

How to set up a mail server on a GNU / Linux system

Step by step guide to install Postfix

Ubuntu + Postfix + Courier IMAP + MySQL + Amavisd-new + SpamAssassin + ClamAV + SASL + TLS + SquirrelMail + Postgrey

Easy to follow howto on setting up a mail server with unlimited users and domains, with IMAP/Pop access, anti-spam, anti-virus, secure authentication, encrypted traffic, web mail interface and more.

Based on an Ubuntu distribution platform, but instructions are distro generic.

[7th edition](#)

Author [Ivar Abrahamsen](#)

License: [Respect](#) (CC by-sa)

Last Update: 2009-01-08

[Contact](#) / [Discuss](#)



Contents

- [Editions](#)
List of different versions of this document.
- [Introduction](#)
Brief description of this document.
 - [Aim](#)
 - [Research](#)
 - [Donate](#)
- [Software](#)
Which software packages are we using and why.
- [Installation](#)
How to install all packages and which ones.
 - [Distrobution](#)
 - [Base Install](#)
 - [Repositories](#)
 - [Packages](#)
- [Configuration](#)
Post install, what to configure for each section, with full command examples.
 - [Firewall \(Shorewall\)](#)
 - [MTA \(Postfix\)](#)
 - [Database \(MySQL\)](#)
 - [Pop/IMAP \(Courier\)](#)
 - [Content Checks \(amavisd-new\)](#)
 - [Anti-Spam \(SpamAssassin\)](#)
 - [Anti-Virus \(ClamAV\)](#)
 - [Policy Check \(PostGrey\)](#)
 - [Authentication \(SASL\)](#)
 - [Encryption \(TLS\)](#)
 - [Webmail \(SquirrelMail\)](#)
 - [Administration \(phpMyAdmin\)](#)
- [Data](#)
Creating the basic stub of data, and how to add your own.
 - [Add users and domains](#)
 - [Common SQL](#)
- [Test](#)
Testing and troubleshooting each element.
- [Initialize](#)
If receiving an already setup machine, a list of actions to do to initialize and configure it.
- [Extend](#)

5th	Released	2006-05	2006-11	Based on Dapper Drake, Ubuntu 6.06 LTS.
6th	Scrapped	2006-11	2007-10	Will be based on Edgy Eft, Ubuntu 6.10. Or may wait for 7.04. May include Domain Key signing. May include my mail admin or my catchall aliases admin.
7th (this)	Released	2008-04	2009-01	Updated, based on Ubuntu 8.04 Hardy Heron. Using Amazon EC2 as example.

Further details available in the [change log](#) and below in the [introduction](#).

[Return to top.](#)

Introduction

Aim

This is a step by step howto guide to set up a mail server on a GNU / Linux system. It is easy to follow, but you end up with a powerfull secure mail server.

The server accepts unlimited domains and users, and all mail can be read via your favourite clients, or via web mail.

It is secure, traffic can encrypted and it will block virtually all spam and viruses.

[Return to top.](#)

Research

Dont take my word for it! Research others opinions and methods. Look at my [references](#), look at [Postfix.org's howtos](#), read the excellent books available (E.g. Kyle's or Hildebrandt's), search the web or read the proper [documentation](#).

If you refer to this howto in your own document, or find useful links, then [let me know](#).

Donate

If you found this howto very useful, spread the word and help others?

If this howto was exceptionally useful why not donate me some **beer** money?

Or buy a [postfix book](#) using my [amazon affiliate links](#) further down?

Or buy a t-shirt from [my t-shirt shop](#)?

Otherwise [send me](#) a **Thank You** note?



[UK](#)



[US](#)



[EU](#)

[Return to top.](#)

Software

What software packages have/will I use and why.

- **OS: Ubuntu Linux**

[www.ubuntu.com](#)

Ah the age old distro argument... Thankfully this set up should work on most distros. I used to base this howto on Mandrake(now Mandriva), and I started this new edition on a Gentoo box. But I don't have the patience for Gentoo, nor the money to stay with Mandriva Power editions. Why Ubuntu? Its free, simple and slick. As Ubuntu is derived from debian the installations used here will be apt-get based. Please refer to my other editions for details on RPM or source based installations.

- **MTA: Postfix**

[www.postfix.org](#)

Simple, free and slick. Yup I am a sucker for anything that works easily. Postfix is powerfull, well established, but not too bloated, and is security concious from the start.

- **Pop/IMAP: Courier IMAP**

[www.courier-mta.org/imap/](#)

My first mail server installation was with Courier. I have not found a reason to change this as again it is simple, and free.

- **Database: MySQL**

[www.mysql.com](#)

Although I use Firebird for my application development, (or Hibernate/C-JDBC hybrids), MySQL is well supported for the sort of lookups required in a mail server.

- **Content Check: Amavisd-new**



- [Base Install](#)
- [Repositories](#)
- [Packages](#)

Distribution

Please refer to [previous edition](#) for a discussion on distribution selection.

Base Install

With installing Ubuntu you have a choice of which base system to install. You may choose server or desktop image or very basic setups. I will assume a server install, but it should not differ.

