

# Debian Wheezy, DBmail 3, PostgreSQL und Postfix

[4. Februar 2014](#)

Einführung

DBmail ist ein Mailspeicher, der komplett auf Datenbanken basiert.

- Es erlaubt das Anlegen von Mailboxen ohne das Vorhandensein von entsprechenden Systembenutzern.
- Nachrichten werden effizienter gespeichert und können somit schneller abgerufen und eingefügt werden als bei anderen Systemen. (DBmail ist derzeit in der Lage ca. 250 Nachrichten pro Sekunde abzurufen.)
- Es ist skalierbar. Man kann mehrere DBmail Programme auf verschiedenen Servern betreiben die auf die gleiche Datenbank(-Cluster) zugreifen
- Es ist sicherer. Man muss keine Systembenutzer pflegen oder in das Dateisystem eingreifen. Das alles wird über die Datenbank abgewickelt.

Quelle: DBmail Readme (Übersetzt aus dem Englischen vom Autor)

Was sind die Vorteile von DBmail?

- Skalierbarkeit

Die Skalierbarkeit von Dbmail wird nur vom verwendeten Datenbanksystem beschränkt

- Management

Dbmail wird gemanagt, indem man die relationale Datenbank oder den Verzeichnisdienst anpasst – ohne Shell Zugriff.

- Geschwindigkeit

Dbmail verwendet sehr effiziente, Datenbank-spezifische Anfragen für den Empfang von Mailinformationen

- Sicherheit

Dbmail benötigt keinen Zugriff auf das Dateisystem. Es ist so sicher wie die Datenbank oder der Verzeichnisserver der verwendet wird.

- Flexibilität

Veränderungen im Dbmail System (z.B. Hinzufügen von Benutzern, Änderungen von Passwörtern, etc) werden praktisch sofort wirksam. Benutzer können sowohl in der Datenbank als auch in einem LDAP Server, wie zum Beispiel OpenLDAP oder Active Directory, hinterlegt werden.

Quelle: dbmail.org (Übersetzt aus dem Englischen vom Autor)

Wie funktioniert es ?

Copyright bleibt bei Teris Cooper und kann jederzeit über [www.root-projekte.de](http://www.root-projekte.de) verädert werden.



Dbmail besteht aus verschiedenen Modulen. Ein Standard MTA (Postfix, SendMail, QMail, Exim) wird zum Empfangen von Nachrichten verwendet.

Der MTA übergibt die Nachricht entweder an dbmail-deliver, unter Verwendung einer Pipe oder an dbmail-lmtpd unter Verwendung des LMTP (Local Mail Transport Protocol). Diese Programme sorgen dafür, dass die Nachrichten zur Datenbank hinzugefügt werden.

Nachrichten können von der Datenbank per POP3 Protokoll unter Verwendung oder per IMAP4Rev1 Protokoll abgerufen werden.

Emails werden in der Datenbank inklusive Anhängen gespeichert. Benutzerinformationen werden auch in der Datenbank hinterlegt, so dass kein Benutzerkonto auf der Maschine benötigt wird, auf der Dbmail betrieben wird.

Quelle: dbmail.org, Übersetzt aus dem Englischen vom Autor

Weitere Infos findet man unter dbmail.org.

### **Vorbereitung**

Der erste Schritt ist sich das passende Paket von dbmail.org in der aktuellsten Version herunterzuladen und zu entpacken:

#### **Quellcode**

1. `wget http://dbmail.org/download/3.0/dbmail-3.0.2.tar.gz && tar -zxvf dbmail-3.0.2.tar.gz`

Alle Dateien stehen jetzt im Verzeichnis /dbmail-3.0.2

Im Verzeichnis (vorher hineinwechseln) schauen wir uns das README Dokument an. Welchen Editor man dazu wählt ist eigentlich egal. Da wir nichts editieren möchten, sondern die Datei nur lesen möchten benutzen wir `less README`

Am Wichtigsten für uns ist der letzte Satz, dass alle Installationshinweise in der Datei `INSTALL` zu finden sind. Diese sollte man direkt öffnen und sich mit den Informationen vertraut machen `less INSTALL`

Da wir jetzt wissen, was zu tun ist, geht es gleich weiter.

Aus der `INSTALL` erfahren wir auch, dass wir `libzdb` brauchen, welches wir aber manuell herunterladen und installieren müssen:

#### **Quellcode**

1. `wget http://www.tildeslash.com/libzdb/dist/libzdb-2.10.2.tar.gz && tar -zxvf libzdb-2.10.2.tar.gz`
2. `cd libzdb-2.10.2`
3. `./configure`
4. `make && make install`

Copyright bleibt bei Teris Cooper und kann jederzeit über [www.root-projekte.de](http://www.root-projekte.de) verändert werden.



