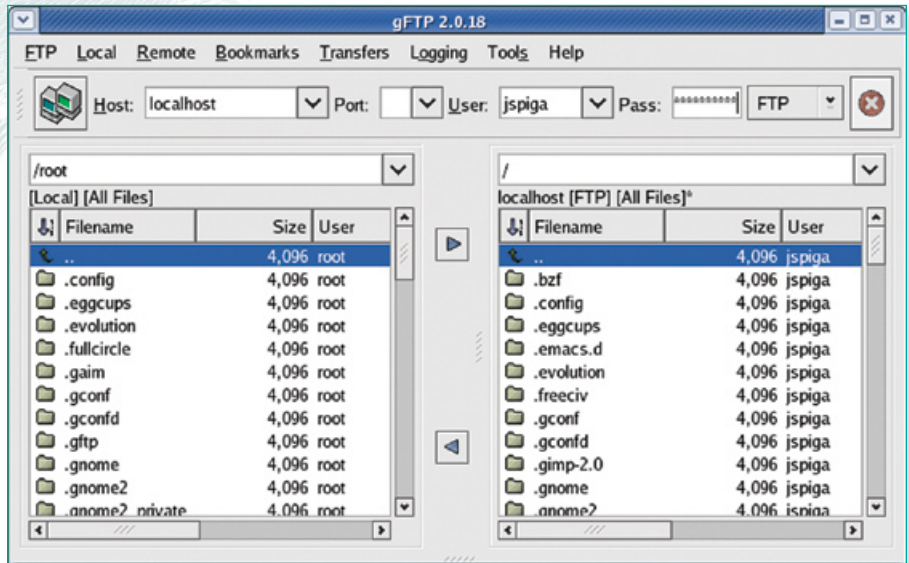
INSIDE INFO

- ProFTPD RPM Binaries:
<ftp://ftp.planetmirror.com/pub/proftpd/distrib/packages/RPMS/proftpd-1.2.10-1.fc1.i386.rpm>
<ftp://ftp.planetmirror.com/pub/proftpd/distrib/packages/RPMS/proftpd-inetd-1.2.10-1.fc1.i386.rpm>
<ftp://ftp.planetmirror.com/pub/proftpd/distrib/packages/RPMS/MD5SUMS>
- Per-user web directories:
http://httpd.apache.org/docs/2.0/howto/public_html.html
- Mapping URLs to file system locations:
<http://httpd.apache.org/docs/2.0/urlmapping.html>
- Authentication and Access Control:
<http://httpd.apache.org/docs/2.0/howto/auth.html>
- Security tips: www.linuxplanet.com/linuxplanet/tutorials/1527/1/
- Setting up Virtual Hosts:
www.activsoftware.com/apache/virtualhosts.cfm
- Postfix FAQ:
www.seaglass.com/postfix/faq.html
- Postfix, Amavis, CalmAV and Spamassassin HOWTO (Debian-based): www.fatofthelam.com/articles/articles.php?pid=22
- Filtering spam with Postfix SPF:
www.freesoftwaremagazine.com/free_issues/issue_02/focus_spam_postfix/
- Fixing common DNS problems: <http://bind8nt.meiway.com/itsaDNSmess.cfm>
- Postfix integration with Microsoft Exchange:
www.personal.umich.edu/~malth/gapuning/postfix/
- Dovecot WIKI: <http://wiki.dovecot.org/>
- ProFTPD documentation:
www.proftpd.org/docs/
- ProFTPD HOWTO (Debian-based):
http://archiv.debianhowto.de/en/proftpd/c_proftpd.html

An index page is configured in the default settings; when no file name is listed in a URL, Apache will attempt to load index.html. If this doesn't exist, a 404 (file not found) error will result.

While the defaults should be fine to get you started, these settings can be changed easily. The Apache framework provides numerous locations for customising the running configuration. The `/etc/httpd/conf/httpd.conf` file is used to define the global configuration. There are a lot of invaluable comments in this file for altering the configuration of Apache (so it does what you want it to). After editing the `httpd.conf` file, you'll need to gracefully reload the web server for your changes to take effect:

```
/sbin/service httpd graceful
```



Application for transfer: without configuring ProFTPD, any user of the Linux system should be able to connect to the server.

