Cyrus the Great (559 - 530 BC), founder of the Persian kingdom, created the first modern postal system, whose motto was: "stopped by neither snow, rain, heat, or gloom of night ".

CYRUS

PostgreSQL fpm Web-Cyradm

POWERED BY
redhat. LINUX

Installation Howto

By Andrew Koros

Postfix+Cyrus+PostgreSQL+Web-cyradm+RedHat Howto

Prepared by Andrew Koros <andkoros@yahoo.com>
Last updated on 23rd May 2003 for web-cyradm-0.5.3-1 and postfix-2.0.8 with SMTP AUTH instructions

Table of Contents

Copyright & Disclaimer

Credits

- Thanks to Simon Matter *<simon.matter@invoca.ch >* for great cyrus-imapd rpms and for tips on how to configure saslauthd and lmtp.
- Thanks to Simon J Mudd <sjmudd@pobox.com > for the most customizable postfix rpms anywhere.
- Luc de Louw <luc@delouw.ch> for the great web-cyradm tool that makes cyrus-imapd look deceptively simple to configure.
- Thanks to David Nyakundi *<dnyambinya@yahoo.com>* and Steven Neill *<steve@wayfarersrest.com>* for giving good feedback on the document.

Introduction

Web-Cyradm was intended as a tool for managing multiple virtual users and domains on a cyrus-imapd server. This it does by creating virtual users that don't exist in the unix system but only in the imap/pop server's system. This document describes how to achieve this with postgresql and postfix on a RedHat Linux System.

There are two ways that web-cyradm supports virtual users. One is of the type "domain0xx" where x is a number, and the other is of the type "username.domain.tld". The second type is only possible if the option "`unixhierarchysep: yes`" is set in the `/etc/imapd.conf.` I like the second scheme because the account names are easier to remember .Unfortunately, for cyrus-imapd versions before 2.1.12 the second scheme doesn't seem to work with sieve(the server-side mail filtering tool included with cyrus-imapd). So if you intend to use sieve you may be better placed with the former or make sure you are using cyrus-imapd-2.1.12 or newer. However you can always map the IMAP/POP usernames to a better scheme using the *Perdition* IMAP/POP proxy server (see the Additional Ideas Section at the end of the document). A full-featured, built-in virtual domain support is scheduled to be in the version 2.2.x series of cyrus-imapd.

Since the users don't exist in the UNIX account, any requests to the imap server are first authenticated by pam which uses two modules, the pam_unix and the pam_pgsql. It checks the pam_unix first which off course doesn't find the account (because its virtual) then it tries pam_pgsql which finds the account in the postgresql database.

This howto will give you virtual accounts of the type "user.domain.example" which means the usernames for logging into the imap/pop server will be of the type "user.domain.example" but the respective addresses will be "user@domain.example"

The web-cyradm in this howto will create cyrus mailboxes of the form "user/username.domain.example" and therefore when a user creates his IMAP folders they will appear internally to cyrus-imap as:

```
"user/username.domain.example"
"user/username.domain.example/Drafts"
"user/username.domain.example/Sent"
"user/username.domain.example/Trash"
```

where "`user/username.domain.example`" will be the "INBOX".

Prepare your Linux Server

If you are just experimenting on postfix + cyrus-imapd or you are new to tweaking linux filesystems then you may just use the default RedHat file system settings and ignore this page (Or put in geek terms, this section is NOT for "newbies", OK?).

      If you intend to run postfix + cyrus-imapd on a production server with a substantial number of users who receive lots of mail per day You are advised to make a few considerations and tweaks:

1. The IMAP service is very I/O intensive because the mail is being manipulated on the server's disks. It is recommended that you use seperate physical disks for `/var/lib/imap` and `/var/spool/imap`. On the other hand if it is not possible then use the fastest disk you have for `/var/spool/imap` and `/var/lib/imap`

2. With ext3 mount `/var/spool/imap` and `/var/lib/imap` with `noatime` option and `data=ordered` mode (the default)

3. The bulk of the mail data goes to `/var/spool/imap` therefore assign disk space according to the number of users you intend to host. (>10GB for 500 users each with a 20 MB quota.)

4. Create a separate partition for `/var/spool/postfix` on a fast disk. The size can be between 256MB and 1000MB or more depending on your maximum possible queue size .

Notes on Postfix+ext3 performance tuning[4]

- mount `/var/spool/postfix` in `data=journal,noatime` (full journal mode and noatime. **IMPORTANT:** *use the latest RedHat errata kernel to avoid known bugs in this mode.* See https://rhn.redhat.com/errata/RHBA-2002-292.html)
- mount `/var/log` in `data=ordered  mode` (the default)
- `chattr -R -S /var/spool/postfix`
  You may optionally add the following to the end of your `/etc/rc.sysinit` file:
- `echo 40 0 0 0 60 300 60 0 0 > /proc/sys/vm/bdflush`
- `/sbin/hdparm -W0 /dev/hda` (To disable write cache)
- `/sbin/elvtune -r 4096 -w 8192 /dev/hdax` (where "x" is the letter containing the `/var/spool/postfix` partition)

So a sample `/etc/fstab` file would look like the one below. (It doesn't have to be _exactly_ like this, this is just an example OK?). It is best that you do the partitions and mount points at install time for simplicity then edit the mount options later

```
### /etc/fstab
LABEL=/                    /              ext3    defaults          1 1
LABEL=/boot                /boot          ext3    defaults          1 2
LABEL=/home                /home          ext3    defaults          1 2
none                       /proc          proc    defaults          0 0
none                       /dev/shm       tmpfs   defaults          0 0
LABEL=/usr                 /usr           ext3    defaults          1 2
LABEL=/var                 /var           ext3    noatime,rw        1 2
LABEL=/var/spool/postfix /var/spool/postfix ext3 noatime,rw,data=journal 1 2
/dev/hda8        swap           swap     defaults       0 0
/dev/cdrom       /mnt/cdrom     iso9660  noauto,owner,user,kudzu,ro 0 0
/dev/fd0         /mnt/floppy    auto     noauto,owner,user,kudzu 0 0
```

Software Requirements

1. RedHat Linux 7.1,7.2,7.3, 8.0 (Tested with 7.3 and 8.0. Please read RedHat 9 Notes at the Appendix)[1]
2. postfix-2.0.8 rpms (http://www.wl0.org/~sjmudd/postfix/en/) sources (http://www.postfix.org)
3. cyrus-imapd-2.1.x rpms (http://home.teleport.ch/simix/), (sources can be found at http://asg.web.cmu.edu/cyrus/imapd/)
4. cyrus-sasl-2.1.x rpms (http://home.teleport.ch/simix/), (sources can be found at http://asg.web.cmu.edu/cyrus/imapd/)
5. mhash-0.8.17 rpms at http://home.teleport.ch/simix/RPMS/Cyrus-imapd/contrib and tarball sources at (http://mhash.sf.net)
6. PostgreSQL-7.2.x rpms (these are include with the RedHat) *Unless you have RedHat 9, please avoid version 7.3.x for now (If you really need the newer features of PostgreSQL-7.3 then please see the Troubleshooting section and the RedHat 9 Notes at the Appendix at the end of this document .)*.
7. pam-pgsql-0.5.2 rpms at http://home.teleport.ch/simix/RPMS/Cyrus-imapd/contrib and tarball sources a (http://sourceforge.net/projects/pam-pgsql)
8. Apache with php-modules (these are include with the RedHat Distribution)
9. php-pgsql (these are include with the RedHat Distribution)
10. web-cyradm-0.5.3-1   (http://www.web-cyradm.org)

Except for the rpms which shipped with RedHat Linux, I built all the rpms from srpm packages as detailed later in the "Installation Procedure" section below.

N.B.

