

# Email-Server: Postfix - Ein Email-System für Linux

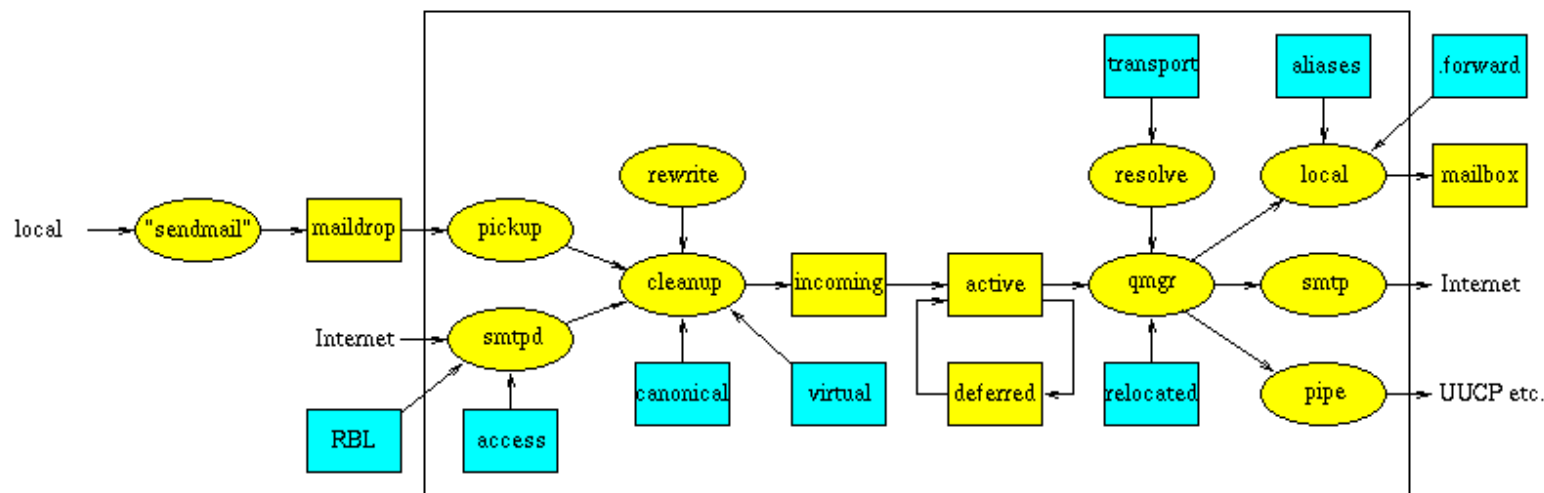
Im dritten und letzten Teil des [Email-Servers](#) [1] erfährst du wie die Emails von deinem Email-Server versendet werden. Dazu dient uns Postfix.

## Postfix

Postfix ist ein sogenannter MTA, ein **Mail Transfer Agent**, er kümmert sich um den Versand und Empfang von Email. Er nimmt die Emails von den Nutzern im Netzwerk an, verteilt sie im eigenen Netzwerk oder übergibt sie an einen anderen MTA. Desweiteren kann er auch Emails außerhalb des eigenen Netzes annehmen und weiterleiten. Er ist quasi der Postbote; ein Tux mit einem Rucksack, der durch das Netz flitzt.

Postfix ist (wird) von Wietse Venema geschrieben worden und steht unter einer von der Mozilla Public License abgeleiteten Lizenz von IBM. Richtig, IBM, dort arbeitet Wietse und bei IBM hat Postfix einen anderen Namen "IBM Secure Mailer".

Postfix ist ein sehr sicheres und ziemlich einfach zu administrierendes Programm. Postfix ist modular aufgebaut und besteht aus vielen kleinen Programmen, hier mal eine kleine Übersicht (Quelle: Postfix-Doku) :



## Erklärung:

- gelbe Ellipsen = Programme
- gelbe Kästchen = Mail-Queues
- grüne Kästchen = Tabellen / Dateien

In der obigen Übersicht erkennst du, welchen Weg die Emails durch die einzelnen Postfix-Programme nehmen. Ich denke mal das reicht als grober Überblick, weiteres findest du in der Doku auf der [Postfix-Homepage](#) [2].

