

Preparing an SMTP/IMAP/HTTP Server for mail with Raid partitions under Geom

Before configuring this server, a fresh FreeBSD OS install must be done on the first disk.

1. Preparing the RAID-1 partition

First, we must reset few bytes of the secondary disk to ensure that this disk is a fresh dis. Warning: You must replace /dev/ad6 by the device name of the second disk (eg: /dev/ad1 or other).

```
# dd if=/dev/zero of=/dev/ad6 bs=512 count=79
79+0 records in
79+0 records out
40448 bytes transferred in 0.011471 secs (3526099 bytes/sec)
```

Now we must put a GEOM label on the second disk (generally the last block of the disk) and activate it

```
# gmirror label -v -n -b round-robbin gm0 /dev/ad6
Metadata value stored on /dev/ad6.
Done.
# gmirror load
GEOM_MIRROR: Device gm0 created (id=2906068201).
GEOM_MIRROR: Device gm0: provider ad6 detected.
GEOM_MIRROR: Device gm0: provider ad6 activated.
GEOM_MIRROR: Device gm0: provider mirror/gm0 launched.
```

After that, we can be able to create a PC MBR on this second disk with a disklabel

```
# fdisk -v -B -I /dev/mirror/gm0
*****Working on device /dev/mirror/gm0 *****
Information from DOS bootblock is:
1: sysid 165 (0xa5), (FreeBSD/NetBSD/386BSD)
   start 63, size 312576642 (152625 Meg), flag 80 (active)
   beg: cyl 0/ head 1/ sector 1;
   end: cyl 0/ head 254/ sector 63
2: <UNUSED>
3: <UNUSED>
4: <UNUSED>
# bsdlabel -w -B /dev/mirror/gm0s1
```

/dev/mirror/gm0s1 is now created correctly, the last step is to create new partitions on it for /usr, /var, /tmp etc...

```
# bsdlabel -e /dev/mirror/gm0s1
```

This command launch the default editor specified by the EDITOR environment variable. The default editor is vi but you can set another one with export EDITOR=

or setenv EDITOR depending of your shell. All partitions must be created with correct offset and size for mounting on the filesystem (/var, /usr, /tmp etc...).

WARNING: the 'a' partition MUST begin with offset 16, don't modify that !

Partition editing example under vi:

```
# /dev/mirror/gm0s1:
8 partitions:
#          size  offset  fstype  [fsize bsize bps/cpg]
a:    4194304     16  4.2BSD     0     0     0
b:    4109696 4194320  4.2BSD     0     0     0
c:   312576642      0  unused     0     0     0
d:    8388608 8304016  4.2BSD     0     0     0
e:    4194304 16692624  4.2BSD     0     0     0
f:   291689714 20886928  4.2BSD     0     0     0
~
~
~
~
```

We need to format these new partitions and copy all partitions data from the first disk to the second disk. Newfs and dump/restore can do that simply. You must apply these commands for all partitions (eg: '/dev/mirror/gm0s1d', '/dev/mirror/gm0s1e' partitions etc...).

```
# newfs -U /dev/mirror/gm0s1a
/dev/mirror/gm0s1a: 2048.0 MB (4194304 sectors) block size 16384, fragment size 2048
  using 12 cylinder groups of 183.77MB, 11761 blks, 23552 inodes.
  with soft updates
super-blocks backups (for fsck -b #) at:
160, 376512, ...
..., 3763680, 4140032
# mount /dev/mirror/gm0s1a /mnt
# dump -L -0 -f- / | (cd /mnt; restore -r -v -f-)
...
Check the symbol table.
Check pointing the restore
DUMP: DUMP IS DONE
```