RedHat 8.0 now includes cyrus-sasl-2.x rpms. Please use them if you are using RedHat 8.0 or newer, to avoid unnecessary compiling/rebuilding of srpms. Also always check for the latest RedHat Errata packages from https://rhn.redhat.com/errata/ for your respective RedHat Linux version. *IMPORTANT*:The rpm-build-x.x.x package must be installed in order to build rpms

Building the srpm for cyrus-imapd from http://home.teleport.ch/simix/ will generate perl-cyrus-2.1.x, cyrus-imapd-utils-2.1.x,cyrus-imapd-devel and cyrus-imapd-2.1.x rpms. (Similarly building cyrus-sasl srpm will generate all the cyrus-sasl rpms)

RPM Package List

```
1.mhash-0.8.17-1.i386.rpm
2.mhash-devel-0.8.17-1.i386.rpm
3.postgresql-7.2.3-5.80.i386.rpm
4.postgresql-server-7.2.3-5.80.i386.rpm
5.postgresql-devel-7.2.3-5.80.i386.rpm
6.postgresql-libs-7.2.3-5.80.i386.rpm
7.pam-pgsql-0.5.2-5.i386.rpm
8.cyrus-sasl-devel-2.1.10-1.i386.rpm
9.cyrus-sasl-plain-2.1.10-1.i386.rpm
10.cyrus-sasl-md5-2.1.10-1.i386.rpm
11.cyrus-sasl-2.1.10-1.i386.rpm
12.cyrus-imapd-devel-2.1.12-9.i386.rpm
13.cyrus-imapd-2.1.12-9.i386.rpm
14.cyrus-imapd-utils-2.1.12-9.i386.rpm
15.perl-Cyrus-2.1.12-9.i386.rpm
16.postfix-2.0.8-1.pgsql.sasl2.tls.rh8.i386.rpm
```
17. `apache-1.3.2x.i386.rpm` (RedHat 7.x) or 2.0.4x.i386.rpm (RedHat 8.x)
18. `php-4.x.x-x.x.x.i386.rpm` (tested on 4.1.x and 4.2.x)
19. `php-pgsql-4.x.x-x.x.x.i386.rpm`
20. `php-imap-4.x.x-x.x.x.i386.rpm` (Optional if you want a web based email client like squirrel-mail see http://www.squirrelmail.org/) also included with RedHat 8.0 and newer

---

1   This could be modified to work with any rpm based Linux or FreeBSD OS if you use their repective packaging tools.

Installation Procedure

Except for the postfix rpm, install all the above rpms for your RedHat Linux Distribution. I found it easier to build the rpms from source rpms to avoid dependency conflicts but you will need a number of development packages installed first like gcc,make,glibc-devel autoconf213, e.t.c.

To build from source rpms (srpm) , preferably as a non-root user, prepare your $HOME/rpm directory (as shown for postfix below), then use the following command on RedHat 7.x

```
rpm --rebuild <package-name>-x.x.x-x.src.rpm
```

Or for RedHat 8.0 or newer

```
rpmbuild --rebuild <package-name>-x.x.x-x.src.rpm
```

To install the built rpms do:
```
rpm -Uvh <package-name>-x.x.x-x.i386.rpm
```

You must build postfix from the source rpm because the default postfix does not include the postgresql database lookup patch. Obtained the postfix-2.0.8 source rpm which includes the patch from `http://postfix.wl0.org/ftp/SRPMS/` (Note that version 2.x of postfix is a must if you want SMTP-AUTH support with sasl2 otherwise postfix-1.1.11 or newer should also work). Build it as follows:

As root install the postgresql-devel package and the other postgresql rpms (see the above list) , cyrus-sasl-devel-2.1.x and openssl-devel (optional) for TLS support. I also noticed that the postfix source rpms from http://www.wl0.org/~sjmudd/postfix/en/ require that openldap-devel packages be installed.

Prepare your $HOME/rpm directory:
(Note: the next few steps must be done as a **non-root** user!)
 In you home directory do:

```
mkdir -p $HOME/rpm/RPMS/i386
mkdir $HOME/rpm/SRPMS
mkdir $HOME/rpm/SPECS
mkdir $HOME/rpm/SOURCES
mkdir $HOME/rpm/BUILD
echo "%_topdir    $HOME/rpm" >$HOME/.rpmmacros
```

Install the source rpm for postfix:
```
rpm -ivh postfix-2.0.8-1.src.rpm
cd `rpm --eval '%{_sourcedir}'`
export POSTFIX_PGSQL=1
export POSTFIX_SASL=2 # This is optional but is needed for SMTP-AUTH
                      # must have cyrus-sasl-devel-2.1.x installed
export POSTFIX_TLS=1  # This is optional if you want TLS support you
                      # must have openssl-devel installed
#export POSTFIX_REDHAT_MYSQL=1 # for example if you want to
            # include mysql support (optional) requires mysql-devel
sh make-postfix.spec
cd `rpm --eval '%{_specdir}'`
rpm -ba postfix.spec
```

Or on RedHat 8.0 or newer :
```
rpmbuild -ba postfix.spec
```

Once the build is complete you will find an rpm similar to this here:
```
$HOME/rpm/RPMS/i386/postfix-2.0.8-1.pgsql.sasl2.tls.rhx.i386.rpm
```

You can then install it as root using
```
rpm -Uvh /home/<username>/rpm/RPMS/i386/postfix-2.0.8-1.pgsql.sasl2.tls.rh8.i386.rpm
```


Similarly you can also obtain the mhash library from following link (*which may change so take note*) http://twtelecom.dl.sourceforge.net/sourceforge/mhash/mhash-0.8.17.tar.gz and build the rpm as follows:
```
rpm -ta mhash-0.8.17.tar.gz
```

Or on RedHat 8.0 or newer:
```
rpmbuild -ta mhash-0.8.17.tar.gz
```

If all goes well, you will find two rpms in :
```
$HOME/rpm/RPMS/i386/mhash-0.8.17-1.i386.rpm
$HOME/rpm/RPMS/i386/mhash-devel-0.8.17-1.i386.rpm
```

Install them as root:
```
rpm -Uvh /home/<username>/rpm/RPMS/i386/mhash*i386.rpm
```


Pam_pgsql

There is a srpm for building a pam-pgsql rpm at http://home.teleport.ch/simix/RPMS/Cyrus-imapd/contrib and a binary rpm built on RedHat 8.0. If you are using RedHat 8.0 then just get the rpm and make sure postgresql-libs and mhash rpms are installed, then install it with:

```
rpm -Uvh pam-pgsql-0.5.2-5.i386.rpm
```

You can rebuild the srpm for your version of redhat as usual `rpm --rebuild <package-name>-x.x.x-x.src.rpm` as long as postgresql-devel, pam-devel, mhash and mhash-devel packages are installed. NOTE:You may need to rebuild this rpm as root. But since the pam_pgsql module is just one file it may be easier to just build it from source as explained hereafter.

To install pam_pgsql from source first make sure postgresql-devel, pam-devel, mhash and mhash-devel packages are installed. Then unpack the `pam_pgsql-0.5.2.tar.gz` to a temp directory say `$HOME/work` Then do the following:
```
cd $HOME/work/pam_pgsql-0.5.2
./configure
su
make; make install
```
If the build is successful it should create the file:

```
/lib/security/pam_pgsql.so
```

Please verify that it did before you continue any further.

Configuration Files

PostgreSQL

Make sure that postgresql is running with the "-i" option. This is done by editing the /etc/init.d/postgresql file and adding -o ' -i'   in the following line:

```
su -l postgres -s /bin/sh -c "/usr/bin/pg_ctl  -D $PGDATA -p \
 /usr/bin/postmaster start  > /dev/null 2>&1" < /dev/null
```

When edited should look like this:
```
su -l postgres -s /bin/sh -c "/usr/bin/pg_ctl  -D $PGDATA -o '-i '  -p \
 /usr/bin/postmaster start  > /dev/null 2>&1" < /dev/null
```

Start the `postgresql` server as root :

```
    su
    /sbin/service postgresql start
    /sbin/chkconfig postgresql on
```

The last command ensures that it always starts at boot.