---

## Installation von Postfix

Die Installation kann einfach oder schwer werden, es hängt davon ab ob und welcher MTA (sendmail, smail, qmail, exim, ...) vorher auf deinem System lief. Am besten einfach Postfix von deiner Distribution installieren, sollte überall dabei sein (bei Mandrake ist es der Standard-MTA). Bei Debian solltest du nicht den exim deinstallieren (sonst wird noch weitere Software deinstalliert !!), sondern einfach Postfix installieren. Dann wird auch exim entfernt.

Wenn du es aus dem Internet im Quelltext laden willst, solltest du vorher alles was mit `sendmail` zu tun hat umbenennen ( d.h. sendmail, newaliases, mailq) und dann kompilieren und installieren (der üblich Dreisatz: `./configure, make, make install`). Aber wie gesagt, es kann beim selberkompilieren sehr verschiedene Probleme geben und deshalb beschreibe ich es nicht, kann dir aber bei Fragen evtl. weiterhelfen.

Wenn dein Provider dir anbietet oder dich zwingt **SMTP mit Authentifizierung (SMTP Auth)** zu verwenden, solltest du das tun. Die Vorgehensweis hierzu findest du im Tuxhausen-Artikel [SMTP AUTH mit Postfix](#) [3].

---

## Konfiguration von Postfix

### main.cf

Die Datei main.cf (meist unter /etc/postfix zu finden) ist die Hauptkonfigurations-Datei von Postfix. In ihr stehen alle wichtigen Parameter. Die Datei erklärt auch direkt (in englisch) was der Parameter bewirkt, und wie er voreingestellt ist. Ich werde hier nicht alle Parameter erklären, sondern nur die wichtigen die wir für unseren Server brauchen!

Für dieses Beispiel hab ich Postfix in der Version 1.1.11 unter Debian Woody (3.0) benutzt.

- **queue\_directory, command\_directory, daemon\_directory**  
dienen zum Einstellen der Arbeitsverzeichnisse und können meist so stehen bleiben.
- **mail\_owner**  
legt den Besitzer der Queues und der meisten Postfix-Programme fest. Auf keinen Fall root verwenden !! Besser einen Benutzer anlegen der nur für Postfix zuständig ist und sich nicht einloggen kann, Empfehlung: `mail_owner = postfix`
- **myhostname**  
legt fest welchen Namen der Mailserver haben soll. Bei uns also:  
`myhostname=benton.example.net`  
Gibst du hier keinen Namen an, probiert Postfix den Namen mit Hilfe von `gethostname()` selber herauszufinden.  
Ein Tipp: Der Name muß nicht der richtige Name des Rechners sein, solange der Name per DNS aufgelöst werden kann.
- **mydomain**  
legt den Domain-Namen fest. Also für uns: `mydomain=example.net`
- **mydestination**  
legt fest wofür der Server zuständig ist; d.h. für welche Domains die Emails angenommen werden. Günstig wäre unser Hostnamen, unsere Domain und noch unser localhost, die Einstellung dafür ist :  
`mydestination= $myhostname, $mydomain, localhost.$mydomain`  
Das \$ vor `$myhostname` besagt, das der Namen benutzt werden soll der in `myhostname` steht. Die Variable `$myhostname` hat also den Wert `benton.example.net`  
Diese Vorgehensweise verhindert Schreibfehler und dient zum schnellen Ändern mehrerer Parameter.
- **alias\_maps**  
legt fest welche Alias-Datenbank verwendet werden soll und welches Format sie hat. Siehe nächsten Abschnitt  
Der richtige Eintrag für uns ist : `alias_maps = hash:/etc/aliases`
- **mynetworks**  
An diesem Punkt trägst du die Netzwerke ein, die Postfix als vertrauenswürdig einstufen soll.  
Also für uns: `mynetworks = 127.0.0.0/8, 192.168.44.0/24`  
Die 127. Adresse stellt den localhost, also den Emailserver selber dar.
- **relayhost**  
legt fest wohin Postfix die Emails die er selber nicht ausliefern kann (also alles außerhalb der Domain), senden soll. Wir wollen sie zu dem Mailserver unseres Providers senden, und zwar auf Port 25 (Standart SMTP-Port). Wir geben also an: `relayhost = smtp.puretec.de`  
Diese Einstellung hängt natürlich davon wie dein Provider es genau managt, es könnte sein daß du dich zuerst via POP3 anmelden muß.