Ps. I actually built this recent mail server using [Amazon Elastic Computing Cloud \(EC2\)](#). And thus I have created public images of my mail server that you can use. For more details see my [EC2 section](#). If you have your own server, then it is not relevant.

Repositories

Please refer to [previous edition](#) for a details of repository configurations.

Packages

You need to install a whole bunch of packages. We will install them bit by bit. But first check your package sources are correctly pointing to **main multiverse restricted universe** repositories of your current Ubuntu version.

```
sudo vi /etc/apt/sources.list
```

Secondly update your current system:

```
sudo aptitude update
sudo aptitude upgrade
```

MySQL

First we'll install MySQL

```
sudo aptitude install mysql-client mysql-server
```

This will prompt you for a root password. Choose something wise and remember it! For purpose of this tutorial I will set it to **rootPASSWORD**

Postfix

Then we'll install postfix

```
sudo aptitude install postfix postfix-mysql
```

This will prompt you to choose type of email server. Select **internet site** It will also suggest a server name. Correct this if needed.

SASL

```
aptitude install libsasl2-modules-sql libgsasl7 libauthen-sasl-cyrus-perl
```

Courier

```
aptitude install courier-base courier-authdaemon courier-authlib-mysql courier-imap courier-imap-ssl courier-ssl
```

will prompt you about webdirectories. You can say no to this. It will also warn you about the certificate location. Ignore it.

Amavis, SpamAssassin, ClamAV, postgrey

```
aptitude install amavisd-new
aptitude install spamassassin spamc
aptitude install clamav-base libclamav3 clamav-daemon clamav-freshclam
aptitude install postgrey
```

SquirrelMail

```
aptitude install squirrelmail squirrelmail-locales php-pear php5-cli
```

phpMyAdmin


```
aptitude install phpmyadmin
```


Accept apache2 as the web server.


ShoreWall


```
aptitude install shorewall shorewall-doc
```


Amazon provides a firewall/ access control for its servers, so not always needed then, but nice to have. And in all



[Postfix](#)


[Postfix Ge-Pack](#)
 Tobias Wassermann


[The etymological manual](#)
 John Oswald


[The Book of Postfix](#)
 Patrick Koetter


[Building Websites with the ASP.NET C...](#)
 Cristian Daria, K...


[New £27.99 The Definitive Guide to Postfix](#)
 Alan Laudicina

[Privacy Information](#)

```
cp /usr/share/doc/shorewall-common/default-config/interfaces /etc/shorewall/
vi /etc/shorewall/interfaces
```

```
net      eth0      detect      dhcp,tcpflags,logmartians,nosmurfs
```

Then we will configure network zones

```
cp /usr/share/doc/shorewall-common/default-config/zones /etc/shorewall/
vi /etc/shorewall/zones
```

Add the firewall if not there and the internet as a zone.

```
fw      firewall
# loc   ipv4
net     ipv4
```

Then if needed to specify hosts you can do it in this file. E.g. If you want to specify what is your home IP etc.

```
cp /usr/share/doc/shorewall-common/default-config/hosts /etc/shorewall/
vi /etc/shorewall/hosts
```

```
# loc   eth0:192.168.0.0/24
```

Then set what is the default policy for firewall access.

```
cp /usr/share/doc/shorewall-common/default-config/policy /etc/shorewall/
vi /etc/shorewall/policy
```

```
$FW      net      ACCEPT
net      $FW      DROP      info
net      all      DROP      info
# The FOLLOWING POLICY MUST BE LAST
all      all      REJECT     info
```

For safety in case it goes down.

```
cp /usr/share/doc/shorewall-common/default-config/routestopped /etc/shorewall/
vi /etc/shorewall/routestopped
```

```
eth0      0.0.0.0      routeback
```

You may put in a netmask of your ip range if you are more concerned.

Now for the main firewall rules. You can find predetermined macro rules for Shorewall in */usr/share/shorewall*.

```
cp /usr/share/doc/shorewall-common/default-config/rules /etc/shorewall/
vi /etc/shorewall/rules
```

```
SSH/ACCEPT      net      $FW
```

Open for business

Once your server is working come back to this step and open up SMTP and Web access to others.

```
vi /etc/shorewall/rules
```

```
Ping/ACCEPT      net      $FW

# Permit all ICMP traffic FROM the firewall TO the net zone
ACCEPT           $FW      net      icmp

# mail lines
SMTP/ACCEPT      net      $FW
SMTPS/ACCEPT     net      $FW
Submission/ACCEPT      net      $FW
IMAP/ACCEPT      net      $FW
IMAPS/ACCEPT     net      $FW

#web
Web/ACCEPT       net      $FW
```

Firewall configuring is always risky business, as it is easy to lock yourself out. To test the setup syntax, run

```
shorewall check
```

Restart it with

```
/etc/init.d/shorewall restart
```

Then to switch it on during boot:

```
vi /etc/default/shorewall
```

```
startup=1
```

```
# will it be a permanent error or temporary
unknown_local_recipient_reject_code = 450
# how long to keep message on queue before return as failed.
# some have 3 days, I have 16 days as I am backup server for some people
# whom go on holiday with their server switched off.
maximal_queue_lifetime = 7d
# max and min time in seconds between retries if connection failed
minimal_backoff_time = 1000s
maximal_backoff_time = 8000s
# how long to wait when servers connect before receiving rest of data
smtp_helo_timeout = 60s
# how many address can be used in one message.
# effective stopper to mass spammers, accidental copy in whole address list
# but may restrict intentional mail shots.
smtpd_recipient_limit = 16
# how many error before back off.
smtpd_soft_error_limit = 3
# how many max errors before blocking it.
smtpd_hard_error_limit = 12
```