N.B.
Optionally instead of adding the 'i'option in the init script you can achieve the same effect by editing the `/var/lib/pgsql/data/postgresql.conf` file (which is only created after a database initialization by running postgresql at least once) and changing/uncommenting the line below then restart postgresql:

```
...  ...
#       Connection Parameters
#
tcpip_socket = true
#ssl = false
....  ....
```

This is what PostgreSQL rpm packagers recommend.

pg_hba.conf

As root open and edit the file /var/lib/pgsql/data/pg_hba.conf
 and add/edit the following line at the bottom:

```
#/var/lib/pgsql/data/pg_hba.conf
###################################
#
# ....
#
# Put your actual configuration here
# ===================================
#
# This default configuration allows any local user to connect with any
# PostgreSQL username, over either UNIX domain sockets or IP.
#
# If you want to allow non-local connections, you will need to add more
# "host" records. Also, remember IP connections are only enabled if you
# start the postmaster with the -i option.
#
# CAUTION: if you are on a multiple-user machine, the default
# configuration is probably too liberal for you. Change it to use
# something other than "trust" authentication.
#
# TYPE  DATABASE IP_ADDRESS  MASK     AUTH_TYPE  AUTH_ARGUMENT
#local      all                                      trust


# Using sockets credentials for improved security. Not available everywhere,
# but works on Linux, *BSD (and probably some others)


local   all                              ident    sameuser
host    all  127.0.0.1   255.255.255.255     password
```

Restart the `postgresql` server as root :

```
su
/sbin/service postgresql restart
```

Cyrus Sasl Configuration

smtp authentication in chroot

This sub-section section is for those who need SMTP AUTH functionality. What I detail here are the changes I made to saslauthd to make it possible to achieve SMTP AUTH using the PLAIN mechanism while running postfix in it' s chroot jail. It is however, highly recommended that you use TLS when using PLAIN smtp authentication for security reasons. First get smtp authentication working without TLS then read `/etc/postfix/samples/sample-tls.cf` to enable TLS. If you don' t know what SMTP AUTH is then you probably don' t need it at the moment and you may therefore leave this out. Suffice it to say that it' s mostly used to allow mobile users to use your MTA from anywhere without turning it into an open relay. (In that case have a look /etc/postfix/README_FILES/SASL_README and the last two references at the end of this document to satisfy your curiosity) .

As root:
```
su
mkdir -p /var/spool/postfix/var/run/saslauthd
rm -rf /var/run/saslauthd
ln -s /var/spool/postfix/var/run/saslauthd /var/run/saslauthd
```

This will avail the saslauthd socket to postfix in it' s chroot jail.

N.B.
The rest of the instructions will be mentioned as you go along, in similar captions.

 You have two options.
Option 1 (recommended): either just create a file `/etc/sysconfig/saslauthd` as root with something like this:

```
MECH=pam
FLAGS="-n 15"
## Optionally if you want SMTP AUTH functionality you may uncomment the
## next line(below) after following the instructions on SMTP AUTH above.
#SOCKET=/var/spool/postfix/var/run/saslauthd/mux
```

(To know what these parameters mean please have a look at the saslauthd man page: `man saslauthd`)

Then start the saslauthd server as root :

```
su
/sbin/service saslauthd start
/sbin/chkconfig saslauthd on
```

The last command ensures that it always starts at boot.

Or ,

Option 2: as root open and edit the file `/etc/init.d/saslauthd` look particularly for line shown below in red and add/edit it to be as shown:

/etc/init.d/saslauthd

```bash
#! /bin/bash
#
# saslauthd     Start/Stop the SASL authentication daemon.
#
# chkconfig: - 95 05
# description: saslauthd is a server process which handles plaintext
#       authentication requests on behalf of the cyrus-sasl library.
# processname: saslauthd
# Source function library.
. /etc/init.d/functions
# Source our configuration file for these variables.
SOCKET=/var/run/saslauthd/mux
MECH=pam
FLAGS="-n 15"
if [ -f /etc/sysconfig/saslauthd ] ; then
        . /etc/sysconfig/saslauthd
fi
RETVAL=0
# Set up some common variables before we launch into what might be
# considered boilerplate by now.
prog=saslauthd
path=/usr/sbin/saslauthd
start() {
        echo -n $"Starting $prog: "
        daemon $path -m $SOCKET -a $MECH $FLAGS
        RETVAL=$?
        echo
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/$prog
        return $RETVAL
}
```

/etc/init.d/saslauthd *(continued ....)*

```
stop() {
        echo -n $"Stopping $prog: "
        killproc $path
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/$prog
        return $RETVAL
}
restart() {
        stop
          start
}
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart)
        restart
        ;;
  status)
        status $path
        ;;
  condrestart)
        [ -f /var/lock/subsys/$prog ] && restart || :
        ;;
  *)
 echo $"Usage: $0 {start|stop|status|reload|restart|condrestart}"
 exit 1
 esac


exit $?
```

Start the saslauthd server as root :

```
    su
    /sbin/service saslauthd start
    /sbin/chkconfig saslauthd on
```

The last command ensures that it always starts at boot.

Cyrus Imap Configuration

Alter and note down the password for cyrus as follows:

```
su
passwd cyrus
Changing password for user cyrus.
New password:
Retype new password:
```

Add a sasl user account called cyrus with the same password as follows:

```
su
saslpasswd2 -c cyrus
Password: (enter your passwd)
Again (for verification): (enter your password)
```

Make sure that you remember the password. You will need it to fill the $CYRUS_PASSWORD required in the file /var/www/html/web-cyradm-0.5.3-1/config.inc.php described later in the document. Test it as follows:

```
su - cyrus
-bash-2.05b$ cyradm --user cyrus --server localhost
IMAP Password:
localhost.localdomain> quit
-bash-2.05b$quit
```

Edit the file **/etc/imapd.conf** to resemble the one in the following table

/etc/imapd.conf
configdirectory: /var/lib/imap
partition-default: /var/spool/imap
admins: cyrus
allowanonymouslogin: no
sieveuserhomedir: no
sievedir: /var/lib/imap/sieve
sendmail: /usr/sbin/sendmail
hashimapspool: true
sasl_pwcheck_method: saslauthd
sasl_mech_list: PLAIN
unixhierarchysep: yes
autocreatequota: 10000
lmtpsocket: /var/spool/postfix/public/lmtp
tls_cert_file: /usr/share/ssl/certs/cyrus-imapd.pem
tls_key_file: /usr/share/ssl/certs/cyrus-imapd.pem

Edit the file /etc/cyrus.conf to resemble the one in the following table:

```
### /etc/cyrus.conf
# standard standalone server implementation
START {
  # do not delete this entry!
  recover     cmd="ctl_cyrusdb -r"


  # this is only necessary if using idled for IMAP IDLE
#  idled           cmd="idled"
}
# UNIX sockets start with a slash and are put into /var/lib/imap/sockets
SERVICES {
  # add or remove based on preferences
  imap        cmd="imapd" listen="imap" prefork=5
  imaps       cmd="imapd -s" listen="imaps" prefork=1
  pop3        cmd="pop3d" listen="pop3" prefork=3
  pop3s       cmd="pop3d -s" listen="pop3s" prefork=1
  sieve       cmd="timsieved" listen="sieve" prefork=0
# at least one LMTP is required for delivery
#  lmtp       cmd="lmtpd" listen="lmtp" prefork=0
# lmtpunix    cmd="lmtpd" listen="/var/lib/imap/socket/lmtp" prefork=1
 lmtpunix     cmd="lmtpd" listen="/var/spool/postfix/public/lmtp" prefork=1
# this is only necessary if using notifications
#  notify     cmd="notifyd" listen="/var/lib/imap/socket/notify" proto="udp" prefork=1
}
EVENTS {
  # this is required
  checkpoint cmd="ctl_cyrusdb -c" period=30
  # this is only necessary if using duplicate delivery suppression
  delprune    cmd="ctl_deliver -E 3" period=1440
  # this is only necessary if caching TLS sessions
  tlsprune    cmd="tls_prune" period=1440
}
```

Web-cyradm

Assuming your apache and php modules rpms are installed correctly on your RedHat Linux, untar `web-cyradm-0.5.3-1.tar.gz` into `/var/www/html`

```
cd /var/www/html
tar -xvzf web-cyradm-0.5.3-1.tar.gz
mkdir /var/log/cyradm
chown apache /var/log/cyradm
```

and you should now find a directory: `/var/www/html/web-cyradm-0.5.3-1`

Inside this directory there exists a scripts directory which contains postgresql and mysql database schema that you should use.