Output dumpis not copied for the next commands because the output must be approximatively the same :-)

```
# newfs -U /dev/mirror/gm0s1d
# mount /dev/mirror/gm0s1d /mnt/var
# dump -L -0 -f- /var | (cd /mnt/var; restore -r -v -f-)
# newfs -U /dev/mirror/gm0s1e
# mount /dev/mirror/gm0s1e /mnt/tmp
# dump -L -0 -f- /tmp | (cd /mnt/tmp; restore -r -v -f-)
# newfs -U /dev/mirror/gm0s1f
# mount /dev/mirror/gm0s1f /mnt/usr
# dump -L -0 -f- /usr | (cd /mnt/usr; restore -r -v -f-)
```

Adjust new configuration for GEOM based setup. This include /mnt/etc/fstab, /mnt/etc/rc.conf and /mnt/boot/loader.conf. Do not forget to replace ad4 by the device name of your first disk and ad6 by the device name of your second disk.

```
# cp -p /mnt/etc/fstab /mnt/etc/fstab.orig
# sed -e 's/dev\ad4/dev\mirror\gm0/g' </mnt/etc/fstab.orig >/mnt/etc/fstab
# echo 'swapoff="YES"' >>/mnt/etc/rc.conf
# echo 'geom_mirror_load="YES"' >>/mnt/boot/loader.conf
# echo "1:ad(6,a)/boot/loader" > /boot.config
```

Now we must reboot the server for booting on the prepared GEOM second disk. This is needed because the next step is to flush the first disk and replicate the second disk on it.

```
# reboot
or
# shutdown -r now
```

It's the time to reset few bytes on the first and original data disk. Other commands is for synchronizing the second disk on the first disk and going to RAID-1 configuration with GEOM gmirror. Replace /dev/ad4 by the device name of your fist disk.

```
# dd if=/dev/zero of=/dev/ad4 bs=512 count=79
79+0 record in
79+0 records out
40448 bytes transferred in 0.010466 secs (3864668 bytes/sec)
# gmirror configure -a gm0
# gmirror insert gm0 /dev/ad4
GEOM_MIRROR: Device gm0: provider ad4 detected
GEOM_MIRROR: Device gm0: rebuilding provider ad4.
```

The disk /dev/ad6 will be synchronizing on /dev/ad4 and /dev/ad4 will be added on the /dev/mirror/gm0. So now, we just wait the end of the synchronization, you can check it with 'gmirror list' and watch the DD led ;-). After that, we can reboot the system and the two disk are completely working under a RAID-1 configuration

```
! WARNING ! Doing that only if the second disk have finished to synchronize. Check it with this command. If State of the second disk is ACTIVE then you can reboot the system:
# gmirror list
...
Name: ad4
...
State: ACTIVE(or SYNCHRONIZING if this is not complete)

# reboot
or
# shutdown -r now
```

2. Installing MySQL server with InnoDB table handler

It's the time to install a database for Postfix and Cyrus (see later), the choice is MySQL server 4.1 (more stable) with transactional and fine grained locking InnoDB handler. The simple way is to use the FreeBSD ports subsystem like this:

```
# cd /usr/ports/database/mysql41-server
# export BUILD_OPTIMIZED=yes
or depending the shell
# setenv BUILD_OPTIMIZED yes
# export BUILD_STATIC=yes
or depending the shell
# setenv BUILD_STATIC yes
# make install clean
```

You may use the following build options:

```
...
=> Extracting for mysql-server-4.1.11_1
...
Compilation and Installtion may take a while
...
=> Cleaning for mysql-client-4.1.11_1
=> Cleaning for libtool-1.5.10_1
=> Cleaning for mysql-server-4.1.11_1
```

We must now configure mysql server. This step is done with an utility installed by mysql that create default tables on /var/db/mysql and editing a configuration file for the database

```
# mysql_install_db
...
# chown -R mysql:mysql /var/db/mysql
# cp /usr/local/share/mysql/my-medium.cnf /var/db/mysql/my.cnf
# vi /var/db/mysql/my.cnf
```

