Virtual Mail hosting - Postfix and Dovecot using Maildir format with System Accounts

(This is a step by step guide)

Other related step by step guides

? **Virtual Mail hosting - Postfix and Dovecot using Mbox format with System Accounts**

? **Virtual Mail hosting - Postfix and Dovecot using Maildir format without system accounts**

? **Virtual Mail hosting - Postfix and Dovecot using Maildir format without system accounts using MySQL**

? **Virtual Mail hosting - Postfix and Dovecot using Maildir format without system accounts using LDAP**

? **Virtual Mail hosting (for ISP) - Postfix and Dovecot using Maildir format without system accounts using LDAP with back-sql using MySQL**

? **Virtual Mail hosting (for ISP) - Postfix and Dovecot using Maildir format without system accounts using LDAP with back-sql using postgresql**

? **Postfix Dovecot spamAssassin ClanAV amavisd-new squirrelmail**

? **Secure Postfix Mail Gateway**

This document gives guidance for mail retrieval and sending using postfix and dovecot in pop3. The setup described here uses maildir format for message storing and system account for the mail users. All you need to change is your domain name since configuration files are also attached at the end of this doc. ✍

Building Postfix

When you download the postfix package from the http://www.postfix.org it comes as a collection which is first subjected to 'tar' and then to compression. Hence first you have to unzip the files using the following command in the terminal. In this example the files are stored in the /usr/local/src directory.

```
$ gzip -d postfix-2.2.10.tar.gz
```

It will extract the files and then you have to use the following command.

```
$ tar -xf postfix-2.2.10.tar
```

Then a directory called postfix-2.2.10 will appear in you current directory (in this case in the src directory). Then change the directory to postfix-2.2.10 using the following command.

```
$ cd postfix-2.2.10
```

If you are intended to use the default parameters, then type

```
$ make
```

It will compile all the codes.

Installing Postfix

To install postfix you have to be the root user. There you need to create an account that will own the postfix mailq and most of the processors. The account should not permit logins and doesn't need either a shell or a home directory. Normally the user name has to be **postfix.** You can simply add a new user with the **useradd** command. Then in the **passwd** file in **/etc** directory make the default directory and shell to false values.

```
postfix:*:2001:2001:postfix:/no/where:/bin/false
```

Then create a user group that is not used by any user account. The name of the group will be **postdrop**.

```
 postdrop:*:2002
```

When you install postfix it will replace the existing **sendmail** service, mailq and newaliases with its own version. Hence as a precaution you need to rename the sendmail packages.

To find the location of sendmail use the following command.

```
$  whereis sendmail
```

You have to rename the binary file with no extensions. For an example assume that it is in the /usr/sbin directory. Then type the following command.

```
$ mv /usr/sbin/sendmail  /usr/sbin/sendmail.orig
```

Then rename the other two files also.

```
$ mv /usr/bin/mailq  /usr/bin/mailq.orig
$ mv /usr/bin/newaliases  /usr/bin/newaliases.orig
```

After renaming the files you have to execute the following command.

```
$ make install
```

When the installation proceeds the script will ask certain questions. The default values are also shown along with the question. You can accept the default values by just pressing the “ENTER” key. If you want to create an installable package you have to change the **install_root** parameter.

Finally the POSTFIX will be installed.

Postfix Configuration

This setup has two separate domains namely ceylonlinux.com and ceygate.com. These two domains have two users named **suranga.** (Two distinct users with the same name.) To configure **postfix** with the above configuration use the following method.

First you have to mention the two domains in the **virtual_alias_domains** parameter.

virtual_alias_domains = ceygate.com, ceylonlinux.com

Then you have to create separate user account for each e-mail address that is used within the system.

useradd suranga –u 2002
useradd suranga1 –u 2200

The system accounts have not to be matched with the e-mail accounts you have created. These accounts are mapped to the e-mail account in a separate file. In the main.cf file make a pointer to that specific file.
virtual_alias_maps = hash: /etc/postfix/virtual_alias
virtual_alias is the name of the file that map the system accounts with the relevant e-mail account.

Then create a **virtual_alias** file.

vim /etc/postfix/virtual_alias

Include the following information in that file.

```
suranga@ceylonlinux.com          suranga@localhost
suranga@ceygate.com              suranga1@localhost
```

As it can be seen suranga is the system account that is used for the suranga@ceylonlinux.com user and suranga1 is the system account that is used for the suranga@ceygate.com user.

After editing the virtual_alias file type the command

$ postmap virtual_alias

Then you have to edit the canonical maps parameter in the main.cf file.

canonical_maps = hash:/etc/postfix/canonical

Then create the /etc/postfix/canonical_maps file and insert the following information.

```
suranga                    suranga@ceylonlinux.com
suranga1                   suranga@ceygate.com
```

Then execute the following command

```
postmap /etc/postfix/canonical
```

Then you have to specify to use maildir format in the main.cf file.

```
home_mailbox = Maildir/
```

Then you need to create **Maildir** directories in the home directory of each user.

```
mkdir  /home/suranga/Maildir
mkdir /home/suranga1/Maildir
```

Then change the owner of these directories

```
chown suranga /home/suranga/Maildir
chown suranga1 /home/suranga1/Maildir
```

Then create the following three subdirectories.

```
mkdir /home/suranga/Maildir/new
mkdir /home/suranga/Maildir/cur
mkdir /home/suranga/Maildir/tmp
```

Then change the mode of those files.

```
chmod  700 /home/suranga/Maildir/*
```

Follow the same procedure for the user suranga1 also.

