



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Cómo montar un potente sistema de correo con postfix (154257 lecturas)

Per **Javi Polo**, [DrSlump](http://drslump.org) (<http://drslump.org>)

Creado el 12/12/2002 17:56 modificado el 16/12/2002 10:58

En este artículo intentaré explicar como montar un sistema de correo con [postfix](#)⁽¹⁾ que contenga usuarios, dominios y demás datos en una base de datos [mysql](#)⁽²⁾ y que a su vez soporte tanto autenticación para envío (usando [cyrus-sasl](#)⁽³⁾) como integrar [courier-imap](#)⁽⁴⁾ y [courier-pop](#)⁽⁵⁾ para que los usuarios puedan bajar su correo.

Así mismo, podremos usar un entorno de correo web como [squirrelmail](#)⁽⁶⁾ para que los usuarios puedan manejar su correo via web.

Por si fuera poco, la administración de las cuentas y demás se hace via web, así que, ¿a que esperas para probarlo? :)

Este documento explica como instalar y configurar un servidor de correo *completo*. Esto es:

- **postfix** para SMTP
- **courier** para IMAP y POP3
- **cyrus-sasl** para autenticación SMTP
- **squirrelmail** para webmail
- **mysql**, para guardar los usuarios

El objetivo de montar todo el chiringuito este es tener todos los datos que solemos modificar a menudo en nuestro servidor de correo en una BBDD, de forma que podamos añadir nuevos dominios de correo, usuarios, redirecciones, etc. de forma bastante simple, sin tener que ir a meter mano a los ficheros de configuración de *postfix*.

Además del salto que supone en el tema de seguridad, al no tener usuarios a piñón en nuestro sistema, esto permite una mayor flexibilidad a la hora de manejar estos usuarios, dar altas, bajas, realizar consultas, ...

Una vez que acabemos de configurar nuestro servidor, tendremos un sistema que nos permitirá:

- Recibir correo para nuestros usuarios y dominios
- Autenticar por SASL para que los usuarios puedan usar nuestro sistema como relayhost
- Repartir el correo de cada usuario, ya sea via POP o IMAP
- Que los usuarios lean/escriban correo a través de un entorno de webmail
- Administrar todo el tinglado desde una página web, donde añadiremos o quitaremos dominios, usuarios y lo que haga falta

Antes que nada aclarar todo está pensado de base para hacerlo en debian sid (la distribución de las abuelitas TM), por lo que si usas otra distro pues algunas cosas cambiarán, o sea, que te tendrás que buscar un poco la vida. Yo soy muy vago, así que no me complico mucho y uso debian :)

Este artículo está bastante basado en la documentación que he encontrado por ahí sobre como hacer tirar postfix+courier contra mysql. Es una muy buena idea leerse los howtos originales y la documentación ...

Iremos paso a paso para entender bien que vamos haciendo :)

Ante todo, aclarar que **NO TENGO PUTA IDEA de SQL, ni de PHP**, y que todo lo que se lo he aprendido con esto :)



1. La base de datos

Primero de todo instalaremos *mysql* y crearemos una base de datos que contendrá la información que queremos.

La instalación de *mysql* no tiene mucha complicación en debian:

```
apt-get install mysql-server mysql-client
```

respondemos a las 2 preguntillas que nos hace, que no importa que las explique, y ale, ya tenemos *mysql* instalado (como amo debian).

Ahora habrá que crear una base de datos y un usuario que pueda usarla ... :

Aquí usaremos **mail** como base de datos, y el usuario será **mail**, con password **cipriano**. Además, le damos permisos para hacer de todo en la base de datos **mail**.

Ejecutamos el cliente de *mysql*:

```
chiberu:~# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 3.23.53-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database mail;
Query OK, 1 row affected (0.00 sec)

mysql> grant all privileges on mail.* to mail@localhost identified by 'cipriano';
Query OK, 0 rows affected (0.00 sec)

y seleccionamos la base de datos:

mysql> use mail;
Database changed
```

Una vez tenemos la base de datos creada, meteremos las tablas que usaremos.

Las tablas que yo uso son:

Tabla transport

```
create table transport (
  domain varchar(128) not null primary key unique,
  transport varchar(128) not null default 'virtual:'
);
```

Esta tabla se usará para transport en el *postfix* (se puede hacer en un fichero, pero ya que tenemos una BBDD, aprovechemosla ;)).