Damit wollen die Provider Spam verhindern. Verlangt dein Provider SMTP mit Authentifizierung (SMTP Auth), solltest du mal in Tuxhausen-Artikel [SMTP AUTH mit Postfix](#) [3] gucken.  
Genauerer bitte bei deinem Provider erfragen!

Wäre unser Server fest mit dem Internet verbunden, wären wir so gut wie fertig (allerdings wäre dieser Server dann extrem unsicher ); aber unser Server wählt sich ja nur bei Bedarf ein und kann nur dann die Emails abschicken.

Also müssen wir noch einige Dinge einstellen:

- `defer_transports = smtp`

verhindert das der Server selbständig versucht die Emails zu verschicken. Wir wollen dem Server nach der Einwahl explizit die Anweisung zum Email versenden geben. Mit dem Befehl `postfix flush` verschickt Postfix die Email, dieser Befehl gehört also in dein Einwahlsript. Leider ist dieses Script sehr Distributionsabhängig, aber unter `/etc/ppp/ip-up` findest du ein Script das sich um die PPP-Anmeldung kümmert. Dort trägst du diesen Befehl irgendwo unten ein, dann sollte der Server nach der Einwahl die Emails abliefern. Ich persönlich habe mir ein Script für diesen Zweck geschrieben (für Debian potato).

Leider gibt es ein kleines Problem: Wenn du online bist und eine Email schreibst, versendet Postfix diese nicht, da er ja nur explizit aufgefodert ausliefern soll!

Tja, ich habe die Funktion wieder herausgenommen, aber im Postfix-Manual wird extra darauf hingewiesen. Weitere Hinweise zum Betrieb von Postfix an einem Einwahl-Rechner findest du [hier](#) [4] (eng.) (Ich danke Ralf Rascher für diesen Tip).

Ist der Server allerdings nicht online und bekommt eine Email, so wird diese gespeichert und später versendet.

Das war es auch schon, alle anderen Parameter sind für unseren Zweck nicht direkt wichtig. Solltest du den Server aber in einer größeren Umgebung einsetzen die permanent mit dem Internet verbunden ist, solltest du noch einige Sicherheitsvorkehrungen gegen Spam einbauen. Alle weiteren Parameter sind in der Datei und in der Doku beschrieben.

Zuerst testen wir mal ob alles soweit in Ordnung ist:

```
[root]# postfix check
```

Achtung !! postfix check findet keinerlei Schreibfehler !!

Zum Überprüfen deiner Parameter folgendes eingeben:

```
[root]# postconf
```

Hier sollten jetzt alle Parameter die du in die main.cf geschrieben hast auftauchen.

Jetzt kannst du deinen Postfix mal starten:

```
[root]# postfix start
```

Gibt es Fehler in deiner Konfigurationsdatei oder falsche Rechte, weist dich Postfix nun darauf hin, ansonsten ist alles glatt gegangen. Dieses Start-Kommando mußt du natürlich noch in deine Startscripte aufnehmen, sonst hast du nach dem nächsten rebbot keinen Postfix mehr. Das brauchst du aber nur zu machen, wenn du Postfix aus den Quellen kompilierst hast.