Then reload postfix.

```
postfix reload
```

Configuring Dovecot

To configure dovecot to use maildir add the following statement in the **/usr/local/dovecot.conf** file.

```
default_mail_env = maildir:/home/%u/Maildir
```

Then you need to specify it to use /etc/passwd and /etc/shadow files for authentication purposes.

```
auth default {
mechanisms = plain
          passdb passwd {
          args = /etc/passwd
           }
          passdb shadow {
          args = /etc/shadow
          }
          userdb passwd {
          args = /etc/passwd
          args = mail = maildir:/home/%u/Maildir
          }
}
```

Then if dovecot is already running find the process that is listening to port 110 (pop3)

```
lsof –i :110
```

Then kill that process and start dovecot

```
/usr/local/sbin/dovecot
```

Then telnet to port 110 and check whether dovecot is working properly.

```
telnet localhost 110
```

type the following commands

```
user suranga
pass ceylonlinux.
```

Note that ceylonlinux is user suranga's password. If the setup is correct you will be able to stay logged in.

Sample configuration files of postfix and dovecot are given below.

Here is our main.cf

```
# Global Postfix configuration file. This file lists only a subset
# of all 300+ parameters. See the postconf(5) manual page for a
# complete list.
#
```

```
# The general format of each line is: parameter = value. Lines
# that begin with whitespace continue the previous line. A value can
# contain references to other $names or ${name}s.
#
# NOTE - CHANGE NO MORE THAN 2-3 PARAMETERS AT A TIME, AND TEST
IF
# POSTFIX STILL WORKS AFTER EVERY CHANGE.

# SOFT BOUNCE
#
# The soft_bounce parameter provides a limited safety net for
# testing.  When soft_bounce is enabled, mail will remain queued that
# would otherwise bounce. This parameter disables locally-generated
# bounces, and prevents the SMTP server from rejecting mail permanently
# (by changing 5xx replies into 4xx replies). However, soft_bounce
# is no cure for address rewriting mistakes or mail routing mistakes.
#
#soft_bounce = no

# LOCAL PATHNAME INFORMATION
#
# The queue_directory specifies the location of the Postfix queue.
# This is also the root directory of Postfix daemons that run chrooted.
# See the files in examples/chroot-setup for setting up Postfix chroot
# environments on different UNIX systems.
#
queue_directory = /var/spool/postfix

# The command_directory parameter specifies the location of all
# postXXX commands.
#
command_directory = /usr/sbin

# The daemon_directory parameter specifies the location of all Postfix
# daemon programs (i.e. programs listed in the master.cf file). This
# directory must be owned by root.
#
daemon_directory = /usr/libexec/postfix

# QUEUE AND PROCESS OWNERSHIP
#
# The mail_owner parameter specifies the owner of the Postfix queue
# and of most Postfix daemon processes.  Specify the name of a user
# account THAT DOES NOT SHARE ITS USER OR GROUP ID WITH OTHER
ACCOUNTS
# AND THAT OWNS NO OTHER FILES OR PROCESSES ON THE SYSTEM.  In
# particular, don't specify nobody or daemon. PLEASE USE A DEDICATED
# USER.
#
mail_owner = postfix

# The default_privs parameter specifies the default rights used by
# the local delivery agent for delivery to external file or command.
# These rights are used in the absence of a recipient user context.
# DO NOT SPECIFY A PRIVILEGED USER OR THE POSTFIX OWNER.
#
#default_privs = nobody

# INTERNET HOST AND DOMAIN NAMES
#
# The myhostname parameter specifies the internet hostname of this
# mail system. The default is to use the fully-qualified domain name
# from gethostname(). $myhostname is used as a default value for many
# other configuration parameters.
#
myhostname = three.ceylonlinux.com
#myhostname = virtual.domain.tld

# 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 = ceylonlinux.com

# SENDING MAIL
#
# The myorigin parameter specifies the domain that locally-posted
# mail appears to come from. The default is to append $myhostname,
# which is fine for small sites.  If you run a domain with multiple
# machines, you should (1) change this to $mydomain and (2) set up
# a domain-wide alias database that aliases each user to
# user@that.users.mailhost.
#
# For the sake of consistency between sender and recipient addresses,
# myorigin also specifies the default domain name that is appended
# to recipient addresses that have no @domain part.
#
myorigin = $myhostname
myorigin = $mydomain

# RECEIVING MAIL
# The inet_interfaces parameter specifies the network interface
# addresses that this mail system receives mail on.  By default,
# the software claims all active interfaces on the machine. The
# parameter also controls delivery of mail to user@[ip.address].
#
# See also the proxy_interfaces parameter, for network addresses that
# are forwarded to us via a proxy or network address translator.
#
# Note: you need to stop/start Postfix when this parameter changes.
#
#inet_interfaces = all
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = ceylonlinux.com,ceygate.com

#inet_interfaces = $myhostname
#inet_interfaces = $myhostname, localhost
#virtual_mailbox_domains = ceylonlinux.com,ceygate.com
virtual_mailbox_base = /usr/local/vmail
virtual_mailbox_maps = hash:/etc/postfix/virtual1
virtual_uid_maps = static:1003
virtual_gid_maps = static:1005
#virtual_alias_domains  = ceylonlinux.com, ceygate.com
#virtual_alias_maps = hash:/etc/postfix/virtual_alias