Copy `/var/www/html/web-cyradm-0.5.3-1/config.inc.php-dist` to `/var/www/html/web-cyradm-0.5.3-1/config.inc.php`

As follows:

```
cp /var/www/html/web-cyradm-0.5.3-1/config.inc.php-dist \
        /var/www/html/web-cyradm-0.5.3-1/config.inc.php
```

Open and edit the file `/var/www/html/web-cyradm-0.5.3-1/config.inc.php` to look like the one shown in the table below. The most critical parts are highlighted.

```
### /var/www/html/web-cyradm-0.5.3-1/config.inc.php
```

```php
<?php
// Set Default language
$DEFAULTLANG="en_US";
# The Cyrus login stuff
$CYRUS_HOST="localhost";
$CYRUS_PORT="143";
$CYRUS_USERNAME="cyrus";
$CYRUS_PASSWORD="cyrussaslsecret";
/* DB_TYPE Possible Values are:
 o mysql
 o pgsql
 To operate a mailsystem with postgreSQL you will need a patch for Postfix.
 Other Databases needs to be supported by PAM and postfix
*/
$DB_TYPE="pgsql";
$DB_HOST="localhost";
$DB_NAME="mail";
$DB_USER="mail";
$DB_PASSWD="secret";
$DB_PROTOCOL="tcp"; // set to "tcp" for TCP/IP or "unix"
$DSN="$DB_TYPE://$DB_USER:$DB_PASSWD@$DB_PROTOCOL+$DB_HOST/$DB_NAME";
#Where should web-cyradm write its log to?
$LOG_DIR="/var/log/cyradm/";
# The default timeout in seconds for a session,
# after that you have to login again
$SESS_TIMEOUT=1000;
# The default quota sets the default quota for new domains
$DEFAULT_QUOTA=20000;
# Defines if passwords are encrypted or not. Valid Values:
#  - plain 0 No encription is used
#  - crypt 1 (shadow compatible encription)
#  - mysql 2 (MySQL PASSWORD cryto funtions
$CRYPT="plain";
# If you are using 2.1.x and wish to use email addresses  with .'s ....
# #  NOTE: you also have to add this
# line to your imapd.conf file /etc/imapd.conf: ####
# unixhierarchysep: yes ####
$DOMAIN_AS_PREFIX=1;
# At the moment, web-cyradm supports two methods of password change:
# - through sql
# - poppassd     # sql is the default
$PASSWORD_CHANGE_METHOD="sql";
?>
```

At this point, if you are using RedHat Linux 9 or have chosen to upgrade to PostgreSQL-7.3.x rpms please go to the Appendix section and read the RedHat 9 Notes before you proceed.

Database Scripts

To create the needed tables in the database:

```
su
su - postgres
createuser -P mail
createdb mail
exit
exit
psql mail -U mail -W -h 127.0.0.1 <  \
        /var/www/html/web-cyradm-0.5.3-1/scripts/create_pgsql.sql

psql mail -U mail -W -h 127.0.0.1
```

You will be prompted for a password. Enter the password you had entered when you run the "`createuser -P mail`" command and execute the following SQL queries:

```
ALTER TABLE domain ADD COLUMN transport VARCHAR(255);
ALTER TABLE domain ALTER COLUMN transport SET DEFAULT 'cyrus';
UPDATE domain SET transport='cyrus';
INSERT INTO adminuser (username, password) VALUES ('admin', 'test');
INSERT INTO domainadmin (domain_name,adminuser) VALUES ('*','admin');
INSERT INTO accountuser (username, password) VALUES ('cyrus', 'secret');
CREATE UNIQUE INDEX accountuser_unique_ndx ON accountuser(username);
CREATE UNIQUE INDEX domain_unique_ndx ON domain(domain_name);
CREATE UNIQUE INDEX virtual_unique_ndx ON virtual(alias,dest);
CREATE INDEX virtual_username_ndx ON virtual(username);
\q
```

Pam_pgsql configuration

Edit/Create the file `/etc/pam_pgsql.conf` to look as follows:

/etc/pam_pgsql.conf
database = mail
host = 127.0.0.1
user = mail
password = secret
table = accountuser
user_column = username
pwd_column = password
pw_type = clear

Edit/Create the file `/etc/pam.d/mail-auth`

/etc/pam.d/mail-auth
| ```
#%PAM-1.0
auth  sufficient /lib/security/pam_pgsql.so
auth  sufficient /lib/security/pam_unix_auth.so
account required /lib/security/pam_pgsql.so
account sufficient /lib/security/pam_unix_acct.so
``` |

Next, edit/Create the file `/etc/pam.d/imap`

/etc/pam.d/imap
| ```
#%PAM-1.0

auth       required      /lib/security/pam_stack.so service=mail-auth
account    required      /lib/security/pam_stack.so service=mail-auth
``` |

The advantage about this scheme is that should one need to change the authentication modules only one file: /etc/pam.d/mail-auth, need be changed. (*Thanks to Simon Matter for this tip!*)

If you will use Cyrus also for POP-Service just copy `/etc/pam.d/imap` to `/etc/pam.d/pop` The current web-cyradm uses sieve, so you must also make a pam module configuration for the sieve service by copying `/etc/pam.d/imap` to `/etc/pam.d/sieve` as follows:

```
cp /etc/pam.d/imap  /etc/pam.d/pop
cp /etc/pam.d/imap  /etc/pam.d/sieve
```

If you also want smtp authentication then (Please remember the additional postfix configuration required for smtp-auth.):

```
cp /etc/pam.d/imap /etc/pam.d/smtp
```

Postfix Configuration

As root open and edit /etc/postfix/main.cf on the following few highlighted sections:
(Please NOTE that there are many other options that you can set in this file. For example maximum mail size e.tc. Look at /etc/postfix/samples/ directory for examples. What is shown is the minimum required for a basic cyrus-imapd system.)

```
### /etc/postfix/main.cf

# You need to replace this with the full qualified name of the mail server
myhostname = localhost.localdomain.example
# The mydomain parameter specifies the local internet domain name.The default is to use $myhostname
# minus the first component. $mydomain is used as a default value for many other configuration
#parameters.
mydomain = localdomain.example
# SENDING MAIL
myorigin = $mydomain
#...etc
# The mydestination parameter specifies the list of domains that this machine considers itself the
# final destination for. That includes Sendmail-style virtual domains hosted on this machine.
mydestination = localhost, localhost.localdomain,
                $myhostname, localhost.$mydomain, $mydomain,
                pgsql:/etc/postfix/pgsql-mydestination.cf
#..... etc
alias_database = hash:/etc/postfix/aliases
alias_maps = hash:/etc/postfix/aliases
# .... etc
#
virtual_maps=pgsql:/etc/postfix/pgsql-virtual.cf


#The following only applies to postfix-2.0.x, I prefer to leave this empty deliberately to
#reduce the number of sql queries postfix makes when looking up domain names.
virtual_alias_domains =


# Outgoing addresses should be rewritten from e.g test0002 at domain to user.name at
# virtualhost.com. This is important if you like to use a webmail interface.
sender_canonical_maps = pgsql:/etc/postfix/pgsql-canonical.cf
#
mailbox_transport = lmtp:unix:public/lmtp


# .... etc
############## SMTP Authentication with SASL and PAM ###############################
##################################################################################
#### This section is optional for those who want SMTP-AUTH. You may leave it out all together.
####In order to enable SASL support in the SMTP server:
smtpd_sasl_auth_enable = yes
#### In order to allow mail relaying by authenticated clients:
smtpd_recipient_restrictions = permit_mynetworks permit_sasl_authenticated reject_unauth_destination
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain =
#### Please NOTE: the above line is deliberately equated to a ablank as in nothing
#
#### Older Microsoft SMTP client software implements a non-standard version of the AUTH protocol
#### syntax, To accommodate such clients in addition to conformant clients, uncomment the next line.
broken_sasl_auth_clients = yes
##################################################################################
```