Dicha tabla sirve para indicar que hacer con el correo para X dominio, de forma que podemos decirle, por ejemplo, el correo para el dominio *arale.org* me lo reenvía a la máquina *mail.arale.org* y el del dominio *niko-chan.net* me lo repartes entre los usuarios virtuales de la máquina.

La intención de este artículo no es la de explicar el funcionamiento de un servidor de correo, así que si no sabes lo que es, mejor **RTFM**: *man transport*.

Tabla virtual

```
create table virtual (
  address varchar(255) not null primary key unique,
  goto varchar(255) not null
```



```
);
```

Esta será una tabla de usuarios virtuales, o redirecciones, o como queráis llamarlo. Es la típica virtusertable de sendmail, que permite reenviar el correo dirigido a *rey@niko-chan.net* a *sembei@drslut.org*, por ejemplo.

Al igual que antes, **RTFMi**: *man virtual*.

Tabla users

```
create table users (
  address varchar(255) not null primary key unique,
  id varchar(128) not null,
  clear varchar(128) not null,
  uid smallint(5) unsigned not null default 10000,
  gid smallint(5) unsigned not null default 8,
  name varchar(128) not null,
  home varchar(128) not null default '/var/mail',
  domain varchar(128) not null,
  maildir varchar(255) not null,
  active tinyint(3) unsigned not null default 1
);
```

Esta tabla es la que tiene la chicha de los usuarios :

Campo	Descripción	Ejemplo
address	Dirección de correo completa	<i>javipolo@chiberu.drslump.org</i>
id	Usuario a secas (lo de la izquierda de la arroba)	<i>javipolo</i>
clear	Password en claro	<i>faryownz</i>
uid	UID que tendrán sus ficheros en el sistema	<i>10000</i>
gid	GID que tendrán sus ficheros en el sistema	<i>8</i>
name	Nombre del fulano en cuestion	<i>Javi Polo</i>
home	Directorio 'base' para almacenar el mail	<i>/var/mail</i>
domain	Dominio (la derecha de la arroba)	<i>chiberu.drslump.org</i>
maildir	Mailbox (o maildir en nuestro caso) a usar, path relativo a home	<i>chiberu.drslump.org/javipolo/</i>
active	Con esta flag controlamos si el usuario puede o no bajarse el correo	<i>1</i>

Como veis, no tiene mucha complicación el tema hasta aquí. Simplemente nos hemos limitado a crear las tablas en las que luego vamos a ir insertando los datos que usará nuestro sistema de correo.

Notese que en el campo *maildir* he acabado de indicar donde guardar el mail con una */*. Esto es así porque queremos que use maildir y no mailbox, ya que courier no soporta mailbox.

Una vez que tengamos esto listo, demos otro paso adelante.

2. Postfix

El siguiente paso es instalar *postfix* con soporte de SASL y de *mysql*. En debian son los paquetes *postfix-tls* y *postfix-mysql*:

```
apt-get install postfix-tls postfix-mysql
```

Si compilas tú mismo *postfix*, acuerdate de que tienes que darle soporte de SASL usando la *libsasl2*.

Antes de meter mano a la configuración, limpiemos el */etc/postfix* de mierda que nos deja la instalación por defecto:

```
chiberu:/etc/postfix# rm access canonical pcre_table regexp_table relocated transport virtual
```

y ahora, vamos a meterle mano al fichero **main.cf**:



añadimos esto antes de *mydestination*:

```
virtual_mailbox_base=/var/mail
virtual_mailbox_maps=mysql:/etc/postfix/mysql_virt.cf
virtual_uid_maps=mysql:/etc/postfix/uids.cf
virtual_gid_maps=mysql:/etc/postfix/gids.cf
transport_maps=mysql:/etc/postfix/transport.cf
virtual_maps =mysql:/etc/postfix/virtual.cf
```

y *mydestination* lo dejamos parecido a esto:

```
mydestination = chiberu, localhost.localdomain, , localhost, $transport_maps
```

Y ahora vamos a indicarle donde buscar las cosas en la BBDD en los diferentes ficheros que hemos dicho a *postfix*:

/etc/postfix/mysql_virt.cf

```
user=mail
password=cipriano
dbname=mail
table=users
hosts=localhost
where_field=address
select_field=maildir
```

/etc/postfix/uids.cf

```
user=mail
password=cipriano
dbname=mail
table=users
hosts=localhost
where_field=address
select_field=uid
```

/etc/postfix/gids.cf

```
user=mail
password=cipriano
dbname=mail
table=users
hosts=localhost
where_field=address
select_field=gid
```