# The proxy_interfaces parameter specifies the network interface
# addresses that this mail system receives mail on by way of a
# proxy or network address translation unit. This setting extends
# the address list specified with the inet_interfaces parameter.
#
# You must specify your proxy/NAT addresses when your system is a
# backup MX host for other domains, otherwise mail delivery loops
# will happen when the primary MX host is down.
#
#proxy_interfaces =
#proxy_interfaces = 1.2.3.4

# The mydestination parameter specifies the list of domains that this
# machine considers itself the final destination for.
#
# These domains are routed to the delivery agent specified with the
# local_transport parameter setting. By default, that is the UNIX
# compatible delivery agent that lookups all recipients in /etc/passwd
# and /etc/aliases or their equivalent.
#
# The default is $myhostname + localhost.$mydomain.  On a mail domain
# gateway, you should also include $mydomain.
#
# Do not specify the names of virtual domains - those domains are
# specified elsewhere (see VIRTUAL_README).
#
# Do not specify the names of domains that this machine is backup MX
# host for. Specify those names via the relay_domains settings for
# the SMTP server, or use permit_mx_backup if you are lazy (see
# STANDARD_CONFIGURATION_README).
#
# The local machine is always the final destination for mail addressed
# to user@[the.net.work.address] of an interface that the mail system
# receives mail on (see the inet_interfaces parameter).
#
# Specify a list of host or domain names, /file/name or type:table
# patterns, separated by commas and/or whitespace. A /file/name
# pattern is replaced by its contents; a type:table is matched when
# a name matches a lookup key (the right-hand side is ignored).
# Continue long lines by starting the next line with whitespace.
#
# See also below, section "REJECTING MAIL FOR UNKNOWN LOCAL USERS".
#
#mydestination = $myhostname, localhost.$mydomain, localhost
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = ceylonlinux.com,ceygate.com
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain,
#         mail.$mydomain, www.$mydomain, ftp.$mydomain

# REJECTING MAIL FOR UNKNOWN LOCAL USERS
#
# The local_recipient_maps parameter specifies optional lookup tables
# with all names or addresses of users that are local with respect
# to $mydestination, $inet_interfaces or $proxy_interfaces.
#
# If this parameter is defined, then the SMTP server will reject
# mail for unknown local users. This parameter is defined by default.
#
# To turn off local recipient checking in the SMTP server, specify
# local_recipient_maps = (i.e. empty).
#
# The default setting assumes that you use the default Postfix local
# delivery agent for local delivery. You need to update the
# local_recipient_maps setting if:
#
# - You define $mydestination domain recipients in files other than
#   /etc/passwd, /etc/aliases, or the $virtual_alias_maps files.
#   For example, you define $mydestination domain recipients in
#   the $virtual_mailbox_maps files.
#
# - You redefine the local delivery agent in master.cf.
#
# - You redefine the "local_transport" setting in main.cf.
#
# - You use the "luser_relay", "mailbox_transport", or "fallback_transport"
#   feature of the Postfix local delivery agent (see local(8)).
#
# Details are described in the LOCAL_RECIPIENT_README file.
#
# Beware: if the Postfix SMTP server runs chrooted, you probably have
# to access the passwd file via the proxymap service, in order to
# overcome chroot restrictions. The alternative, having a copy of
# the system passwd file in the chroot jail is just not practical.
#
# The right-hand side of the lookup tables is conveniently ignored.
# In the left-hand side, specify a bare username, an @domain.tld
# wild-card, or specify a user@domain.tld address.
#
#local_recipient_maps = unix:passwd.byname $alias_maps
#local_recipient_maps = proxy:unix:passwd.byname $alias_maps
#local_recipient_maps =

# The unknown_local_recipient_reject_code specifies the SMTP server
# response code when a recipient domain matches $mydestination or
# ${proxy,inet}_interfaces, while $local_recipient_maps is non-empty
# and the recipient address or address local-part is not found.
#
# The default setting is 550 (reject mail) but it is safer to start
# with 450 (try again later) until you are certain that your
# local_recipient_maps settings are OK.
#
unknown_local_recipient_reject_code = 550

# TRUST AND RELAY CONTROL

# The mynetworks parameter specifies the list of "trusted" SMTP
# clients that have more privileges than "strangers".
#
# In particular, "trusted" SMTP clients are allowed to relay mail
# through Postfix.  See the smtpd_recipient_restrictions parameter
# in postconf(5).
#
# You can specify the list of "trusted" network addresses by hand
# or you can let Postfix do it for you (which is the default).
#
# By default (mynetworks_style = subnet), Postfix "trusts" SMTP
# clients in the same IP subnetworks as the local machine.
# On Linux, this does works correctly only with interfaces specified
# with the "ifconfig" command.
#
# Specify "mynetworks_style = class" when Postfix should "trust" SMTP
# clients in the same IP class A/B/C networks as the local machine.
# Don't do this with a dialup site - it would cause Postfix to "trust"
# your entire provider's network.  Instead, specify an explicit
# mynetworks list by hand, as described below.
#
# Specify "mynetworks_style = host" when Postfix should "trust"
# only the local machine.
#
#mynetworks_style = class