Now we can specify some restrictions. Be careful that each setting is on one line only.

```
# Requirements for the HELO statement
smtpd_helo_restrictions = permit_mynetworks, warn_if_reject reject_non_fqdn_hostname,
                        reject_invalid_hostname, permit
# Requirements for the sender details
smtpd_sender_restrictions = permit_mynetworks, warn_if_reject reject_non_fqdn_sender,
                        reject_unknown_sender_domain, reject_unauth_pipelining, permit
# Requirements for the connecting server
smtpd_client_restrictions = reject_rbl_client sbl.spamhaus.org,
                        reject_rbl_client blackholes.easynet.nl,
                        reject_rbl_client dnsbl.njabl.org
# Requirement for the recipient address
smtpd_recipient_restrictions = reject_unauth_pipelining, permit_mynetworks,
                        reject_non_fqdn_recipient, reject_unknown_recipient_domain,
                        reject_unauth_destination, permit
smtpd_data_restrictions = reject_unauth_pipelining
```

Further restrictions:

```
# require proper helo at connections
smtpd_helo_required = yes
# waste spammers time before rejecting them
smtpd_delay_reject = yes
disable_vrfy_command = yes
```

Next we need to set some maps and lookups for the virtual domains.

```
# not sure of the difference of the next two
# but they are needed for local aliasing
alias_maps = hash:/etc/postfix/aliases
alias_database = hash:/etc/postfix/aliases
# this specifies where the virtual mailbox folders will be located
virtual_mailbox_base = /var/spool/mail/virtual
# this is for the mailbox location for each user
virtual_mailbox_maps = mysql:/etc/postfix/mysql_mailbox.cf
# and their user id
virtual_uid_maps = mysql:/etc/postfix/mysql_uid.cf
# and group id
virtual_gid_maps = mysql:/etc/postfix/mysql_gid.cf
# and this is for aliases
virtual_alias_maps = mysql:/etc/postfix/mysql_alias.cf
# and this is for domain lookups
virtual_mailbox_domains = mysql:/etc/postfix/mysql_domains.cf
# this is how to connect to the domains (all virtual, but the option is there)
# not used yet
# transport_maps = mysql:/etc/postfix/mysql_transport.cf
```

You need to set up an alias file. This is only used locally, and not by your own mail domains.

```
cp /etc/aliases /etc/postfix/aliases
# may want to view the file to check if ok.
# especially that the final alias, eg root goes
# to a real person
postalias /etc/postfix/aliases
```

Next you need to set up the folder where the virtual mail will be stored. This may have already been done by the apt-get. And also create the user whom will own the folders.

```
# to add if there is not a virtual user
mkdir /var/spool/mail/virtual
groupadd virtual -g 5000
useradd virtual -u 5000 -g 5000
chown -R virtual:virtual /var/spool/mail/virtual
```

If using Amazon EC2 run out these in /usr

to 'localhost') then it will communicate over tcp and not the mysql socket. (chroot restriction)

[Return to top.](#)



Database

MySQL

Now we will need to create the tables for thos lookups just specified. First you need to create a user to use in MySQL for mail only. Then you need to create the database, Take note of your chosen mail username and password. You will need the password you specified for **root** during MySQL package installation.

```
# If not already done...
mysqladmin -u root password new_password
# log in as root
mysql -u root -p
# then enter password for the root account when prompted
Enter password:
# then we create the mail database
create database maildb;
# then we create a new user: "mail"
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
ON maildb.* TO 'mail'@'localhost' IDENTIFIED by 'apassword';
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
ON maildb.* TO 'mail'@'%' IDENTIFIED by 'apassword';
exit;
```

Obviously replace **apassword** with your chosen password!

Then you will need to create these tables:

- aliases
- domains
- users

We will create more later on for further extensions, but only these are relevant now.

Log in to mysql as the new mail user

```
mysql -u mail -p maildb
# enter the newly created password
Enter password:
```

Then run this commands to create the tables:

```
CREATE TABLE `aliases` (
  `pkid` smallint(3) NOT NULL auto_increment,
  `mail` varchar(120) NOT NULL default '',
  `destination` varchar(120) NOT NULL default '',
  `enabled` tinyint(1) NOT NULL default '1',
  PRIMARY KEY (`pkid`),
  UNIQUE KEY `mail` (`mail`)
) ;
```

```
CREATE TABLE `domains` (
  `pkid` smallint(6) NOT NULL auto_increment,
  `domain` varchar(120) NOT NULL default '',
  `transport` varchar(120) NOT NULL default 'virtual:',
  `enabled` tinyint(1) NOT NULL default '1',
  PRIMARY KEY (`pkid`)
) ;
```

```
CREATE TABLE `users` (
  `id` varchar(128) NOT NULL default '',
  `name` varchar(128) NOT NULL default '',
  `uid` smallint(5) unsigned NOT NULL default '5000',
  `gid` smallint(5) unsigned NOT NULL default '5000',
  `home` varchar(255) NOT NULL default '/var/spool/mail/virtual',
  `maildir` varchar(255) NOT NULL default 'blah/',
  `enabled` tinyint(3) unsigned NOT NULL default '1',
  `change_password` tinyint(3) unsigned NOT NULL default '1',
  `clear` varchar(128) NOT NULL default 'ChangeMe',
  `crypt` varchar(128) NOT NULL default 'sdtrusfx0Jj66',
  `quota` varchar(255) NOT NULL default '',
  `procmailrc` varchar(128) NOT NULL default '',
  `spamassassinrc` varchar(128) NOT NULL default '',
  PRIMARY KEY (`id`),
  UNIQUE KEY `id` (`id`)
) ;
```