Beendet wird Postfix so:

```
[root]# postfix stop
```

Nach jeder Änderung mußt du Postfix sagen, daß er die Konfiguration neu einlesen soll, und zwar so:

```
[root]# postfix reload
```

---

## Die Alias-Datenbank

Mit dieser Datenbank lassen sich lokale Adressen umschreiben und umleiten.

Wir brauchen diese Datenbank um einen virtuellen Email-Namen einem realen Benutzer-Namen zuordnen, also alle Emails an B.Beimer@example.net gehen an den lokalen Benutzer bert.

Verwendest du für die lokalen Benutzer-Namen den Email-Namen, brauchst du die Alias eigentlich nicht, aber guck trotzdem mal rein.

Die Eintragungen finden in der Datei **aliases** statt, diese ist meistens im /etc Verzeichnis zu finden.

Achtung, sollte diese Datei noch von sendmail stammen, könnte es damit evtl. Probleme geben!

Ein Eintrag in der Alias-Datenbank sieht bei Postfix so aus:

```
"B.Beimer": bert
"A.Allen": angela

root: bert
postmaster: root

test: tux@web.de
```

Die erste Zeile bedeutet, alle Emails an B.Beimer(@example.net) werden an bert(@example.net) weitergeleitet. Das gleiche machen wir mit Angela.

Danach legen wir fest, daß alle Emails die an root gehen, zu bert geleitet werden. Danach gehen alle Emails von postmaster an root , aber wir eben festgelegt das diese an bert gehen; also kriegt bert alle Emails die an postmaster gehen.

### Als kleinen Zusatz

zu den Fähigkeiten von Postfix steht in der letzten Zeile: Alle Emails die nach test@example.net gehen, werden an tux@web.de weitergeleitet.

Wichtig ist noch: Kommt ein Punkt im Namen vor, benutze die " " Zeichen (siehe letzte Zeile).

Ich denk mal, so schwer war das nicht, oder?

Nach jeder Änderung in aliases mußt du Postfix sagen, daß er die Datenbank neu erzeugen soll (keine Angst, die aliases wird dabei nicht überschrieben) und das geht so:

```
[root]# postalias /etc/aliases
```

## Mailing-Listen

Mit der Alias Datenbank kannst du auch prima kleine Email-Verteiler-Listen schreiben. Nehmen wir mal an du willst allen lokalen Benutzern eine Mitteilung schreiben, wäre es nicht prima wenn du eine Email an alle@example.net schreiben würdest, und alle bekommen sie?

Kein großes Problem mit Postfix:

Zuerst eine Text-Datei ( /etc/postfix/lists/alle ) mit der Empfänger-Liste anlegen.

Hier die Liste:

```
#Interne Mailingliste von example.net
#DAS LETZTE KOMMA IST WICHTIG!!!
angela,
bert,
```

Und in die Alias-Datenbank kommt dieser Eintrag:

```
#Mailinglisten
alle: :include:/etc/postfix/lists/alle
```

Dann noch schnell die Alias-Datenbank updaten und das war es schon, schreibst du nun eine Email an alle@example.net, bekommen angela und bert diese Email.

Das Ganze geht natürlich auch mit Adressen die nicht innerhalb deiner Domain liegen. Der [Tuxhausen-Newsletter](#) [5] funktioniert so. Deine Textdatei sieht dann in etwas so aus:

```
#Tuxhausen Newsletter
#DAS LETZTE KOMMA IST WICHTIG

mu@web.de,
tux@linux.org,
"billy.regal@ikea.de",
```