As root create the file `/etc/postfix/pgsql-virtual.cf` and put the following entries:

```
## /etc/postfix/pgsql-virtual.cf
#
# pgsql config file for alias lookups on postfix
# comments are ok.
#
# the user name and password to log into the pgsql server
hosts = 127.0.0.1
user = mail
password = secret


# the database name on the servers
dbname = mail


# the table name
table = virtual
#
select_field = dest
where_field = alias
additional_conditions = and status = '1'
```

As root create the file `/etc/postfix/pgsql-mydestination.cf` and put the following entries:

```
## /etc/postfix/pgsql-mydestination.cf

# /etc/postfix/pgsql-mydestination.cf

####################################

# pgsql config file for local domain (like sendmail's sendmail.cw)

# lookups on postfix

# comments are ok.

#

# the user name and password to log into the pgsql server

hosts = 127.0.0.1

user = mail

password = secret


# the database name on the servers

dbname = mail


# the table name

table = domain

#

select_field = domain_name

where_field = domain_name
```

As root create the file `/etc/postfix/pgsql-canonical.cf` and put the following entries:

```
## /etc/postfix/pgsql-canonical.cf

# /etc/postfix/pgsql-canonical.cf

##############################################

# pgsql config file for canonical lookups on postfix comments are ok.

#

# the user name and password to log into the pgsql server

hosts = 127.0.0.1

user = mail

password = secret

# the database name on the servers

dbname = mail


# the table name

table = virtual

#

select_field = alias

where_field = username

# Return the first match only

additional_conditions = and status = '1' limit 1
```

This next caption is for those who require SMTP -AUTH functionality.

As root create the file `/usr/lib/sasl2/smtpd.conf` and put the following one line:

```
pwcheck_method: saslauthd
```

This is enough to achieve PLAIN SMTP-AUTH functionality. After starting postfix as detailed below outside this caption, test it as detailed in `/etc/postfix/README_FILES/SASL_README` file. Here is a typical successful test session (the highlighted lines are what I typed):

```
 [koros@tux koros]$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 tux.hiddendomain.co.ke ESMTP Postfix
EHLO localhost
250-tux.hiddendomain.co.ke
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH GSSAPI PLAIN LOGIN DIGEST-MD5 CRAM-MD5
250-AUTH=GSSAPI PLAIN LOGIN DIGEST-MD5 CRAM-MD5
250-XVERP
250 8BITMIME
AUTH PLAIN YW5kcmV3Lmtvcm9zZS5uZXXQAYW5kcmV3Lmtvcm9zZS5uZXXQAdGVzdA==
235 Authentication successful
quit
221 Bye
Connection closed by foreign host.
```

Once everything is working now read `/etc/postfix/samples/sample-tls.cf` to learn about TLS. To enable TLS service for postfix edit you `/etc/postfix/master.cf` and uncomment the smtps line by removing the highlighted character:

```
##/etc/postfix/master.cf
## ========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#              (yes)   (yes)   (yes)   (never) (100)
# ========================================================================
smtp    inet    n       -       y       -       -       smtpd
#smtps   inet    n       -       n       -       -       smtpd
```

This is not enough. You must now read and understand `/etc/postfix/samples/sample-tls.cf` and add all the entries in it to `/etc/postfix/main.cf`. A quick way to do it is:

```
echo "$(cat /etc/postfix/samples/sample-tls.cf)" >>/etc/postfix/main.cf
```

This will append the contents of the file to the end of `/etc/postfix/main.cf`.
Now edit `/etc/postfix/main.cf` accordingly.
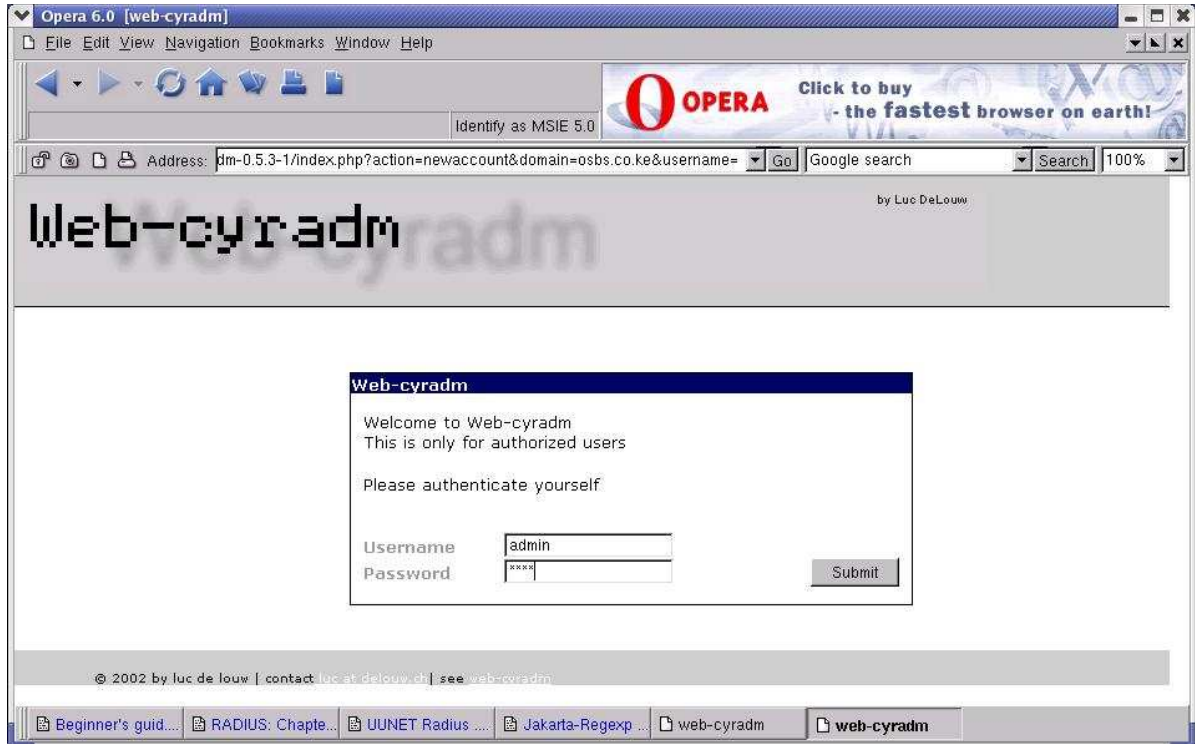
Start the `postfix` server as root :

```
    su
/sbin/service postfix start
/sbin/chkconfig postfix on
```

The last command ensures that it always starts at boot.