The last few fields in the **users** table are not required, but useful if you extend later.

Next is to edit the MySQL's **my.cnf** file. In Ubuntu/debian this is created by default. In Mandrake I had to manually create a blank one in /etc. But we need to configure it, so:

```
vi /etc/courier/imapd
```

Leave unchanged.

[Return to top.](#)

You know have a basic mail server. You can use this, but I'd recommend continuing. However this is a good point to [test](#) the set up so far and to insert some [data](#) in the db.

I've created an EC2 bundle for this stage: [flurdy-amis/ubuntu-mail-server-simple](#).

[Return to top.](#)

Advanced mail server

Now let's extend this setup with more useful content checks, security and user interfaces.

Content Checks (Anti spam & anti virus)

Amavisd-new

Amavisd ties together all the different ways of checking email content for spam and viruses.

The defaults are pretty good and also the [ubuntu documentation](#) is pretty clear, and recommended.

Here is a tweaked version of it:

Initially we will not enable spam or virus detection! This is so we can get amavis set up to receive, check and pass on emails before we go on and over-complicate it.

All of amavis' configuration files are in `/etc/amavisd`. They are now spread across several files in `conf.d`. Debian and Ubuntu defaults are now very sensible and spread into separate files.

```
cd /etc/amavis.d/conf.d
```

01-debian defaults are fine.

Have a look at

```
less 05-domain-id
```

but don't change anything in it.

Have a look at

```
less 05-node-id
```

but don't change anything in it.

Have a look at

```
less 15-av_scanners
```

but don't change anything in it.

Edit content check file

```
sudo vi 15-content_filter_mode
```

Comment out both virus and spam scans. (Default).

```
# #@bypass_virus_checks_maps = (
#   \#@bypass_virus_checks, \#@bypass_virus_checks_acl, \#@bypass_virus_checks_re);
# #@bypass_spam_checks_maps = (
#   \#@bypass_spam_checks, \#@bypass_spam_checks_acl, \#@bypass_spam_checks_re);
```

Have a look at

```
less 20-debian_defaults
```

but don't change anything in it.

25-amavis_helpers defaults are fine.

30-template-localization defaults are fine.

Edit user file

```
sudo vi 50-user
```

In the middle insert:

```
#####
```



[No, I will not fix your computer](#)

```
$sa_kill_level_deflt = 8.0; # triggers spam evasive actions
#$final_spam_destiny      = D_PASS;
$final_spam_destiny      = D_DISCARD;
```

[Return to top.](#)

Anti-Spam

SpamAssassin

The default config of spam assassin is okay. You could refer to [previous edition](#) for more configuration options.

You do need to tell SpamAssassin to start *smamd* on boot.

```
vi /etc/default/spamassassin
```

```
ENABLED=1
```

One configuration option you could tweak is to enable Bayes and auto learning.

```
vi /etc/spamassassin/local.rf
```

[Return to top.](#)

Anti Virus

ClamAV

ClamAV does not need setting up. Configuration files are in */etc/clamav*, but they are automatically generated, so do not edit.

By default *freshclam*, the daemon that updates the virus definition database, is run 24 times a day. That seems a little excessive, so I tend to set that to once a day.

```
sudo dpkg-reconfigure clamav-freshclam
```

If needed, this will redefine the configuration with a lot of questions. Not needed unless you need to configure.

```
sudo dpkg-reconfigure clamav-base
```

[Return to top.](#)

Postgrey

The default config of postgrey is okay. However you need to tell Postfix to use it.

```
sudo vi /etc/postfix/main.cf
```

And then edit the recipient restrictions:

```
s      mtpd_recipient_restrictions = reject_unauth_pipelining, permit_mynetworks, permit_sasl_authenticated, reject_non_fqdn_rec:
```

You can tweak whitelisting in */etc/postgrey*. You can tweak postgrey configuration by tweaking */etc/default/postgrey*. E.g. delay, auto whitelisting, or reject message.

[Return to top.](#)

You know have an advanced mail server. You can use this, but I'd recommend continuing. However this is a good point to [test](#) the set up so far and to insert some [data](#) in the db.

I've created an EC2 bundle for this stage: [flurdy-amis/ubuntu-mail-server-spam](#).

[Return to top.](#)

Secure mail server

Stopping hackers, phishers, spammers, your boss and your neighbour from accessing your server or the traffic in between is important, and easily done.