## Tipps zu Postfix

- Auch Postfix hat ein `sendmail`, es ist aber nicht das Original `sendmail` gemeint, sondern das "Postfix-sendmail-Ersatz-Programm", damit bleiben die wichtigsten alten `sendmail` Befehle verfügbar, nur dass sie jetzt von Postfix ausgeführt werden.
- Mit dem Befehl `mailq` werden alle Emails angezeigt die noch nicht ausgeliefert wurden. Im Grunde versteckt sich hinter `mailq` der Befehl `sendmail -bp`
- Mit `postfix flush` fordest du Postfix auf, alle wartenden Emails jetzt zu verschicken. Sehr nützlich wenn die Einwahl ins Internet erfolgt ist
- Alle noch nicht abgeschickten Emails lagern unter `/var/spool/postfix/deferred/`. Löscht du eine Email aus dem Verzeichnis, wird sie nicht mehr versendet. Nützlich um eine fehlerhafte Email noch schnell abzufangen.
- Hast du eine echte Domain (wie `tuxhausen.de`) und verwendest du diese Domain auch für dein Intranet, möchtest aber eine Email an einen externen Kollegen (`extern@tuxhausen.dee`) senden, der Kollege aber keinen Zugang zu deinem Postfix-Server hat, hast du ein kleines Problem. Schickst du die Email an unseren Server nimmt er sie an und leitet sie weiter, aber Lokal !!  
Postfix ist ja der Mailer der Domain (`mydestination= $myhostname, $mydomain, ...`), und so bleibt die Email in deinem Netz !  
Der einzige Weg den ich kenne: `$mydomain` aus `mydestination` löschen und alle internen Emails an `@benton.example.net` senden.
- Wenn du eine Email aus der Warteschlange löschen willst mußt du zuerst die Queue ID ermitteln und diese dann löschen:

```
[root]# mailq                                #Wartende Emails anzeigen
-Queue ID- --Size-- ----Arrival Time---- -Sender/Recipient-----
9064B165C4* 582 Tue Aug 20 22:11:11 m.ungermann@tuxhausen.de
                                         mu@elzet80.de
```

```
-- 0 Kbytes in 1 Request.  
  
[root]# postsuper -d 9064B165C4          #Wartende Email löschen  
postsuper: 9064B165C4: removed  
postsuper: Deleted: 1 message
```

---

## Quellen und weitere Dokus

Der deutsche Mirror der [Postfix-Homepage](#) [2]  
Außerdem gibt es das [Red Hat Postfix HowTo](#) [6] (englisch).  
Im Linux-Magazin 06/2000 erschien ein Artikel über Postfix: [Modular und sicher](#) [7]

Über Postfix gibt ein recht gutes deutschsprachiges Buch: "Das Postfix Buch" von Peer Heinlein, erschienen bei SuSE Press (ist aber nicht nur für SuSE geeignet), ISBN 3-935922-41-8  
Und natürlich die üblichen man Pages....

---

## Im Text genannte Links

- [1] Tuxhausen - Email-Server: <http://www.tuxhausen.de/email-server.html>
- [2] Postfix Homepage: <http://www.informatik.uni-bonn.de/pub/software/postfix/>
- [3] SMTP AUTH mit Postfix [http://www.tuxhausen.de/postfix\\_smtp\\_auth.html](http://www.tuxhausen.de/postfix_smtp_auth.html)
- [4] Hinweise Postfix mit Dial-Up-Rechner (eng.): <http://www-dt.e-technik.uni-dortmund.de/~ma/postfix/dialup.html>
- [5] Tuxhausen-Newsletter <http://www.tuxhausen.de/newsletter.html>
- [6] Red Hat Postfix HowTo: <http://www.redhat.com/support/resources/howto/RH-postfix-HOWTO/book1.html>
- [7] Artikel Linux-Magazin: <http://www.linux-magazin.de/ausgabe/2000/06/Postfix/postfix.html>

---

Erstellt: 2001-01-28

Autor: Markus Ungermann [Kontakt](#)

Letztes Update: 2005-07-11

URL: [http://www.tuxhausen.de/software\\_postfix.html](http://www.tuxhausen.de/software_postfix.html)

---