#mynetworks_style = subnet
#mynetworks_style = host

# Alternatively, you can specify the mynetworks list by hand, in
# which case Postfix ignores the mynetworks_style setting.
#
# Specify an explicit list of network/netmask patterns, where the
# mask specifies the number of bits in the network part of a host
# address.
#
# You can also specify the absolute pathname of a pattern file instead
# of listing the patterns here. Specify type:table for table-based lookups
# (the value on the table right-hand side is not used).
#
#mynetworks = 192.168.101.3/24, 127.0.0.0/8
#mynetworks = $config_directory/mynetworks
#mynetworks = hash:/etc/postfix/network_table

# The relay_domains parameter restricts what destinations this system will
# relay mail to.  See the smtpd_recipient_restrictions description in
# postconf(5) for detailed information.
#
# By default, Postfix relays mail
# - from "trusted" clients (IP address matches $mynetworks) to any destination,
# - from "untrusted" clients to destinations that match $relay_domains or
#   subdomains thereof, except addresses with sender-specified routing.
# The default relay_domains value is $mydestination.
#
# In addition to the above, the Postfix SMTP server by default accepts mail
# that Postfix is final destination for:
# - destinations that match $inet_interfaces or $proxy_interfaces,
# - destinations that match $mydestination
# - destinations that match $virtual_alias_domains,
# - destinations that match $virtual_mailbox_domains.
# These destinations do not need to be listed in $relay_domains.
#
# Specify a list of hosts or domains, /file/name patterns or type:name
# lookup tables, separated by commas and/or whitespace.  Continue
# long lines by starting the next line with whitespace. A file name
# is replaced by its contents; a type:name table is matched when a
# (parent) domain appears as lookup key.
#
# NOTE: Postfix will not automatically forward mail for domains that
# list this system as their primary or backup MX host. See the
# permit_mx_backup restriction description in postconf(5).
#
#relay_domains = $mydestination

# INTERNET OR INTRANET

# The relayhost parameter specifies the default host to send mail to
# when no entry is matched in the optional transport(5) table. When
# no relayhost is given, mail is routed directly to the destination.
#
# On an intranet, specify the organizational domain name. If your
# internal DNS uses no MX records, specify the name of the intranet
# gateway host instead.
#
# In the case of SMTP, specify a domain, host, host:port, [host]:port,
# [address] or [address]:port; the form [host] turns off MX lookups.
#
# If you're connected via UUCP, see also the default_transport parameter.
#
#relayhost = $mydomain
#relayhost = [gateway.my.domain]
#relayhost = [mailserver.isp.tld]
#relayhost = uucphost
#relayhost = [an.ip.add.ress]

# REJECTING UNKNOWN RELAY USERS
#
# The relay_recipient_maps parameter specifies optional lookup tables
# with all addresses in the domains that match $relay_domains.
#
# If this parameter is defined, then the SMTP server will reject
# mail for unknown relay users. This feature is off by default.
#
# The right-hand side of the lookup tables is conveniently ignored.
# In the left-hand side, specify an @domain.tld wild-card, or specify
# a user@domain.tld address.
#
#relay_recipient_maps = hash:/etc/postfix/relay_recipients

# INPUT RATE CONTROL
#
# The in_flow_delay configuration parameter implements mail input
# flow control. This feature is turned on by default, although it
# still needs further development (it's disabled on SCO UNIX due
# to an SCO bug).
#
# A Postfix process will pause for $in_flow_delay seconds before
# accepting a new message, when the message arrival rate exceeds the
# message delivery rate. With the default 100 SMTP server process
# limit, this limits the mail inflow to 100 messages a second more
# than the number of messages delivered per second.
#
# Specify 0 to disable the feature. Valid delays are 0..10.
#
#in_flow_delay = 1s

# ADDRESS REWRITING
#
# The ADDRESS_REWRITING_README document gives information about
# address masquerading or other forms of address rewriting including
# username->Firstname.Lastname mapping.

# ADDRESS REDIRECTION (VIRTUAL DOMAIN)
#
# The VIRTUAL_README document gives information about the many forms
# of domain hosting that Postfix supports.

# "USER HAS MOVED" BOUNCE MESSAGES
#
# See the discussion in the ADDRESS_REWRITING_README document.

# TRANSPORT MAP
#
# See the discussion in the ADDRESS_REWRITING_README document.

# ALIAS DATABASE
#
```

```
# The alias_maps parameter specifies the list of alias databases used
# by the local delivery agent. The default list is system dependent.
#
# On systems with NIS, the default is to search the local alias
# database, then the NIS alias database. See aliases(5) for syntax
# details.
#
# If you change the alias database, run "postalias /etc/aliases" (or
# wherever your system stores the mail alias file), or simply run
# "newaliases" to build the necessary DBM or DB file.
#
# It will take a minute or so before changes become visible.  Use
# "postfix reload" to eliminate the delay.
#
#alias_maps = dbm:/etc/aliases
#alias_maps = hash:/etc/aliases
#alias_maps = hash:/etc/aliases, nis:mail.aliases