Authentication

Normal email traffic between clients and servers are in open plain text. That includes passwords and content of emails.

SASL

Please refer to [previous edition](#) for more detail.

SASL secures the actual authentication (login), by encoding the passwords so that it can be easily intercepted. The rest of the emails are however in clear plain text.



[I read your email](#)




```
-out imapd.pem -nodes -days 999
```

For more details [review last edition](#).

Then you need to edit

```
vi /etc/courier/imapd-ssl
```

By default Ubuntu already points to you certificate

```
TLS_CERTFILE=/etc/courier/imapd.pem
```

Modify this if needed.

Also you if want to restrict IMAP users to SSL/TLS only toggle this setting to 1.

```
IMAP_TLS_REQUIRED=1
```

For maximum compatability it is not wise to restrict to TLS only for the traffic between servers. As this means not all valid emails sent by others can reach your server. However enabling them the option to encrypt is a good idea.

Be aware that the emails are not encrypted on your machine, nor on the server. For this type of client encryption, please refer to [previous edition](#) for more on GnuPG.

In some situations SASL and TLS do not play well together. Those situations are in combinations of storing encrypted passwords, using MD5 authentication over encrypted traffic. I recommend, insisting on TLS traffic with your authenticating clients, which then negates the need for SASL.

You know have an advanced secure mail server. Now is another good point to [test](#) the set up so far and to insert some [data](#) in the db.

Ive created an EC2 bundle for this stage: [flurdy-amis/ubuntu-mail-server-secure](#).

[Return to top](#).

Webmail

Using among others the <https://help.ubuntu.com/community/Squirrelmail> as an updated reference.

Enable web access

You may need to enable web access in the firewall. Check the [firewall configuration](#) if this neccessary.

You need to copy a SquirrelMail configuration to apache.

```
sudo cp /etc/squirrelmail/apache.conf /etc/apache2/sites-available/squirrelmail
```

And enable with this:

```
sudo ln -s /etc/apache2/sites-available/squirrelmail /etc/apache2/sites-enabled/500-squirrelmail
```

Or as Florent recommends, use:

```
sudo a2ensite squirrelmail
```

You may accept the default apache configuration where squirrelmail is folder in all sites. But I prefer virtual hosting. But you dont need to do these next steps.

```
sudo vi /etc/apache2/sites-available/squirrelmail
```

Comment out the alias.

```
# alias /squirrelmail /usr/share/squirrelmail
```

Uncomment the virtual settings., and insert your servers name.

```
# users will prefer a simple URL like http://webmail.example.com

DocumentRoot /usr/share/squirrelmail
ServerName webmail.example.com
```

If you have apache SSL enabled in apache, then you can also uncomment the mod_rewrite section for further security.

Reload apache to activate changes. First test if ok.

```
sudo apache2ctl -t
```

Then reload it.

Please refer to [previous edition](#) for example on htaccess, and mysql user restriction.

You know have a finished mail server. This is as far as the main guide goes. Hope it was clear enough to follow.

Now it is time to insert [data](#), and to [test](#) how it works.

Feel free to [extend it](#) with my suggestions further down.

Ive created an EC2 bundle for this stage: [flurdy-amis/ubuntu-mail-server-webmail](#).

Install DNS Config Tool

Network Management Software & IT Tools for Free. Download Now!
[Spiceworks.com](#)

Computer Network Services

Certified Network Engineers
VPNs, Routers, Firewalls,
Wireless
[www.GenieLogic.com](#)

Ultimate VPS

VPS packages starting at \$24.95/mo Fast, Fully Managed, Easy to use
[www.isomedia.com](#)

Cheap Linux Servers

100% Uptime, World-Class Support! Discount Codes & Server Clearance
[www.ThePlanet.com/Promotions](#)

[Return to top.](#)

Data

- [Add users and domains](#)
- [Common SQL](#)

Add users and domains

So we got a fully set up mail server... Well no, there is no users, domains, no nothing!

Okay, first you need add some default data, some which are required, some which make sense.

Then we'll add your own users and domains.

First the required domains for local mail

```
# Use phpMyAdmin or command line mysql
INSERT INTO domains (domain) VALUES
('localhost'),
('localhost.localdomain');
```

Then some default aliases. Some people say these are not needed, but I'd include them.

```
INSERT INTO aliases (mail,destination) VALUES
('postmaster@localhost','root@localhost'),
('sysadmin@localhost','root@localhost'),
('webmaster@localhost','root@localhost'),
('abuse@localhost','root@localhost'),
('root@localhost','root@localhost'),
('@localhost','root@localhost'),
('@localhost.localdomain','@localhost');
```

Then a root user.

```
INSERT INTO users (id,name,maildir,encrypt) VALUES
('root@localhost','root','root/', encrypt('apassword')) ;
```

Now lets add some proper data.

Say you want this machine to handle data for the fictional domains of "blobber.org", "whopper.nu" and "lala.com".

Then say this machine's name is "mail.blobber.org".

You also have two users called "Xandros" and "Vivita".

You want all mail for *whopper* to go to *xandros*.