We just uncomment innodb_ parameters lines. All these parameters are sufficient for a postfix/cyrus server tables. Now it's the time to add mysql to startup scripts and launch mysql server.

```
# echo 'mysql_enable="YES"' >> /etc/rc.conf
# /usr/local/etc/rc.d/mysql-server.sh start
Starting mysql.
```

Just a little test to see if the mysql server respond correctly. You can check /var/db/mysql/<hostname>.err file if there are some troubles.

```
# mysql -u root
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 4.1.11-log

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

mysql> show databases;
+-----+
| Database |
+-----+
| mysql   |
| test    |
+-----+
2 rows in set (0.00 sec)

mysql> exit
Bye
```

3. Installing Apache / PHP

You need an Apache Web server and PHP for using Web cyradm interface. After installing php4-pear, there is some modules required by webcyradm. These modules are php4-gettext, php4-session and php4-mysql for accessing database and save login sessions.

Apache 1.3 at first:

```
# cd /usr/ports/web/apache13-modssl
# make install clean

You may use the following build options:
...
====> Cleaning for libtool-1.5.10_1
====> Cleaning for mm-1.3.1
====> Cleaning for perl-5.8.6_2
====> Cleaning for expat-1.95.8_1
====> Cleaning for apache+mod_ssl-1.3.33+2.8.22
```

And then, php4 pear:

```
# cd /usr/ports/devel/php4-pear
# make install clean
...
====> Cleaning for expat-1.95.8_1
====> Cleaning for php4-xml-4.3.11
====> Cleaning for apache-1.3.33_1
====> Cleaning for php4-pear-4.3.11_2
```

Now we can copy the default php.ini file recommended and add the include path of pear packages:

```
# cp /usr/local/etc/php.ini-recommended /usr/local/etc/php.ini
# touch /var/log/php.log
# chown www: /var/log/php.log
```

Now edit /usr/local/etc/php.ini and change/add these values:

```
error_log = /var/log/php.log
include_path = ".:usr/local/share/pear"
```

Install other necessary packages for PHP WebCyradm:

```
# cd /usr/ports/devel/php4-gettext
# make install clean
...
====> Cleaning for help2man- 1.35.1
====> Cleaning for expat- 1.95.8_1
====> Cleaning for apache- 1.3.33_1
====> Cleaning for php4- gettext-4.3.11
```

```
# cd /usr/ports/www/php4-session
# make install clean
...
====> Cleaning for help2man- 1.35.1
====> Cleaning for expat- 1.95.8_1
====> Cleaning for apache- 1.3.33_1
====> Cleaning for php4- session-4.3.11
```

```
# cd /usr/ports/databases/php4-mysql
# make install clean
...
====> Cleaning for help2man- 1.35.1
====> Cleaning for expat- 1.95.8_1
====> Cleaning for apache- 1.3.33_1
====> Cleaning for php4- mysql-4.3.11
```

You must configure some parameter in Apache default configuration for using it as a simple web server with one virtual host.

Edit the configuration file /usr/local/etc/apache/httpd.conf and change/add `ServerName` by your server hostname

```
ServerName mx0.bousol.net
```

Add PHP Application type to the configuration file:

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

For running correctly, Apache need that `ServerName` entry is a valid resolved name, if you don't have DNS on /etc/resolv.conf or name doesn't really exist on the DNS referred by this file, replace `mx0.bousol.net` by an IP address or add this on your /etc/hosts file:

```
Replace 192.168.1.219 by the server IP address and mx0.boussol.net mx0 by the ServerName entry
192.168.1.219                mx0.bousol.net mx0
```

Just add a line to /etc/rc.conf for running Apache/PHP automatically

```
# echo 'apache_enable="YES"' >> /etc/rc.conf
If you don't need SSL, add apache_flags too
# echo 'apache_flags=""' >> /etc/rc.conf
# /usr/local/etc/rc.d/apache.sh start
Starting apache.
```