/etc/postfix/transport.cf

```
user=mail
password=cipriano
dbname=mail
table=transport
hosts=localhost
select_field=transport
where_field=domain
```

/etc/postfix/virtual.cf

```
user=mail
password=cipriano
dbname=mail
table=virtual
hosts=localhost
where_field=address
select_field=goto
```



Tampoco tiene misterio ... le decimos que conecte a *\$hosts* con usuario *\$user*, password *\$password* y use *\$dbname*, y allí haga un *select \$select_field from \$table where \$where_field ...* amos, por ejemplo en caso del *uids.cf*, si se recibiera un mail para *javipolo@chiberu.drslump.org*, *postfix* haría esta consulta:

```
select uid from users where address='javipolo@chiberu.drslump.org'
```

IMPORTANTE: Asegurate de que estos ficheros solo los puede leer el grupo *postfix* puesto que si no cualquiera podría trincarnos el password de acceso a la BBDD, y no es plan ...:

```
for i in mysql_virt.cf uids.cf gids.cf virtual.cf ; do chgrp postfix $i; chmod 640 $i; done
```

Ahora se nos plantea un problema ... *postfix* corre como *chroot* en */var/spool/postfix*. Para acceder a la BBDD lo hace a través de un socket (por defecto */var/run/mysqld/mysqld.sock*) y al estar este fuera del entorno *chroot* en el cual se ejecuta *postfix* (por defecto */var/spool/postfix*), no tenemos acceso a dicho socket.

Una posible solución es sacar del *chroot* a *postfix*, editando el fichero */etc/postfix/master.cf*, yo personalmente **PASO**, así que he optado por la otra posible solución.

Esta otra solución consiste en crear el directorio dentro del *chroot*:

```
mkdir -p /var/spool/postfix/var/run/mysqld
chown mysql /var/spool/postfix/var/run/mysqld
```

y crear un hard link al socket actual:

```
ln /var/run/mysqld/mysqld.sock /var/spool/postfix/var/run/mysqld/mysqld.sock
```

Ojo, esto solo funcionará si */var/spool/postfix/var/run/mysqld/* y */var/run/mysqld/* están en la misma partición. En caso contrario, bueno, es trivial, así que piensa un poquillo ;)

Bien, tenemos *postfix* funcionando guay tirando contra *mysql* ... next step ...

3. Courier IMAP y POP

Empezamos con la consabida instalación:

```
apt-get install courier-imap courier-pop courier-authmysql
```

y ale, vamos a */etc/courier* y a configurar ;)

primero en *authdaemonrc* hay que decirle:

```
authmodulelist="authmysql"
```

y luego hay que tocar */etc/courier/authmysqlrc*:

```
MYSQL_SERVER          localhost
MYSQL_USERNAME        mail
MYSQL_PASSWORD        cipriano
MYSQL_SOCKET          /var/run/mysqld/mysqld.sock
MYSQL_OPT              0
MYSQL_DATABASE        mail
MYSQL_USER_TABLE      users
MYSQL_CLEAR_PWFIELD   clear
MYSQL_UID_FIELD       uid
MYSQL_GID_FIELD       gid
MYSQL_LOGIN_FIELD     address
MYSQL_HOME_FIELD      home
MYSQL_NAME_FIELD      name
MYSQL_MAILDIR_FIELD   maildir
```



Creo no importa comentar nada, el fichero está bastante comentado y además es bastante obvio para que sirva cada cosa ..

Ale, ya está casi todo listo ... tenemos una base de datos donde metemos los usuarios, un servidor smtp que recibe correos para esos usuarios, y un servidor pop y otro imap que despachan el correo para esos usuarios ...

4. Autenticación por SMTP

Queremos que nuestros usuarios puedan enviar a través de nuestro sistema, y discriminar por IP puede resultar un auténtico coñazo, y el *pop before smtp*, personalmente, no me gusta mucho, así que ... usemos *cyrus-sasl* para añadir a nuestro *postfix* un mecanismo de autenticación.

Lo adivináis, ¿no? si si, instalaaaaaar:

```
apt-get install libsasl2-modules
```

En realidad, la librería *sasl2* (con *sasl1* esto no chusca. Hay un parche por ahí para añadir el auxprop de mysql, pero no viene de casa) la hemos tenido que instalar antes a la hora de instalar *postfix-tls* .. (los que lo hayan compilado ellos mismos, espero que hayan pensado en esto, o si no, a recompilar O:)).