Finally start the apache web server as root :

```
su
/sbin/service httpd start
/sbin/chkconfig httpd on
```
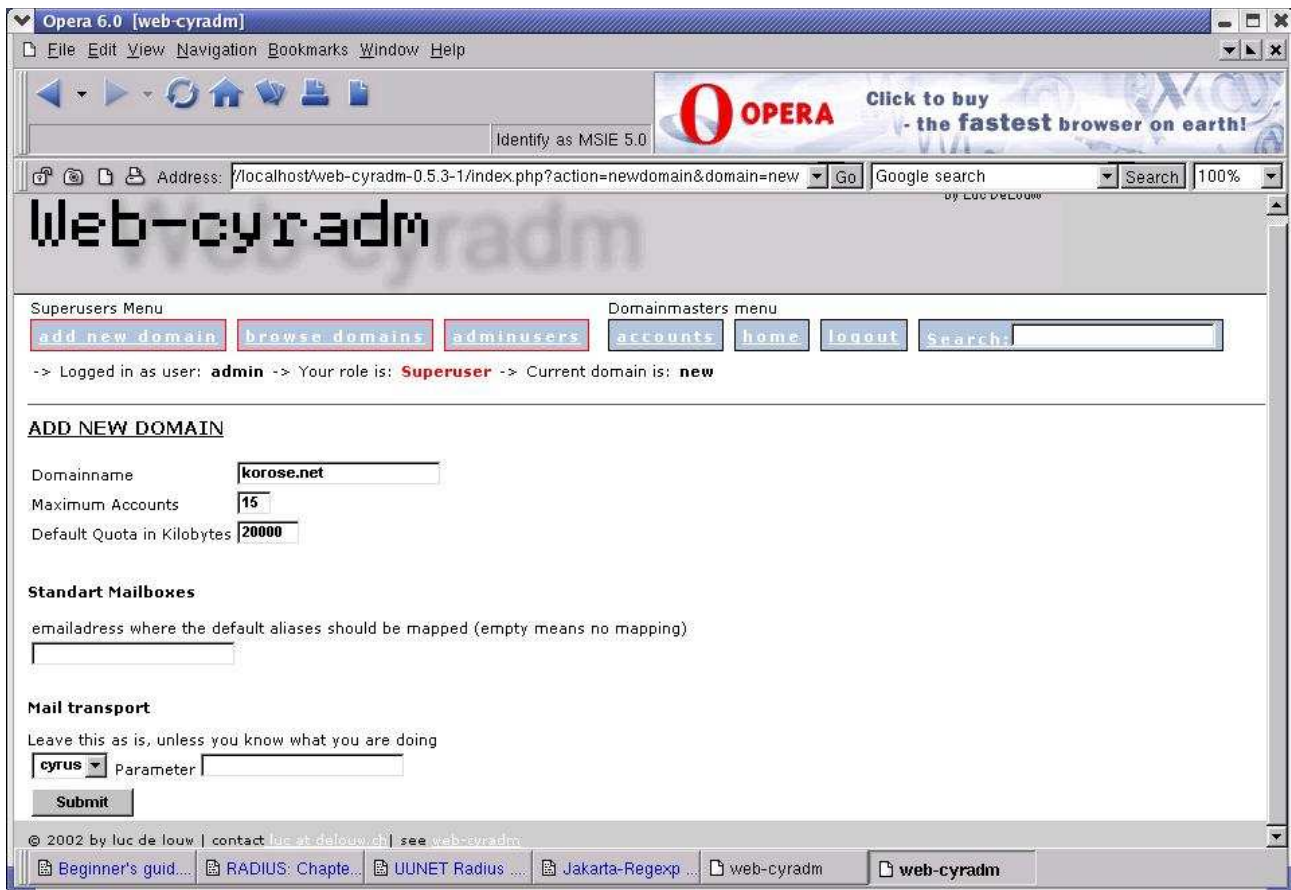
The last command ensures that it always starts at boot. Now launch your favorite browser and point it to http://localhost/web-cyradm-0.5.3-1 You should see the following screen :

You can now add domains and accounts. Login as admin with password "test" or whatever password you used when you had run this SQL command:

"INSERT INTO adminuser (username, password) VALUES ('admin', 'test');"

to add domains and domain admins and domain users. Look at the screenshots below :

Once you add a few domains you will see the following:

Click on the "accounts" for the repective domain to add accounts

Web-cyradm

by Luc DeLouw

Superusers Menu

add new domain | **browse domains** | **adminusers**

Domainmasters menu

accounts | **home** | **logout** | **Search:**

-> Logged in as user: **admin** -> Your role is: **Superuser** -> Current domain is: **korose.net**

ADD NEW ACCOUNT TO DOMAIN KOROSE.NET

Total accounts: 4

Email address	andrew	@korose.net
Quota	20000	
Password	****	
Confirm Password	****	

Submit

© 2002 by luc de louw | contact luc at delouw.ch | see web-cyradm

Beginner's guid... | RADIUS: Chapte... | UUNET Radius ... | Jakarta-Regexp ... | web-cyradm | **web-cyradm**

Web-cyradm

by Luc DeLouw

BROWSE ACCOUNTS FOR DOMAIN KOROSE.NET

Total accounts: 4

Add new account | **Previous 10 entries** | **Next 10 entries**

action						Email address	Username	Quota used
Edit account	Change Password	Forward	Delete account	Set quota	Set catch all	andrew@korose.net	andrew.korose.net	1 Kbytes out of 20000 Kbytes (0.01 %)
Edit account	Change Password	Forward	Delete account	Set quota	Set catch all	dan@korose.net	dan.korose.net	0 Kbytes out of 20000 Kbytes (0.00 %)
Edit account	Change Password	Forward	Delete account	Set quota	Set catch all	peter@korose.net	peter.korose.net	0 Kbytes out of 20000 Kbytes (0.00 %)
Edit account	Change Password	Forward	Delete account	Set quota	Set catch all	rose@korose.net	rose.korose.net	0 Kbytes out of 20000 Kbytes (0.00 %)

© 2002 by luc de louw | contact luc at delouw.ch | see web-cyradm

Beginner's guid... | RADIUS: Chapte... | UUNET Radius ... | Jakarta-Regexp ... | web-cyradm | **web-cyradm**

Email forwarding...

vacation settings...

Security Considerations

Cyrus imap is meant to be run on "sealed" servers, where users are not normally permitted to log in. But usually, sometimes due to shortage of resources, the server could have shell accounts for users for other purposes. This presents a security concern since a large number of the configuration files detailed here have passwords shown in plain text.

To minimize the risk of unprivileged users getting these passwords do the following:

```
su -
chown root /etc/pam_pgsql.conf /etc/postfix/pgsql-canonical.cf \
           /etc/postfix/pgsql-virtual.cf /etc/postfix/pgsql-mydestination.cf

chown apache /var/www/html/web-cyradm-0.5.3-1/config.inc.php

chmod 600 /etc/pam_pgsql.conf  /var/www/html/web-cyradm-0.5.3-1/config.inc.php \
          /etc/postfix/pgsql-canonical.cf /etc/postfix/pgsql-virtual.cf \
          /etc/postfix/pgsql-mydestination.cf

psql -U mail -W -h 127.0.0.1 mail
REVOKE ALL PRIVILEGES ON
 accountuser, adminuser, alias, domain, domainadmin, search, virtual FROM PUBLIC;
\q
```

The next few commands are optional for the truly paranoid:

```
groupadd wheel
chgrp wheel /bin/su /usr/bin/reboot /usr/bin/halt
chmod 4750 /bin/su /usr/bin/reboot /usr/bin/halt
```

If you have sudo installed you may also want to:
```
chgrp wheel /usr/bin/sudo; chmod 4750 /usr/bin/sudo
```

In this case only members of the "wheel" group can su , sudo halt etc on the system.