If Apache run perfectly (default page on a web browser), you can delete the default data link.

```
# rm -f /usr/local/www/data
```

4. Installing WebCyrAdm PHP interface for Cyrus

WebCyrAdm doesn't exist on the ports tree, so you must download it form:
<http://www.web-cyradm.org/>

Then, de-tar it on /usr/local/www/data

```
# tar xvfz web-cyradm-0.5.4-1.tar.gz -C /usr/local/www
# mv /usr/local/www/web-cyradm-0.5.4-1 /usr/local/www/data
# cp /usr/local/www/data/config/conf.php.dist /usr/local/www/data/config/conf.php
```

Some parameters must be configured in /usr/local/www/data/config/conf.php, edit this file and add/change these parameters:

```
'PASS' => '<password of the mail user>',
$LOG_DIR = "/var/log/";
$CRYPT = "plain";
$DOMAIN_AS_PREFIX = 1;
```

Create the log file with good rights:

```
# touch /var/log/web-cyradm.log
# chown www: /var/log/web-cyradm.log
```

5. Configuring WebCyrAdm PHP interface for Cyrus

Now we need to create the database and tables for Postfix and Web-cyradm and add a user to the database. Web-cyradm comes with several MySQL scripts: insertuser_mysql.sql and create_mysql.sql.

The first inserts the Database user to the database »mysql« and creates the database »mail«. The second creates the required tables and populates the database with an initial admin-user and the cyrus user. The other scripts are used for incremental upgrading from older releases.

The password for the database user »mail« in this example is »secret«. Please insert whatever user and password you like. The username for the initial superuser is »admin« with the password »test«. Change the default password! If a malicious user wants to gain unauthorized access to a system, the first attempt is always the default username and password supplied by the vendor. It is **IMPORTANT** that you change them in the scripts before applying them.

Edit the script /usr/local/www/data/scripts/insertuser_mysql.sql and remove ENCRYPT() keyword from the password. Plain text must be used for the moment... With cyrus db solution, it's the same problem, if you strings the db, you can fetch user passwords of imap mailboxes. After customizing the username and password, apply the scripts:

```
# mysql -u root -p < /usr/local/www/data/scripts/insertuser_mysql.sql
Enter password:      <- Use the root password defined for mysql or Enter
# mysql mail -u mail -p < /usr/local/www/data/scripts/create_mysql.sql
Enter password:      <- the password of the mail user
```

6. Installing Postfix with MySQL support

Postfix is already on the FreeBSD ports. When launching installation there is a ncurses menu that appears with some options to select. For our use, these options must be selected:

```
# cd /usr/ports/mail/postfix
# export WITH_MYSQL=YES
or depending the shell
# setenv WITH_MYSQL YES
# make install clean
...
[X] SASL2    Cyrus SASLv2 (Simple Authentication and Security Layer)
...
[X] TLS      SSL and TLS
...
[X] MySQL    MySQL map lookups (choose with WITH_MYSQL_VER)

Now, go to OK with tabular and press enter, the compilation must start...

You need user 'postfix' added to group 'mail'
Would you like me to add it [y]? y
...
Would you like to activate Postfix in /etc/mail/mailer.conf [n]? n
...
=> Cleaning for mysql-client-4.1.11_1
=> Cleaning for libtool-1.5.10_1
=> Cleaning for pcre-5.0
=> Cleaning for cyrus-sasl-2.1.20_1
=> Cleaning for postfix-2.2.2,1
```

The next step consist to edit the /usr/local/etc/postfix/main.cf file to configure or

change some parameters.