There is also a "Karl" user, but he does want all mail forwarded to an external account.

```
INSERT INTO domains (domain) VALUES
('blobber.org'),
('whopper.nu'),
('lala.com');
INSERT INTO aliases (mail,destination) VALUES
('xandros@blobber.org','xandros@blobber.org'),
('vivita@blobber.org','vivita@blobber.org'),
('karl@blobber.org','karl.vovianda@gmail.com'),
('@whopper.nu','xandros@blobber.org'),
('@lala.com','@blobber.org'),
('postmaster@whopper.nu','postmaster@localhost'),
('abuse@whopper.nu','abuse@localhost'),
('postmaster@blobber.org','postmaster@localhost'),
('abuse@blobber.org','abuse@localhost');
INSERT INTO users (id,name,maildir,clear) VALUES
```

```
SELECT al.*
FROM aliases al
LEFT JOIN domains dom
      ON dom.domain = SUBSTRING(al.destination,LOCATE('@',al.destination)+1)
WHERE dom.domain is null
ORDER BY al.enabled, al.destination ASC, al.mail ASC
```

Find all aliases for a certain domain

```
SELECT al.*
FROM aliases al
WHERE SUBSTRING(al.mail,LOCATE('@',al.mail)+1) = 'domain.tld'
ORDER BY al.enabled, al.mail ASC
```

Find all aliases for a certain domains, checking if enabled for both domain and alias

```
select *
from domains d
join aliases a
  on a.mail like concat( '%','@',d.domain)
  and a.enabled = 1
where d.enabled = 1
and d.domain like '%foobar%'
order by d.domain,a.mail
```

[Return to top.](#)

Test

Please refer to [previous edition](#) for how to test your setup. That edition have an extensive testing section.

[Return to top.](#)

Intialize

Brief hints if you receive a ready setup machine (or EC2 AMI), and what then to check and to customize it to your setup.

- Stop services
- Restrict firewall
- Change passwords
- Check configurations
- Set machine name
- Certificates
- Start and test services
- Insert data
- Reload postfix
- Open firewall
- Test

Stop services

First stop services so they wont accidentally do something.

```
sudo /etc/init.d/postfix stop
sudo /etc/init.d/courier-imap-ssl stop
sudo /etc/init.d/courier-imap stop
sudo /etc/init.d/courier-authdaemon stop
sudo /etc/init.d/mysql stop
sudo /etc/init.d/amavisd stop
sudo /etc/init.d/spamassassin stop
sudo /etc/init.d/clamav stop
```

Restrict firewall

Check what the firewall rules are.

```
vi /etc/shorewall/rules
```

Refer to the . Restrict to just SSH access for now.

Change passwords

Next the passwords needs to be changed. For both the system and mysql.

System passwords

Check what users are defined on the system

This can be reset by

```
sudo hostname smtp.yourdomain.com.
```

Al though this does not have to be the same as your postfix mail server name. You may want to specifiy some hosts in hosts file as well,

```
sudo vi /etc/hosts
```

```
127.0.0.1 localhost.localdomain localhost
127.0.0.1 smtp.yourdomain.com smtp
```

Certificates

You could go along with the generated certificates (if they are there, default for Ubuntu). Or if you could create new ones with the correct machine name in them. Especially if this a mail server used by many, and authenticiciy is important. Follow the [TLS certificate instructions](#) for Postfix and Courier.

Start and test services.

Next you need to start your mail services and test them.

```
sudo /etc/init.d/mysql start
sudo /etc/init.d/spamassassin start
sudo /etc/init.d/clamav start
sudo /etc/init.d/amavisd start
sudo /etc/init.d/postfix start
sudo /etc/init.d/courier-imap-ssl start
sudo /etc/init.d/courier-imap start
sudo /etc/init.d/courier-authdaemon start
```

So test tjenesene via [testing section](#).

Insert data

Insert your mail domains, aliases and users using the [data section](#).

Some times there are test data already in the database. Remove them. E.g.;

```
mysql -u mail -p$password maildb
```

```
delete from domains where domain = 'bar.com';
delete from aliases where mail = 'foo@bar.com';
```

Open firewall

Then open up the firewall, follow the world access bit in the [firewall configuration](#). Voila. Up and running. Well we hope.

[Return to top](#).

Extend

Please refer to [previous edition](#) for how and why you can extend this mail server.

By now you should have a fully working system. No point extending and complicating it until then. What next? There are many ways to extend the server, to create your own powerfull customized version.



- [Remote MX mail backup](#)
- [Local file backup](#)
- [Sender ID & SPF](#)
- [Spam reporting](#)
- [White/Black lists](#)
- [PGP & S/MIME](#)
- [Relocation notice](#)
- [Pop-before-SMTP](#)
- [Admin Software](#)
- [Auto Reply](#)
- [Block Addresses](#)
- [Throttle Output](#)
- [Mail Lists](#)
- [Sugestions?](#)

You noticed I added a transport lookup. This is a field in both the domain and the backup tables. In domains it is used to determine how to deliver the email, ie either virtual (correct) or local (not used in this howto). When backing up servers, your also need to specify in the transport field how to connect to the correct servers.

Say you are backup for a friends server, mail.friend.com, for the domains of friend1.com and friend2.com. So you should insert this into your backup table.

```
INSERT INTO backups (domain,transport)
VALUES ('friend1.com' , ':[mail.friend.com]' ),
('friend2.com' , ':[mail.friend.com]' );
```

The :[] tells to connect directly to this server, not doing any more look ups for valid MX servers.