#alias_maps = netinfo:/aliases

# The alias_database parameter specifies the alias database(s) that
# are built with "newaliases" or "sendmail -bi".  This is a separate
# configuration parameter, because alias_maps (see above) may specify
# tables that are not necessarily all under control by Postfix.
#
#alias_database = dbm:/etc/aliases
#alias_database = dbm:/etc/mail/aliases
#alias_database = hash:/etc/aliases
#alias_database = hash:/etc/aliases, hash:/opt/majordomo/aliases

# ADDRESS EXTENSIONS (e.g., user+foo)
#
# The recipient_delimiter parameter specifies the separator between
# user names and address extensions (user+foo). See canonical(5),
# local(8), relocated(5) and virtual(5) for the effects this has on
# aliases, canonical, virtual, relocated and .forward file lookups.
# Basically, the software tries user+foo and .forward+foo before
# trying user and .forward.
#
#recipient_delimiter = +

# DELIVERY TO MAILBOX
#
# The home_mailbox parameter specifies the optional pathname of a
# mailbox file relative to a user's home directory. The default
# mailbox file is /var/spool/mail/user or /var/mail/user.  Specify
# "Maildir/" for qmail-style delivery (the / is required).
```

```
#
#home_mailbox = Mailbox
#home_mailbox = Maildir/

# The mail_spool_directory parameter specifies the directory where
# UNIX-style mailboxes are kept. The default setting depends on the
# system type.
#
#mail_spool_directory = /var/mail
#mail_spool_directory = /var/spool/mail

# The mailbox_command parameter specifies the optional external
# command to use instead of mailbox delivery. The command is run as
# the recipient with proper HOME, SHELL and LOGNAME environment settings.
# Exception:  delivery for root is done as $default_user.
#
# Other environment variables of interest: USER (recipient username),
# EXTENSION (address extension), DOMAIN (domain part of address),
# and LOCAL (the address localpart).
#
# Unlike other Postfix configuration parameters, the mailbox_command
# parameter is not subjected to $parameter substitutions. This is to
# make it easier to specify shell syntax (see example below).
#
# Avoid shell meta characters because they will force Postfix to run
# an expensive shell process. Procmail alone is expensive enough.
#
# IF YOU USE THIS TO DELIVER MAIL SYSTEM-WIDE, YOU MUST SET UP AN
# ALIAS THAT FORWARDS MAIL FOR ROOT TO A REAL USER.
#
#mailbox_command = /some/where/procmail
#mailbox_command = /some/where/procmail -a "$EXTENSION"

# The mailbox_transport specifies the optional transport in master.cf
# to use after processing aliases and .forward files. This parameter
# has precedence over the mailbox_command, fallback_transport and
# luser_relay parameters.
#
# Specify a string of the form transport:nexthop, where transport is
# the name of a mail delivery transport defined in master.cf.  The
# :nexthop part is optional. For more details see the sample transport
# configuration file.
#
# NOTE: if you use this feature for accounts not in the UNIX password
# file, then you must update the "local_recipient_maps" setting in
# the main.cf file, otherwise the SMTP server will reject mail for
# non-UNIX accounts with "User unknown in local recipient table".
#
#mailbox_transport = lmtp:unix:/file/name
#mailbox_transport = cyrus

# The fallback_transport specifies the optional transport in master.cf
# to use for recipients that are not found in the UNIX passwd database.
# This parameter has precedence over the luser_relay parameter.
#
# Specify a string of the form transport:nexthop, where transport is
# the name of a mail delivery transport defined in master.cf.  The
# :nexthop part is optional. For more details see the sample transport
# configuration file.
#
# NOTE: if you use this feature for accounts not in the UNIX password
# file, then you must update the "local_recipient_maps" setting in
# the main.cf file, otherwise the SMTP server will reject mail for
# non-UNIX accounts with "User unknown in local recipient table".
#
#fallback_transport = lmtp:unix:/file/name
#fallback_transport = cyrus
#fallback_transport =

# The luser_relay parameter specifies an optional destination address
# for unknown recipients.  By default, mail for unknown@$mydestination,
# unknown@[$inet_interfaces] or unknown@[$proxy_interfaces] is returned
# as undeliverable.
#
# The following expansions are done on luser_relay: $user (recipient
# username), $shell (recipient shell), $home (recipient home directory),
# $recipient (full recipient address), $extension (recipient address
# extension), $domain (recipient domain), $local (entire recipient
# localpart), $recipient_delimiter. Specify ${name?value} or
# ${name:value} to expand value only when $name does (does not) exist.
#
# luser_relay works only for the default Postfix local delivery agent.
#
# NOTE: if you use this feature for accounts not in the UNIX password
# file, then you must specify "local_recipient_maps =" (i.e. empty) in
# the main.cf file, otherwise the SMTP server will reject mail for
# non-UNIX accounts with "User unknown in local recipient table".
#
#luser_relay = $user@other.host
#luser_relay = $local@other.host
#luser_relay = admin+$local
```

```