First change the hostname and the default domain name and the origin:

```
Change mx0.bousol.net by the hostname of the server or the MX FQDN  
myhostname = mx0.bousol.net  
mydomain = bousol.net  
myorigin = $mydomain
```

Now, you can configure mydestination parameter that specify all domain names that are considered as local (for delivery). In this case, we use the mysql database, but you can set some domain name manually before the mysql lookup. The mysql-mydestination.cf file contains all parameters necessary to lookup the correct mysql host, database and table. We see how to create this file on the next chapter.

```
mydestination = mysql:/usr/local/etc/postfix/mysql-mydestination.cf
```

The next parameter to uncomment from the default main.cf is mailbox_transport. This is used to specify the delivery program. In our case, it's cyrus !

```
mailbox_transport = cyrus
```

We need to tell to postfix to use sasl2:

```
smtpd_sasl_auth_enable = yes  
smtpd_recipient_restrictions = reject_invalid_hostname, reject_non_fqdn_hostname,  
reject_non_fqdn_sender, reject_non_fqdn_recipient, reject_unknown_sender_domain,  
reject_unknown_recipient_domain, reject_unauth_pipelining,  
permit_sasl_authenticated, permit_mynetworks, reject_unauth_destination,  
reject_rbl_client zombie.dnsbl.sorbs.net, reject_rbl_client relays.ordb.org,  
reject_rbl_client opm.blitzed.org, reject_rbl_client list.dsbl.org,  
reject_rbl_client sbl.spamhaus.org, permit  
smtpd_sasl_security_options = noanonymous  
smtpd_sasl_local_domain =  
broken_sasl_auth_clients = yes
```

At the end of the configuration, you must add some parameters for looking other mysql table like virtual and canonical:

```
virtual_alias_maps = mysql:/usr/local/etc/postfix/mysql-virtual.cf  
sender_canonical_maps = mysql:/usr/local/etc/postfix/mysql-canonical.cf
```

Normally, mysql ports had installed sasl2 ports with a dependancy.

7. Creating maps file for Postfix

Now, we need to create the mysql maps configuration files. Here, /usr/local/etc/postfix/mysql-virtual.cf:

```
hosts = localhost
user = mail
password = <your secret password for mail>
dbname = mail
table = virtual
select_field = dest
where_field = alias
additional_conditions = and status = '1'
```

/usr/local/etc/postfix/mysql-canonical.cf file:

```
hosts = localhost
user = mail
password = <your secret password for mail>
dbname = mail
table = virtual
select_field = alias
where_field = username
additional_conditions = and status = '1' limit 1
```

Finally, /usr/local/etc/postfix/mysql-mydestination.cf file:

```
hosts = localhost
user = mail
password = <your secret password for mail>
dbname = mail
table = domain
select_field = domain_name
where_field = domain_name
```

8. Installing Cyrus

Cyrus is on the ports:

```
# cd /usr/ports/mail/cyrus-imapd22
# make install clean
Leave default options and Press OK
...
====> Cleaning for makedepend- 2000.12.28
====> Cleaning for perl-5.8.6_2
====> Cleaning for cyrus-sasl-2.1.20_1
====> Cleaning for cyrus-imapd- 2.2.12
```

Now, edit the /usr/local/etc/imapd.conf and uncomment or change the following parameters:

```
postmaster: postmaster
allowanonymouslogin: no
allowplaintext: yes
sasl_mech_list: PLAIN
Replace mx0.bousol.net by your server hostname
servername: mx0.bousol.net
Replace 10000 by the amount of desired quota
autocreatequota: 10000
reject8bit: no
quotawarn: 90
timeout: 30
poptimeout: 10
dracinterval: 0
drachost: localhost
sasl_pwcheck_method: auxprop
sendmail: /usr/sbin/sendmail
sievemaxscriptsize: 32
sieve_maxscripts: 5
unixhierarchysep: yes
```

If You want to use TLS/SSL certificates for using imap ssl then add these lines and follow the certificate generation instruction. Else skip this step !

```
tls_ca_file: /var/imap/server.pem
tls_cert_file: /var/imap/server.pem
tls_key_file: /var/imap/server.pem
```