This should now work fine. Further tweaking of the queue values, review these and modify as appropriate. Shorter warning times are good for the sender, so that they realise the email has not arrived yet, but may also be annoying. Tradeoffs.. Look in the first [main.cf configurations](#) for ways to do so.

[Return to top.](#)

Local file backup

Here is rough backup script to backup your configurations and mail folders. You may want to backup the folders separately as they can quickly grow to GBs. Adding this to a cronjob automates this process. Be aware that you should stop postfix and courier while backing up the mail folders. And that if they have grown large, that this may take some time.

```
tar czf mail-config.xxxxx.tgz /etc/postfix /etc/courier /etc/spamassassin /etc/clamav /etc/amavis /etc/mysql/my.cnf
tar czf mail-fold.xxxx.tgz /var/spool/mail/virtual
mysqldump -u mail -p$password -t maildb > data.sql
mysqldump -u mail -p$password -d maildb > schema.sql
tar czf mail-data.xxx.tgz schema.sql data.sql
tar cf mail.xxxxx.tar mail-*.xxxxx.tgz
```

You may combine a full backup with a intermediate update of what has changed recently only.

```
tar --newer-mtime "2005-01-01"
```

[Return to top.](#)

Sender ID & SPF

todo

Further security features is using Microsoft's Sender ID or Pobox's SPF. I'd use SPF as there is much argument over Sender ID.

spf.pobox.com/

www.microsoft.com/mscorp/safety/technologies/senderid/

While SPF should limit who can send mail on behalf of your domains, (so basically less spoofed spam addresses), I do have some technical issues with SPF as the design of it is a bit iffy. That is because of the limitation of DNS and that it has to fit inside the limited TEXT part. No nice XML config file....

While Microsoft is not always entirely evil, as sometimes they do nice things and make some useful software, I would prefer not to be locked into their Sender ID technology.

[Return to top.](#)

Spam reporting

todo

Reporting spam to Pyzor, Razor and SpamCop, for collaboration in spam fighting.

More detail on [SpamCop is here.](#)

<http://pyzor.sourceforge.net/>

<http://razor.sourceforge.net/>

[Return to top.](#)

White/Black Lists

todo

You can implement white and black lists to explicitly allow or block domains and users.

You have already visited the option of a [blackhole list of known open relays](#) in the postfix configuration.

You can implement further lists inside Postfix or SpamAssassin. Amavisd-new already has a few well known white/black listed items in its config files. SpamAssassin also as a feature to automatically learn white lists.

[Return to top.](#)

PGP & S/MIME

Adding support for GnuPG and S/MIME increases individual security.

This is not implemented on the postfix server side, as this totally a client side option.

However SquirrelMail has a GnuPG option. It is a plugin that can be downloaded from their website. Which can then be enabled via SquirrelMail's config script.

Here is how to create a GnuPG key pair.

[Return to top.](#)

Block Addresses

If you use catch alls, which are useful for some domains, then eventually some addresses will be target for spam. You can then either stop the catch all, or stop individual addresses.

By implementing a lookup and adding this restriction to smtpd_recipient_restrictions accomplishes this.

```
check_recipient_access mysql:/etc/postfix/mysql_block_recip.cf,
smtpd_recipient_restrictions = permit_mynetworks, permit_sasl_authenticated, \
    check_recipient_access mysql:/etc/postfix/mysql_block_recip.cf, \
    reject_non_fqdn_recipient, reject_unauth_destination, \
    check_relay_domains
```

Beware of the order is important here, if any options says ok before check_recipient_access it will ignore it.

Next create mysql_block_recip.cf to lookup addresses. Either create a another table, or add a blocked field to aliases table.

[Return to top.](#)

Throttle Output

todo

For some users with restrictions on bandwidth, you may wish to control how much mail is sendt out. Postfix has long refused to implement these features, out of ideolocial beliefs that mail servers should not be restricted. However there are some ways around this. More to come later.

[Return to top.](#)

Mail Lists

Rich Brown has written a howto on adding Mailman, a mail list program, to my howto. [Click here](#) to read it.

Do note it is not part of my howto, so do not contact me regarding it. And although I think it is fine, I can't guarantee it will work.

If you do need assistance or need to talk about it, contact Rich via [his howto](#) or use the [forums](#) for this howto.

If you want a simple mailing list, it can be implemented by simply seperating aliases in the destination field in the aliases table with a comma.

```
INSERT INTO aliases (mail,destination) VALUES
( 'listof@domain.com' , 'john@ppp.com,vic@domain.com,jj@somewhere.tld' );
```

[Return to top.](#)

Google Apps / Gmail

Currently writting this one...

I have for various reasons integrated some Google Apps hosted domains into my mail server. And you can still have good control over the addresses by using your server with Google Apps.

More information on [Google Apps](#).

Why

- Some already have their domain's email hosted with Google.
- Some people prefer Google's web based interface.
- Temporary Migrations.
- Include Google's security features on top of yours.