# JUNK MAIL CONTROLS
#
# The controls listed here are only a very small subset. The file
# SMTPD_ACCESS_README provides an overview.

# The header_checks parameter specifies an optional table with patterns
# that each logical message header is matched against, including
# headers that span multiple physical lines.
#
# By default, these patterns also apply to MIME headers and to the
# headers of attached messages. With older Postfix versions, MIME and
# attached message headers were treated as body text.
#
# For details, see "man header_checks".
#
#header_checks = regexp:/etc/postfix/header_checks

# FAST ETRN SERVICE
#
# Postfix maintains per-destination logfiles with information about
# deferred mail, so that mail can be flushed quickly with the SMTP
# "ETRN domain.tld" command, or by executing "sendmail -qRdomain.tld".
# See the ETRN_README document for a detailed description.
#
# The fast_flush_domains parameter controls what destinations are
# eligible for this service. By default, they are all domains that
# this server is willing to relay mail to.
#
#fast_flush_domains = $relay_domains

# SHOW SOFTWARE VERSION OR NOT
#
# The smtpd_banner parameter specifies the text that follows the 220
# code in the SMTP server's greeting banner. Some people like to see
# the mail version advertised. By default, Postfix shows no version.
#
# You MUST specify $myhostname at the start of the text. That is an
# RFC requirement. Postfix itself does not care.
#
#smtpd_banner = $myhostname ESMTP $mail_name
#smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)

# PARALLEL DELIVERY TO THE SAME DESTINATION
#
# How many parallel deliveries to the same user or domain? With local
# delivery, it does not make sense to do massively parallel delivery
# to the same user, because mailbox updates must happen sequentially,
# and expensive pipelines in .forward files can cause disasters when
# too many are run at the same time. With SMTP deliveries, 10
# simultaneous connections to the same domain could be sufficient to
# raise eyebrows.
#
# Each message delivery transport has its XXX_destination_concurrency_limit
# parameter.  The default is $default_destination_concurrency_limit for
# most delivery transports. For the local delivery agent the default is 2.

#local_destination_concurrency_limit = 2
#default_destination_concurrency_limit = 20

# DEBUGGING CONTROL
#
# The debug_peer_level parameter specifies the increment in verbose
# logging level when an SMTP client or server host name or address
# matches a pattern in the debug_peer_list parameter.
#
debug_peer_level = 2

# The debug_peer_list parameter specifies an optional list of domain
# or network patterns, /file/name patterns or type:name tables. When
# an SMTP client or server host name or address matches a pattern,
# increase the verbose logging level by the amount specified in the
# debug_peer_level parameter.
#
#debug_peer_list = 127.0.0.1
#debug_peer_list = some.domain

# The debugger_command specifies the external command that is executed
# when a Postfix daemon program is run with the -D option.
#
# Use "command .. & sleep 5" so that the debugger can attach before
# the process marches on. If you use an X-based debugger, be sure to
# set up your XAUTHORITY environment variable before starting Postfix.
#
debugger_command =
         PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
         xxgdb $daemon_directory/$process_name $process_id & sleep 5

# If you can't use X, use this to capture the call stack when a
# daemon crashes. The result is in a file in the configuration
# directory, and is named after the process name and the process ID.
#
# debugger_command =
#         PATH=/bin:/usr/bin:/usr/local/bin; export PATH; (echo cont;
#         echo where) | gdb $daemon_directory/$process_name $process_id 2>&1
#         >$config_directory/$process_name.$process_id.log & sleep 5
#
# Another possibility is to run gdb under a detached screen session.
# To attach to the screen sesssion, su root and run "screen -r
# <id_string>" where <id_string> uniquely matches one of the detached
# sessions (from "screen -list").
#
# debugger_command =
#         PATH=/bin:/usr/bin:/sbin:/usr/sbin; export PATH; screen
#         -dmS $process_name gdb $daemon_directory/$process_name
#         $process_id & sleep 1

# INSTALL-TIME CONFIGURATION INFORMATION
#
# The following parameters are used when installing a new Postfix version.
#
# sendmail_path: The full pathname of the Postfix sendmail command.
# This is the Sendmail-compatible mail posting interface.
#
sendmail_path = /usr/sbin/sendmail

# newaliases_path: The full pathname of the Postfix newaliases command.
# This is the Sendmail-compatible command to build alias databases.
#
newaliases_path = /usr/bin/newaliases

# mailq_path: The full pathname of the Postfix mailq command.  This
# is the Sendmail-compatible mail queue listing command.
#
mailq_path = /usr/bin/mailq

# setgid_group: The group for mail submission and queue management
# commands.  This must be a group name with a numerical group ID that
# is not shared with other accounts, not even with the Postfix account.
#
setgid_group = postdrop

# html_directory: The location of the Postfix HTML documentation.
#
html_directory = no

# manpage_directory: The location of the Postfix on-line manual pages.
#
manpage_directory = /usr/local/man
```

```
# sample_directory: The location of the Postfix sample configuration files.
# This parameter is obsolete as of Postfix 2.1.
#
sample_directory = /etc/postfix

# readme_directory: The location of the Postfix README files.
#
readme_directory = no
virtual_mailbox_domains = ceylonlinux.com,ceygate.com
```