Now we can create a TLS/SSL certificate for person that want check mail with an encrypted tunnel:

```
# openssl req -new -nodes -out req.pem -keyout key.pem
Generating a 1024 bit RSA private key
.+++++++
.....+++++++
writing new private key to 'key.pem'
Country Name (2 letter code) [AU]:CH
State or Province Name (full name) [Some-State]:Geneve
Locality Name (eg, city) []:Geneve
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Bousol
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:mx0.bousol.net
Email Address []:support@bousol.net
A challenge password []:
An optional company name []:
# openssl rsa -in key.pem -out new.key.pem
writing RSAkey
# openssl x509 -in req.pem -out ca-cert -req -signkey new.key.pem -days 999
Signature ok
subject=/C=CH/ST=Geneve/L=Geneve/O=Bousol/CN=mx0.bousol.net/emailAddress=support@bousol.net
Getting Private key
# mkdir /var/imap
# cp new.key.pem /var/imap/server.pem
# rm new.key.pem
# cat ca-cert >> /var/imap/server.pem
# chown cyrus:mail /var/imap/server.pem
# chmod 600 /var/imap/server.pem
```

TLS/SSL is finished, continue the configuration...

Don't forget to run `/usr/local/cyrus/bin/mkimap` for configuring correct permissions and files on `/var/imap`:

```
# /usr/local/cyrus/bin/mkimap
reading configure file...
i will configure directory /var/imap.
i saw partition /var/spool/imap.
done
configuring /var/imap...
creating /var/spool/imap...
done
```

You can modify some parameters on `/usr/local/etc/cyrus.conf` but the default file is already good for us.

The last step is to add `imapd` to `rc.conf` for running automatically and launch it:

```
# echo 'cyrus_imapd_enable="YES"' >> /etc/rc.conf
# /usr/local/etc/rc.d/imapd.sh start
```

9. Installing Horde/IMP

Horde/IMP is a webmail PHP based software for accessing to your mail account from an HTTP/HTTPS interface.

Horde/IMP is in the FreeBSD ports, to install it do:

```
# cd /usr/ports/mail/imp
# make install clean
```

Press CTRL-C and define WITHOUT_IMAPSERVER
if you intend to run an IMAP server on an other machine.

...
Select PEAR_NET_POP3 option and Press OK

...
Select PEAR_AUTH_SASL and Press OK

...

Horde has been installed in /usr/local/www/horde with your blank
configuration files.

If you want Horde to access a database, you have to run the
appropriate scripts located in /usr/local/www/horde/scripts/sql.
It is recommended that you change the password of the 'hordemgr'
user used to connect to the horde database.
Horde is setup by default to access MySQL.

WARNING! if you are upgrading from Horde v. 2.2.x, you have to alter your
*****database schemas. Please read the doc UPGRADING.

You can now access Horde without a password at <http://localhost/horde/>,
and you will be logged in as an administrator. You should first configure
a real authentication backend. Click on "Setup" in the "Administration"
menu and configure Horde. Start in the "Authentication" tab.

See the doc in /usr/local/share/doc/horde for details.

(tip: if you plan to install IMP, just keep "Automatic authentication as
a certain user", add your login to be treated as administrator, and once
IMP will be installed, switch to "Let a Horde application handle auth").

Select a log driver; if you keep 'file', do not forget to add a line
in /etc/newsyslog.conf.

Then select and configure a preferences driver.

Your /usr/local/etc/apache/httpd.conf has been updated,
you have to restart Apache.

When everything is OK, you should be able to access Horde from
<http://localhost/horde/>.

(If <http://localhost/horde/> does not run, but
<http://localhost/horde/index.php> is OK, then you have
to define index.php as a DirectoryIndex in /usr/local/etc/apache/httpd.conf.)

There is a testing script at <http://localhost/horde/test.php>.

After that, go to <http://mx0.bousol.net/horde> for configuring horde and other
components like imp.

You can define a specific VirtualHost on the Apache configuration for going
directly to imp like this <http://webmail.bousol.net/>