How

Options

The easiest and simples solution is not to have a domain MXed to your server, and simply alias email to those domains. eg All email to joeblogs.co.uk hosted on your server are forwarded to joeblogs.com hosted with google.

You may set up your own server to simple be a mail server backup (mx) for a domain hosted with google. If you are the first priority in the MX details of the DNS, you still have some control, but not all will obey the priority listing. E.g. spammers, but some valid senders as well.

However the one I use and the option where you are most in control is to keep you server as the only MX server in the DNS. And only forward certain aliases onto Google after all your servers checks. Other aliases and user can just use your mail server if you prefer. I will explain how to do this in the next steps.

DNS

You only put your mail server as the mx for the domain in question. Google will complain about this, as it will not be able to verify that email is setup correctly. Ignore this as it will still accept emails.

MySQL tables

You setup vou aliases as normal. However vou domain table needs tweakina. This is because otherwise your

[base 32bit Ubuntu 8.04 Hardy Heron AMI image](#). You can cheat by using my other images, but you should really know how the whole server was built by starting from the bottom.

When using EC2 images, be aware of security groups as they restricts access to your server on top of the firewall. Initially you will need SSH (22) access, quite soon you will need SMTP and IMAP ports opened, 25,143,465,587 and 993, and eventually webserver ports of 80 and 443. [Read here](#) for tips on securing AMIs.

Also do not terminate your instances without backing up your machine. This you can do by either create your own image. Or backup certain data if you got an image to instantiate from. Back up to S3 or your local machine. Create images only now and then. Backup configurations, database, maildirs more regularly.

Note: You probably want to remove my ssh key from root's authorized_keys2 file.

2nd Note: [Spamhaus.org](#) lists amazons ec2 ip ranges as dynamic, thus many mail servers will reject emails from it. (Including other people using this howto.) But Spamhaus has a simple web page to remove ips, which they link to in rejection messages. Simple look in your logs, click on the link on follow the instructions: basically fill in your ip, email and state its for a mail server. Then Spamhaus will remove your IP from their database.

3rd Note: [This fix needs to applied to the instances.](#)

Amazon EC2 Images: AMIs

Public AMIs to use as base:

AMI	Description	S3 Name	Extended from
ami-ce44a1a7	Eric Hammond's base Ubuntu 8.04 Hardy Heron		
ami-0f41a466	Clean with packages but no configuration	flurdy-amis/ubuntu-mail-server-clean-080502-1	ami-ce44a1a7 (Eric Hammond's base)
ami-8541a4ec	Just mysql, postfix and courier configured	flurdy-amis/ubuntu-mail-server-simple-080504-1	ami-0f41a466 (Clean)
ami-9941a4f0	Including anti spam and anit virus	flurdy-amis/ubuntu-mail-server-spam-080504-1	ami-8541a4ec (Simple)
ami-395fba50	Including TLS and SASL encryption and authentication	flurdy-amis/ubuntu-mail-server-secure-080527-2	ami-9941a4f0 (Spam)
ami-275fba4e	With webmail and admin enabled	flurdy-amis/ubuntu-mail-server-webmail-080527-1	ami-395fba50 (Secure)
ami-xxx	With back up mx	flurdy-amis/ubuntu-mail-server-backup-xxx	ami-275fba4e (Webmail)
ami-xxx	With back up mx only	flurdy-amis/ubuntu-mail-server-backup-only-xxx	ami-395fba50 (Secure)

EC2 Links

- [Amazon web services \(AWS\)](#)
- [Elastic Computing Cloud \(EC2\)](#)
- [Simple Storage Service \(S3\)](#)
- [AWS Cost Calculator](#)
- [EC2 Resource Centre](#)
- [EC2 Starter Guide](#)
- [EC2 Firefox extension: Elasticfox](#)
- [Elasticfox for Firefox 3](#)
- [S3 Firefox extension: S3Fox](#)
- [EC2 to S3 Admin Scripts](#)

[Return to top.](#)

Appendix

- [About author](#)
- [Contact](#)
- [Why](#)
- [References](#)
- [Software Links](#)
- [Difference between Ubuntu versions](#)
- [Download](#)
- [Todo](#)



- [Hildebrand's book](#)
- [Hildebrand's website](#)
- [List-Petersen](#)
- [Genco Yilmaz](#)
- [Christop Haas](#)
- [Nenzel & Peet](#)
- [Peters](#)
- [Matthews](#)
- [Stepanov](#)
- [Andy "BesV"](#)
- [Meta Consultancy](#)

New references

- [Postfix TLS](#)
- [Postfix main.cf doc](#)
- [saslauthd](#)
- [Bypassing amavisd](#)
- [Ubuntu Help: Squirrelmail](#)

Todo

- Populate some of the: Refer to previous edition...
- Spell check!
- Pad with better text. Copy some across from last edition.
- Check bookmark links
- remove uid and guid
- Create backup mx AMI

Please refer to the [previous edition](#) for some old todos....

Software Links

Please refer to the [previous edition](#).

Difference between Ubuntu versions

Please refer to the [previous edition](#).

Download

Please refer to the [previous edition](#).

Change log

Please refer to the [previous edition](#).

[Return to top.](#)



This work is licensed under a [Creative Commons Attribution-ShareAlike 2.5 License](#).

Flurdy