Here is our dovecot.conf

```
## Dovecot configuration file

# If you're in a hurry, see http://wiki.dovecot.org/QuickConfiguration

# '#' character and everything after it is treated as comments. Extra spaces
# and tabs are ignored. If you want to use either of these explicitly, put the
# value inside quotes, eg.: key = "# char and trailing whitespace  "

# Default values are shown after each value, it's not required to uncomment
# any of the lines. Exception to this are paths, they're just examples
# with real defaults being based on configure options. The paths listed here
# are for configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var
# --with-ssldir=/etc/ssl

# Base directory where to store runtime data.
#####################uncomented###############################
base_dir = /var/run/dovecot/

# Protocols we want to be serving: imap imaps pop3 pop3s
#protocols = imap imaps
########################insert pop3 protocol#######################
protocols = pop3




# IP or host address where to listen in for connections. It's not currently
# possible to specify multiple addresses. "*" listens in all IPv4 interfaces.
# "[::]" listens in all IPv6 interfaces, but may also listen in all IPv4
# interfaces depending on the operating system.
#
# If you want to specify ports for each service, you will need to configure
# these settings inside the protocol imap/pop3 { ... } section, so you can
# specify different ports for IMAP/POP3. For example:
#   protocol imap {
#     listen = *:10143
#     ssl_listen = *:10943
#     ..
#   }
#   protocol pop3 {
#     listen = *:10100
#     ..
#   }
#listen = *

# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that if the remote IP
# matches the local IP (ie. you're connecting from the same computer), the
# connection is considered secure and plaintext authentication is allowed.
########################enable plain text authentication################
disable_plaintext_auth = no

# Should all IMAP and POP3 processes be killed when Dovecot master process
# shuts down. Setting this to "no" means that Dovecot can be upgraded without
# forcing existing client connections to close (although that could also be
# a problem if the upgrade is eg. because of a security fix). This however
# means that after master process has died, the client processes can't write
# to log files anymore.
#shutdown_clients = yes

##
## Logging
##

# Use this logfile instead of syslog(). /dev/stderr can be used if you want to
# use stderr for logging (ONLY /dev/stderr - otherwise it is closed).
#log_path =

# For informational messages, use this logfile instead of the default
#info_log_path =

# Prefix for each line written to log file. % codes are in strftime(3)
# format.
#log_timestamp = "%b %d %H:%M:%S "

# Syslog facility to use if you're logging to syslog. Usually if you don't
# want to use "mail", you'll use local0..local7. Also other standard
# facilities are supported.
```

```
#syslog_facility = mail

##
## SSL settings
##

# IP or host address where to listen in for SSL connections. Defaults
# to above if not specified.
#ssl_listen =

# Disable SSL/TLS support.
#############################disable ssl##############################
ssl_disable = yes

# PEM encoded X.509 SSL/TLS certificate and private key. They're opened before
# dropping root privileges, so keep the key file unreadable by anyone but
# root. Included doc/mkcert.sh can be used to easily generate self-signed
# certificate, just make sure to update the domains in dovecot-openssl.cnf
#ssl_cert_file = /etc/ssl/certs/dovecot.pem
#ssl_key_file = /etc/ssl/private/dovecot.pem

# If key file is password protected, give the password here. Alternatively
# give it when starting dovecot with -p parameter.
#ssl_key_password =

# File containing trusted SSL certificate authorities. Usually not needed.
#ssl_ca_file =

# Request client to send a certificate.
#ssl_verify_client_cert = no

# How often to regenerate the SSL parameters file. Generation is quite CPU
# intensive operation. The value is in hours, 0 disables regeneration
# entirely.
#ssl_parameters_regenerate = 168

# SSL ciphers to use
#ssl_cipher_list = ALL:!LOW

# Show protocol level SSL errors.
#verbose_ssl = no

##
## Login processes
##
```

```
# Directory where authentication process places authentication UNIX sockets
# which login needs to be able to connect to. The sockets are created when
# running as root, so you don't have to worry about permissions. Note that
# everything in this directory is deleted when Dovecot is started.
#####################uncomented########################################
login_dir = /var/run/dovecot/login

# chroot login process to the login_dir. Only reason not to do this is if you
# wish to run the whole Dovecot without roots.
# http://wiki.dovecot.org/Rootless
#login_chroot = yes

# User to use for the login process. Create a completely new user for this,
# and don't use it anywhere else. The user must also belong to a group where
# only it has access, it's used to control access for authentication process.
# Note that this user is NOT used to access mails.
# http://wiki.dovecot.org/UserIds
#############################uncomented#################################
login_user = dovecot

# Set max. process size in megabytes. If you don't use
# login_process_per_connection you might need to grow this.
#login_process_size = 32

# Should each login be processed in it's own process (yes), or should one
# login process be allowed to process multiple connections (no)? Yes is more
# secure, espcially with SSL/TLS enabled. No is faster since there's no need
# to create processes all the time.
#login_process_per_connection = yes

# Number of login processes to create. If login_process_per_connection is
# yes, this is the number of extra processes waiting for users to log in.
#login_processes_count = 3