## Installation und Vorbereitung des PostgrSQL-Servers

Selbstverständlich brauchen wir für unser dbmail einen passenden Datenbankserver. Wir verwenden einen PostgreSQL. Ein MySQL ist selbstverständlich auch möglich. Der Postgresql Server wird wie gewohnt über apt-get installiert.

### Quellcode

1. apt-get install postgresql

Nach der Installation ist nur der Linux-Benutzer postgres berechtigt, die Datenbank zu starten. Eine Anmeldung als User postgres kann auf jeden Fall auf dem Umweg über die Anmeldung als root erfolgen. Auch wenn dieser kein gültiges Passwort besitzt oder aus Sicherheitsgründen keines besitzen soll.

### Quellcode

1. su - postgres

Wir legen einen neuen Benutzer namens dbmail an:

### Quellcode

1. createuser -U postgres dbmail

Soll die neue Rolle ein Superuser sein? (j/n) n

Soll die neue Rolle Datenbanken erzeugen dürfen? (j/n) n

Soll die neue Rolle weitere neue Rollen erzeugen dürfen? (j/n) n

Postgres Konsole aufrufen via Befehl:

### Quellcode

1. psql

Sie können die Datenbankkommandozeile mit Strg-D (ctrl-D) oder q wieder verlassen. Aber zunächst soll eine Datenbank angelegt werden.

### Quellcode

1. createdb -E utf8 -U postgres -O dbmail dbmail

Passwort setzen für user dbmail:

### Quellcode

1. ALTER USER dbmail WITH PASSWORD '<password>';

Für <password> bitte das entsprechende Passwort einsetzen, welches Sie benutzen möchten!

Damit ist die Einrichtung des PostgreSQL abgeschlossen.

Installation + Konfiguration von DBMail

Copyright bleibt bei Teris Cooper und kann jederzeit über [www.root-projekte.de](http://www.root-projekte.de) verädert werden.



Wir benutzen eine Wheezy Distribution und müssen noch einige Pakete installieren.

### Quellcode

1. apt-get install gcc libgmime-2.4-dev
2. libglib2.0-dev libmhash-dev libevent-dev libssl-dev flex
3. postgresql-server-dev-8.4 make automake

Zunächst wechseln wir wieder in das richtige Verzeichnis:

### Quellcode

1. cd dbmail-3.0.2

Danach folgt:

### Quellcode

1. ./configure --with-zdb=/usr/local
- 2.
3. make && make install

Wir legen uns nun einen Testbenutzer an, der sich später testweise mit der Datenbank verbinden können soll und damit ein Postfach zur Verfügung hat.

Starten der Datenbankkommandozeile:

### Quellcode

1. psql

### Quellcode

1. dbmail-users -a <testbenutzer > -w <testpasswort>

Konfiguration von postfix für die Benutzung mit Dbmail

Weiterführende Informationen: [dbmail.org/dokuwiki/doku.php/setup\\_postfix?s\[\]=postfix](http://dbmail.org/dokuwiki/doku.php/setup_postfix?s[]=postfix)

**Zunächst die passenden Pakete installieren:**

### Quellcode

1. apt-get install postfix postfix-pgsql
2. cat /etc/postfix/main.cf

folgende Einträge müssen geändert bzw. hinzugefügt werden:

myhostname = dbmail1.openlab.de

mydestination = localhost.openlab.de, , localhost

virtual\_transport = dbmail-lmtp:localhost:24

virtual\_mailbox\_domains = pgsql:/etc/postfix/sql-virtual\_mailbox\_domains.cf

Copyright bleibt bei Teris Cooper und kann jederzeit über [www.root-projekte.de](http://www.root-projekte.de) verädert werden.



```
virtual_mailbox_maps = pgsqldb:/etc/postfix/sql-virtual_mailbox_maps.cf
Jetzt müssen noch die beiden folgenden Dateien entsprechend angepasst werden:
- /etc/postfix/sql-virtual_mailbox_domains.cf
```

In dieser Datei müssen folgende Einträge gemacht werden:

```
user = dbmail
password = <password>
hosts = localhost
dbname = dbmail
query = SELECT DISTINCT 1 FROM dbmail_aliases WHERE SUBSTRING(alias FROM
POSITION('@' in alias)+1) = '%s';
- /etc/postfix/sql-virtual_mailbox_maps.cf
```

## Quellcode

1. nano /etc/postfix/sql-virtual\_mailbox\_maps.cf

```
user = dbmail
password = <password >
hosts = localhost
dbname = dbmail
query = SELECT 1 FROM dbmail_aliases WHERE alias='%s';
```

## Test

Für den Test benötigt man einen Mailclient. Mit dem Passwort <testpasswort> für den Benutzer <testbenutzer> können wir uns mit dem Rechner verbinden. In diesem Howto ist nur die unverschlüsselte Verbindung beschrieben.

Nachdem wir die richtigen Benutzerdaten eingeben haben, können wir die Verbindung zum DBmail testen.