Apache's config can also be overridden on a directory-by-directory basis using hidden `.htaccess` files (containing further configuration directives). By creating these files, you can modify Apache's behaviour over one directory (as well as its subdirectories) without affecting the configuration elsewhere on the server. While `.htaccess` can be edited without requiring a server restart, these files are processed each time your server dishes out a web page, making them less efficient than global configuration.

EMAIL FOR ALL

Simple Mail Transfer Protocol is the underlying technology to all email systems, so before you can get any kind of email hosting on your Linux system, you'll need to have an SMTP server installed and running. There are some really powerful (and complicated) SMTP servers available for free download including Sendmail and qmail. These servers are overkill for a home or small business server though — Postfix is much easier and still provides enough features.

Installing Postfix and configuring it to start running on system boot is again simple:

```
yum install postfix
/sbin/chkconfig postfix --levels 2345 on
/sbin/service postfix start
```

To test it all works, establish a telnet connection on port 25 (the port that SMTP servers listen to):

```
telnet localhost 25
```

If you receive any response, it's a good indication that the server is at least running. A full list of SMTP commands can be seen in the SMTP RFC located at www.ietf.org/rfc/rfc0821.txt.

CUSTOMISING FOR YOUR NEEDS

Like most SMTP servers, there are literally hundreds of configuration parameters that can be customised within Postfix, and most of these are set to sensible defaults. However, there are some basic configuration settings that generally need to be changed in order to get Postfix running according to your requirements.

The configuration file is found under `/etc/postfix/main.cf`. The comments in this file explain many of the configuration directives. Many of the directives look similar to shell variables.

The first parameters to check are the `myhostname` and `mydomain`, as the values assigned to these parameters are often used to build the other parameter values. The `myhostname` parameter should specify the fully-qualified domain name of your Linux system, while `mydomain` should represent the parent domain name for your system.

The `mydestination` parameter specifies the domains that your server will accept email for (and deliver to mailboxes on the local system). By default, Postfix accepts email addressed to your system. If this server is to accept email addressed for the domain that your system is a member of, add the `$mydomain` value to the end of this parameter.

The `mynetworks` parameter defines which hosts can relay through your SMTP server without authentication. If this setting is not set correctly, spammers could use your email server to propagate their unwanted messages. You should only permit trusted hosts relay access, including localhost addresses (127.0.0.0/8) as well as your local IP subnet.

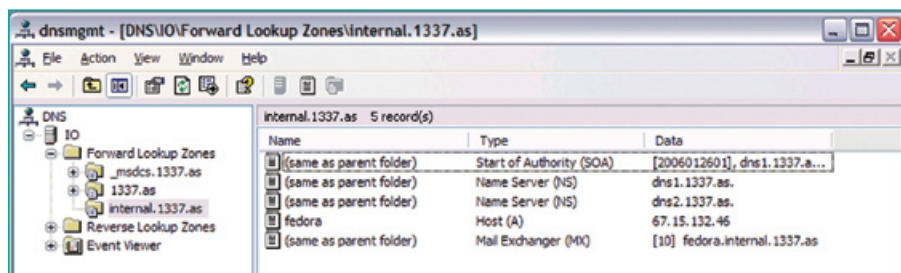
Since many ISPs block outbound connections on port 25 (as a mechanism to prevent spam and worm propagation), you may need to relay outgoing messages through your ISP's SMTP

```

root@fedora:~
File Edit View Terminal Tabs Help
[root@fedora ~]# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
220 fedora.internal.1337.as ESMTP Sendmail 8.13.1/8.13.1; Thu, 26 Jan 2006 21:38:52 +1100
helo fedora.internal.1337.as
250 fedora.internal.1337.as Hello fedora.internal.1337.as [127.0.0.1], pleased to meet you
mail from:jspiga@fedora.internal.1337.as
250 2.1.0 jspiga@fedora.internal.1337.as... Sender ok
rcpt to:jspiga@fedora.internal.1337.as
250 2.1.5 jspiga@fedora.internal.1337.as... Recipient ok
data
354 Enter mail, end with "." on a line by itself
This is a test email message.