Lastly, change the passwords shown in this document to more complex ones. For example to change the admin password for web-cyradm interface do:

```
psql -U mail -W -h 127.0.0.1 mail
UPDATE adminuser SET password='xxxxxx' WHERE username='admin';
```

Similarly, if you have created some domain admins:
```
UPDATE adminuser SET password='xxxxxx' WHERE username='someDomainadmin';
\q
```

This should minimize local security concerns.

Troubleshooting

If you have reached this point and you are still having problems, here is a quick checklist to help in troubleshooting. (You are however advised to check postfix, cyrus and web-cyradm on-line FAQ's for more help).

Authentication problems

1. Is PostgreSQL running and listening on port 5432 .

    Test it with :
    ```
    psql -U mail -h 127.0.0.1 -W mail
    ```
      or
    ```
    telnet  127.0.0.1 5432
    ```

    Follow all the instructions about postgresql at the beginning of
    the document if the tests fail.

2. Have you setup the correct database name, protocol, username and password to postgresql on
    `config.inc.php` ?

    ```
    $DB_TYPE="pgsql";
    $DB_HOST="localhost";
    $DB_NAME="mail";
    $DB_USER="mail";
    $DB_PASSWD="secret";
    $DB_PROTOCOL="tcp";
    ```

3. Is the postgresql php driver installed? :
    ```
    "rpm -q php-pgsql"
    ```
    should give something like:
    php-pgsql-4.2.2-8.0.7

4. Is the postgresql php driver enabled in the `/etc/php.ini`:?
    make sure the following line is in the "Dynamic Extensions" section of `/etc/php.ini`
    exists and is not commented out:

    ```
    extension=pgsql.so
    ```

5. Are you having sieve authentication problems with **avelsieve** on **squirrelmail ?**
    Some php scripts require that you turn `register_globals` on in the `/etc/php.ini` file.
    Therefore find the following line and edit it to look as follows:

    ```
    register_globals = On
    ```

6. Having problems changing superuser password and domain admin passwords using web-cyradm? At the time of writing there seems to be a bug with web-cyradm- 0.5.3-1 and earlier when dealing with "plain" passwords and domain admins. It seems like in postgresql it creates the account but sets a blank password. Similarly, if you use web-cyradm to change the admin (superuser) password it will insert a blank password.

    The workaround is to manually set the password in the database.

To manually set the password on postgresql do the following:

```
psql -U mail -W -h 127.0.0.1 mail

UPDATE adminuser SET password='xxxxxx' WHERE username='admin';
UPDATE adminuser SET password='xxxxxx' WHERE username='someDomainadmin';
\q
```

where 'xxxxxx' is the password you want.

7. Can't receive systems' postmaster's mail?: Please add an alias in /etc/postfix/aliases pointing to the email of the system adminstator then run "`/usr/sbin/postalias /etc/postfix/aliases`":

```
##/etc/postfix/aliases
#########################
#.... there are other entries here
postmaster:     root
root:           postfix
postfix:        admin@mydomainsomwhere.com
```

8. Are you using PostgreSQL version 7.2.x or 7.3.x? Please note that version 7.3.x of PostgreSQL is NOT compatible with 7.2.x. If you choose to upgrade to Postgresql-7.3.x please note that 7.3 requires all client programs to be recompiled with the newer library and a dump and restore of your database data if you were already running postgresql-7.2.x or earlier.

   RedHat provides a special rpm called `postgresql72-libs` together with the `postgresql-7.3.2-3` rpms to solve the problem and avoid recompiling client applications. The two srpms are available here:

   http://mirrors.kernel.org/redhat/redhat/linux/rawhide/SRPMS/SRPMS/

   download and rebuild them as follows
   ```
   rpmbuild --rebuild postgresql-7.3.2-3.src.rpm
   rpmbuild --rebuild postgresql72-1-3.src.rpm
   ```

   Now install the rpms as usual then proceed to the Appendix Section and read the RedHat 9 notes which apply to PostgreSQL-7.3.x

Logging Errors:

1. Did you create the log directory?
   ```
   mkdir /var/log/cyradm
   ```

2. Is it writeable by apache?
   ```
   chown apache /var/log/cyradm
   ```

Routine tasks

After a creating or deleting a large number of accounts always vacuum the database:

```
su
su - postgres
/usr/bin/vacuumdb -f -a -z
```

A good idea would be to create a cron job to run once at night say at midnight to vacuum the postgresql database. Especially if you often add/delete accounts. Here is how to do a simple crontab that runs one hour after midnight. This assumes that you know how to use the vi editor :

```
su
su - postgres
crontab -e
```

###Sample postgres crontab to vacuumdb
| ```
MAILTO=root
0 1 * * * /usr/bin/vacuumdb -f -a -z >>/dev/null 2>&1
``` |

Important Security/Bug Note:

PostgreSQL versions 7.2.1 and 7.2.2 contain a serious issue with the VACUUM command when it is run by a non-superuser. It is possible for the system to prematurely remove old transaction log data (pg_clog files), which can result in unrecoverable data loss. **Please upgrade to 7.2.3** *See* **https://rhn.redhat.com/errata/RHSA-2003-001.html**

Final Words Of Advice

Cyrus is a complex piece of software. To really manage it properly my advice is that you need to understand a little more about how cyrus works.
Have a look at the book :

"Managing IMAP" By Dianna Mullet &Kevin Mullet Published by O'Reilly&Associates

while it refers to version 1.x series and a bit of version 2.0.x of cyrus, it is well written and will help to give the big picture and answer a few of any puzzling questions about the cyrus-imap system.

Also look at the docs in `/usr/share/doc/cyrus-imapd-2.1.x/` on your Linux machine.

Above all be patient,don't give up, check the various cyrus, web-cyradm and postfix mailing list archives and it will eventually work as it worked for me.

Additional Ideas

- If you like Sieve and you use squirrelmail then, checkout **avelsieve**!! *http://pacific.edunet.uoa.gr/projects/avelsieve/* . Another nice sieve tool is *Smartsieve* (http://smartsieve.sourceforge.net/) also written in php. Note however, that for cyrus-imapd versions before 2.1.12 sieve doesn't seem to work with cyrus mailboxes of the type 'user/firstname.secondname" for example when using the " `unixhierarchysep: yes` "option in the `/etc/imapd.conf` because of the dot.
- If you use squirrelmail heavily consider up-imap proxy: http://freshmeat.net/projects/imapproxy/
- Java developers can replace the web-cyradm with a servlet and take advantage of the javamail API
- Use encrypted passwords (needs postgresql-contrib package and web-cyradm-0.5.2 or newer)
- Extend the interface to handle domain aliases
- Use OpenLDAP to manage users and addresses Agood example is Jamm (http://jamm.sourceforge.net/) but doesn't use cyrus-imapd though :(-
- Multiple IMAP/POP servers for distributed load. (See Perdition http://vergenet.net/linux/perdition/)
- Use Replication for redundancy and load balancing: checkout **dbmirror** works beautifully :-) http://developer.postgresql.org/cvsweb.cgi/pgsql-server/contrib/dbmirror/

Appendix

SECTION I - Redhat 9 Notes

RedHat Linux 9 ships with PostgreSQL -7.3.2 and this presents a special problem for those who want to use cyrus-imapd with web-cyradm and postgresql7.3. While RedHat Linux 9 has many enhancements notably the Native POSIX Thread Library(NPTL) which improves performance and many more (see http://www.gurulabs.com/RedHatLinux9-review.html), the php version that it ships has a bug in it's PEAR:DB that relates to PostgreSQL 7.3.x.PEAR is a framework and distribution system for reusable PHP components. Web-cyradm uses PEAR to connect to the PostgreSQL/MySQL databases.

Specifically, modifyLimitQuery() function in pear generates wrong LIMIT clause. This is not RedHat's fault though, this bug is fixed in pear-1.4-beta1 (http://pear.php.net/package-changelog.php?pacid=46&release=1.4b1)

The work-around to this problem is to upgrade the php PEAR:DB component to the version 1.4b1. However since this is not a part of the official RedHat rpm for php then if you later use rpm to update php, which usually happens when RedHat gives a security update, then your changes will be overwritten. So always remember to repeat this procedure whenever that happens.

I describe this 'hack' here as a temporary solution in **Section II** but since it's a beta version I don't know what else it may break. You are therefore advised to use RedHat-7.3/8.0 instead while waiting for an official php rpm that includes the newer PEAR:DB. However as another option in RedHat 9 feel free to use MySQL instead which I briefly describe in **Section III**

A second problem that relates directly to PostgreSQl-7.3 and web-cyradm is timestamp precission values. `timestap(p)` has changed between PostgreSQL 7.2.x and 7.3.x According to the Postgresql-7.3 reference manual: (http://www.postgresql.org/docs/view.php?version=7.3&idoc=1&file=datatype-datetime.html) "timestap(p) The optional precision p should be between 0 and 6, This can easily be fixed in the `create_pgsql.sql` script.

SECTION II -Upgrading RedHat 9 PEAR:DB
*   Upgrading RedHat 9 PEAR:DB to 1.4b1

Download PEAR:DB Distribution DB-1.4b1 here: http://pear.php.net/get/DB

Then do the following:
```
su
tar xvfz DB-1.4b1.tgz -C /usr/local
mv /usr/share/pear/DB /usr/share/pear/DB_RedHat
ln -s /usr/local/DB-1.4b1/DB /usr/share/pear/DB
```

*   Edit the web-cyradm `create_pgsql.sql` script and find the line:

```
timestamp timestamp(13) NOT NULL,
```
Edit it to look like this:
```
timestamp timestamp NOT NULL,
```
You may now proceed with section "**Database Scripts**"

SECTION III -MySQL rpm Notes

For those who want to use RedHat 9 with web-cyradm and MySQL this is not a complete reference but a few tips on getting an rpm based system working. The rest of the cyrus-imapd setup is as described in the earlier part of the document but with a few changes to postfix configuration. For further clarification look at Luc's Howto (http://www.delouw.ch/linux/Postfix-Cyrus-Web-cyradm-HOWTO/html/index.html)

Install the following rpms included in your RedHat Linux CD's usually in disc3:

1. mysql-3.23.xx.i386.rpm
2. mysql-server-3.23.xx.i386.rpm
3. php-mysql-4.x.x.i386.rpm
4. mysql-devel-3.23.xx.i386.rpm

As follows:
```
su
rpm  -Uvh <package-x.xx-x.i386.rpm>
```

Then in your home directory do the following as a non-root user if you haven' t done it already:

```
mkdir -p $HOME/rpm/RPMS/i386
mkdir $HOME/rpm/SRPMS
mkdir $HOME/rpm/SPECS
mkdir $HOME/rpm/SOURCES
mkdir $HOME/rpm/BUILD
echo "%_topdir    $HOME/rpm" >$HOME/.rpmmacros
```

Obtain and install the source rpm for postfix(see Software Requirements section for the URL'):
```
rpm -ivh postfix-2.0.8-1.src.rpm
cd `rpm --eval '%{_sourcedir}'`
export POSTFIX_REDHAT_MYSQL=1
export POSTFIX_SASL=2
export POSTFIX_TLS=1 # This is optional. It requires openssl-devel
sh make-postfix.spec
cd `rpm --eval '%{_specdir}'`
rpmbuild -ba postfix.spec
```

Download pam_mysql srpm from http://home.teleport.ch/simix/RPMS/Cyrus-imapd/contrib/

```
wget -c http://home.teleport.ch/simix/RPMS/Cyrus-imapd/contrib/pam_mysql-0.5-0.src.rpm
```

Make sure pam-devel rpm is installed, then rebuild the pam_mysql rpm as usual:
```
rpmbuild --rebuild  pam_mysql-0.5-0.src.rpm
```

Then install the resulting rpms:
```
rpm -Uvh /home/<username>/rpm/RPMS/i386/pam_mysql-0.5-0.i386.rpm
rpm -Uvh /home/<username>/rpm/RPMS/i386/postfix-2.0.8-1.mysql.sasl2.tls.rh9.i386.rpm
```
Then do the following:
```
su -
/sbin/service mysqld start
/sbin/chkconfig mysqld on
mysql mysql </var/www/html/web-cyradm-0.5.3-1/scripts/insertuser_mysql.sql
mysql mail < /var/www/html/web-cyradm-0.5.3-1/scripts/create_mysql.sql
mysql mysql
GRANT ALL ON mail.* TO mail@127.0.0.1 IDENTIFIED BY 'secret' WITH GRANT OPTION;
quit
mysqladmin reload
```

Edit/create /etc/pam.d/mail-auth

```
#%PAM-1.0

auth sufficient /lib/security/pam_mysql.so user=mail passwd=secret host=localhost \
db=mail table=accountuser usercolumn=username   passwdcolumn=password crypt=1 sqlLog=0

auth        sufficient  /lib/security/pam_unix_auth.so

account required /lib/security/pam_mysql.so user=mail passwd=secret host=localhost \
db=mail table=accountuser usercolumn=username   passwdcolumn=password crypt=1 sqlLog=0

account     sufficient  /lib/security/pam_unix_acct.so
```

Note: The ″\″ should not be there. They indicate that the line continues onto the next and should not wrap.

/etc/pam.d/imap remains as it was shown for postgresql i.e.

```
#%PAM-1.0

auth        required     /lib/security/pam_stack.so service=mail-auth
account     required     /lib/security/pam_stack.so service=mail-auth
```

Then ,

```
     cp /etc/pam.d/imap  /etc/pam.d/pop
     cp /etc/pam.d/imap  /etc/pam.d/sieve
     cp /etc/pam.d/imap /etc/pam.d/smtp
```

Now you need to configure postfix and web-cyradm. It's quite similar to what was done for the postgresql version but with "crypt" type passwords. Please make reference to Luc's howto . A text version is included in the web-cyradm doc directory:

```
 /var/www/html/web-cyradm-0.5.3-1/doc/Postfix-Cyrus-Web-cyradm-HOWTO.txt
```

IMPORTANT:
The postfix rpm version described here runs fully chrooted. It will need to connect to mysql over tcp. So remember to replace "hosts = localhost" in the mysql-mydestination.cf, mysql-virtual.cf and mysql-canonical.cf with 127.0.0.1 for example:

```
     # /etc/potsfix/mysql-mydestination.cf
     ####################################################
     # the user name and password to log into the mysql server
     hosts = 127.0.0.1
     user = mail
     password = secret

     # the database name on the servers
     dbname = mail

     # the table name
     table = domain
     #
     select_field = domain_name
     where_field = domain_name
```

A tail of /var/log/maillog as postfix is running will give you an idea of what problems postfix may be having while attempting to connect to mysql:

```
     su
     tail -f /var/log/maillog
```

References

1. http://www.delouw.ch/linux/Postfix-Cyrus-Web-cyradm-HOWTO/html/index.html
2. http://www.wl0.org/~sjmudd/postfix/en/building-rpms/
3. "Managing IMAP" By Dianna Mullet &Kevin Mullet Published by O' Reilly &Associates
4. "Postfix on an ext3 filesystem"
   http://www.stahl.bau.tu-bs.de/~hildeb/postfix/postfix_ext3.shtml
5. Advanced filesystem implementor' s guide, Part 8 "Surprises in ext3"
   http://www-106.ibm.com/developerworks/linux/library/l-fs8/
6. "Postfix SMTP AUTH (and TLS) HOWTO"
   http://postfix.state-of-mind.de/patrick.koetter/smtpauth/index.html
7. "SMTP Authentication with Postfix and MySQL"
   http://small.dropbear.id.au/myscripts/postfixmysql.html