Una vez instalado este paquete, tenemos que decirle a *cyrus-sasl* que tiene que usar una BBDD *mysql*, y esto se le dice en **/etc/postfix/sasl/smtpd.conf**:

```
mysql_user: mail
mysql_passwd: cipriano
mysql_hostnames: localhost
mysql_database: mail
mysql_statement: select clear from users where id='%u' and domain='%r'
pwcheck_method: auxprop
```

Al igual que con los ficheros que creamos para *postfix*, pongámosle unos permisos decentes para que no nos afanen el password.

Ahora le decimos a *postfix* que vamos a usar *SASL*, y que deje pasar los mails de los que se hayan autenticado por *SASL*. En **/etc/postfix/main.cf**:

```
smtpd_sasl_auth_enable = yes
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions =
    permit_sasl_authenticated
    permit_mynetworks
    reject_unauth_destination
```

Llegados a este punto, ya podríamos crear cuentas en la base de datos y utilizarlas tranquilamente.

5. Correo por web

Bueno, el sistema de correo está funcionando de forma bastante guapa ya, así que ahora vamos a apijotarlo un poco ;)

Como gestor de correo por web vamos a usar *squirrelmail*, que es muy completo.

Este ... que iba ahora? ah si ...:

```
apt-get install squirrelmail
```

Si no los tenemos instalados, se nos instalarán *php4* y *apache*.

Podemos ya cambiar lo que queramos de la configuración *squirrelmail* ejecutando **/etc/squirrelmail/conf.pl**, aunque la configuración que viene por defecto ya funciona bien.



Ahora tenemos que configurar *apache* en `/etc/apache/httpd.conf` para que cargue el modulo de *php4*:

```
LoadModule php4_module /usr/lib/apache/1.3/libphp4.so
```

Tambien añadiremos a la linea *DirectoryIndex* un *index.php*:

```
DirectoryIndex index.php index.html index.htm index.shtml index.cgi
```

Y finalmente, vamos a meter un alias para acceder al *squirrelmail*

```
Alias /webmail/ /usr/share/squirrelmail/
```

Ya tenemos todo listo, reiniciamos apache con

```
killall -l apache
```

y a probar nuestro chiringo :)

6. Manejo de la BBDD

Como anexo, aqui va [un .tgz](#)⁽⁷⁾ con unas paginas en php para administrar la base de datos. La verdad es que me da MUCHA vergüenza que se vea este código, cuando lo veais sabreis porque, pero bueno, ahí está. Necesitaremos de *php4-mysql* para que funcionen :)

Como digo al principio, **NO TENGO NI PUTA IDEA de php ni de html ni de nada**, así que el código es una mierda. Supongo que poco a poco lo iré mejorando, y además acepto sugerencias/parches/whatever :)

Para instalar esto es todo bastante simple, copiamos los .php donde queramos (visible por web) y añadimos una nueva tabla a la BBDD llamada **admins**, que contendrá las cuentas de los administradores que podrán manejar todo el chiringuito. Luego editamos el config.php *to suit our needs* y tira millas.

En caso de meter usuario@dominio.tal de usuario, con su correspondiente password, se le permitirá a dicho usuario cambiarse la contraseña.

7. Cosas que faltan

Pues bueno, creo que todo está dicho, pero aun faltan cosas por hacer. Una breve lista de lo que yo veo que le falta al tema (ya me lo miraré, pero con calma que ando liado ahora :() podría ser esta:

1. Implementar cuotas
2. Que los passwords vayan cifrados en la BBDD. Para ello hace falta modificar el auxprop para SASL ya que solo soporta passwords en claro. A ver si saco un rato y lo apaño
3. Que se creen automaticamente los maildirs al crear las cuentas, y que, si así se indica, se borren a la hora de eliminarlas
4. Inserta aqui lo que creas que falta ;)

En fin, nada es perfecto, así que si se te ocurren ideas para mejorar esto, no dudes en contarmelas

Lista de enlaces de este artículo:

1. <http://www.postfix.org>
 2. <http://www.mysql.com>
 3. <http://asg.web.cmu.edu/sasl/>
 4. <http://www.inter7.com/courierimap/>
 5. <http://www.courier-mta.org/>
 6. <http://www.squirrelmail.org>
 7. <http://bulma.net/~javipolo/files/mailadmin.tgz>
-

BULMA: Cómo montar un potente sistema de correo con postfix



E-mail del autor: javipolo _ARROBA_ drslump.org

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1621>