```

Speaking SMTP: if you know the commands, you can Telnet to your SMTP server and manually send an email message.



Name	Type	Data
(same as parent folder)	Start of Authority (SOA)	[2006012601], dns1.1337.a...
(same as parent folder)	Name Server (NS)	dns1.1337.as.
(same as parent folder)	Name Server (NS)	dns2.1337.as.
fedora	Host (A)	67.15.132.46
(same as parent folder)	Mail Exchanger (MX)	[10] fedora.internal.1337.as

What's in a name? Setting up your DNS records properly is the key to ensuring that remote hosts can access your services.

server. The `relayhost` parameter allows you to do this. Simply add the name of your ISP's mail server to the argument surrounded by brackets. For instance:

```
relayhost = [mail.m.iinet.net.au]
```

While the tuning of the above parameters will provide a basic Postfix configuration, there are hundreds of other parameters that you could change for more advanced implementations. One thing to remember though is that whenever you make changes to the configuration file, you need to reload Postfix:

```
service postfix reload
```

POP AND IMAP

Once SMTP is up and running, your system has the ability to accept email messages addressed to your domain (and to relay your messages to the outside world). You could use a local application such as PINE to read your email messages directly from the server but, in order to download your messages into an email client, you'll need to set up a POP or IMAP server.

One of the more secure POP/IMAP server combinations available is Dovecot, and installation is trivial on most Linux systems:

```
yum install dovecot
```

```
chkconfig --levels 2345 dovecot on
```

```
service dovecot start
```

The default binary installation of `dovecot` will enable both POP and IMAP methods of retrieving mail messages, and happily read mbox-formatted mailboxes (the default format for Fedora and Red Hat systems).

In general, Dovecot works perfectly "straight out of the box" — no additional configuration is usually required (unless you're looking at implementing advanced authentication methods or implementing SSL).

FILE TRANSFERS

The other major internet service that is frequently hosted on a Linux system is FTP services. The ProFTPD server is one of the more popular ones available and is used by the entire SourceForge community. It too is fairly easy to set up and use, but the installation is a little more complicated.

RPM binaries of the latest stable version (1.2.10 as at the time of writing) have been listed in Inside Info. To install, copy both RPMs across to your home directory and as the root user, run:

```
rpm -Uvh proftpd*.rpm
```

Once the packages have been installed, you'll need to make at least one change to the configuration file, which can be found at `/etc/proftpd.conf`. Locate the line in the configuration file that starts with `Group` and change it to:

```
Group nobody
```

This setting needs to be changed because the "nogroup" group account doesn't exist on Red Hat-based Linux systems — if left unchanged, ProFTPD will not start.

While you're in the configuration file, it's a good idea to comment out all of the lines that allow anonymous access to your FTP server. Anonymous access can be a good thing to enable for short periods of time but, in general, it's a security hole that can be exploited. Even if anonymous access is disabled, regular user accounts, with the exception of the root user, can still log on to the FTP server. The root user is usually blocked for security reasons (this can be overwritten by removing the `root` line from the `/etc/ftpusers` file).

ProFTPD gets installed as a service under the `xinetd` wrapper — so starting the daemon is also a little different to the other services. Xinetd should automatically start upon boot, so restarting this service should start your ftp server:

```
service xinetd restart
```

SORTING OUT DNS

Once all of your services are running on your Linux system, it's time to get them visible to the internet. For HTTP, POP, IMAP and FTP services, all that needs to be configured is a single A record pointing your fully qualified domain name to your IP address. In order for your SMTP server to receive email destined for your domain, you'll also need to set up an MX record that points to the A record.

Things do get a little more difficult if your ISP provides you with a dynamic IP address each time that you connect to the internet. Subscribing to a Dynamic DNS service should at least allow you to set up your A record and possibly an MX record. [ETDC](#)

NEXT MONTH

- In next month's Mastering Linux, we look at performing backups of sections of your Linux system and provide a rundown on how to troubleshoot problems with your Linux system.