# Maximum number of extra login processes to create. The extra process count
# usually stays at login_processes_count, but when multiple users start logging
# in at the same time more extra processes are created. To prevent fork-bombing
# we check only once in a second if new processes should be created - if all
# of them are used at the time, we double their amount until limit set by this
# setting is reached. This setting is used only if
# login_process_per_connection is yes.
#login_max_processes_count = 128

# Maximum number of connections allowed in login state. When this limit is
# reached, the oldest connections are dropped. If login_process_per_connection
# is no, this is a per-process value, so the absolute maximum number of users
# logging in actually login_processes_count * max_logging_users.
#login_max_logging_users = 256

# Greeting message for clients.
##############################uncomented###########################
login_greeting = Dovecot ready.

# Space-separated list of elements we want to log. The elements which have
# a non-empty variable value are joined together to form a comma-separated
# string.
#login_log_format_elements = user=<%u> method=%m rip=%r lip=%l %c

# Login log format. %$ contains login_log_format_elements string, %s contains
# the data we want to log.
#login_log_format = %$: %s

##
## Mailbox locations and namespaces
##

# Default MAIL environment to use when it's not set. By leaving this empty
# dovecot tries to do some automatic detection as described in
# doc/mail-storages.txt. There are a few special variables you can use, eg.:
#
#   %u - username
#   %n - user part in user@domain, same as %u if there's no domain
#   %d - domain part in user@domain, empty if there's no domain
#   %h - home directory
#
# See doc/variables.txt for full list. Some examples:
#
#   default_mail_env = maildir:/var/mail/%1u/%u/Maildir
#   default_mail_env = mbox:~/mail/:INBOX=/var/mail/%u
#   default_mail_env = mbox:/var/mail/%d/%n/:INDEX=/var/indexes/%d/%n
#
##################add maildir format for system accounts##########
default_mail_env = maildir:/home/%u/Maildir

# If you need to set multiple mailbox locations or want to change default
# namespace settings, you can do it by defining namespace sections:
#
# You can have private, shared and public namespaces. The only difference
# between them is how Dovecot announces them to client via NAMESPACE
# extension. Shared namespaces are meant for user-owned mailboxes which are
# shared to other users, while public namespaces are for more globally
# accessible mailboxes.
```

```
#
# REMEMBER: If you add any namespaces, the default namespace must be added
# explicitly, ie. default_mail_env does nothing unless you have a namespace
# without a location setting. Default namespace is simply done by having a
# namespace with empty prefix.
#namespace private {
  # Hierarchy separator to use. You should use the same separator for all
  # namespaces or some clients get confused. '/' is usually a good one.
  #separator = /

  # Prefix required to access this namespace. This needs to be different for
  # all namespaces. For example "Public/".
  #prefix =

  # Physical location of the mailbox. This is in same format as
  # default_mail_env, which is also the default for it.
  #location =

  # There can be only one INBOX, and this setting defines which namespace
  # has it.
  #inbox = yes

  # If namespace is hidden, it's not advertised to clients via NAMESPACE
  # extension or shown in LIST replies. This is mostly useful when converting
  # from another server with different namespaces which you want to depricate
  # but still keep working. For example you can create hidden namespaces with
  # prefixes "~/mail/", "~%u/mail/" and "mail/".
  #hidden = yes
#}

# Grant access to these extra groups for mail processes. Typical use would be
# to give "mail" group write access to /var/mail to be able to create dotlocks.
#mail_extra_groups = dovecot

# Allow full filesystem access to clients. There's no access checks other than
# what the operating system does for the active UID/GID. It works with both
# maildir and mboxes, allowing you to prefix mailboxes names with eg. /path/
# or ~user/.
#mail_full_filesystem_access = no

##
## Mail processes
##

# Enable mail process debugging. This can help you figure out why Dovecot
# isn't finding your mails.
```

```
#####################uncomented####################################
mail_debug = no

# Log prefix for mail processes. See doc/variables.txt for list of possible
# variables you can use.
####################uncomented###################################
mail_log_prefix = "%Us(%u): "

# Use mmap() instead of read() to read mail files. read() seems to be a bit
# faster with my Linux/x86 and it's better with NFS, so that's the default.
# Note that OpenBSD 3.3 and older don't work right with mail_read_mmaped = yes.
#mail_read_mmaped = no

# Don't use mmap() at all. This is required if you store indexes to shared
# filesystems (NFS or clustered filesystem).
#mmap_disable = no

# Don't write() to mmaped files. This is required for some operating systems
# which use separate caches for them, such as OpenBSD.
#mmap_no_write = no

# Locking method for index files. Alternatives are fcntl, flock and dotlock.
# Dotlocking uses some tricks which may create more disk I/O than other locking
# methods. NOTE: If you use NFS, remember to change also mmap_disable setting!
#lock_method = fcntl

# Drop all privileges before exec()ing the mail process. This is mostly
# meant for debugging, otherwise you don't get core dumps. It could be a small
# security risk if you use single UID for multiple users, as the users could
# ptrace() each others processes then.
#mail_drop_priv_before_exec = no

# Show more verbose process titles (in ps). Currently shows user name and
# IP address. Useful for seeing who are actually using the IMAP processes
# (eg. shared mailboxes or if same uid is used for multiple accounts).
#verbose_proctitle = no

# Valid UID range for users, defaults to 500 and above. This is mostly
# to make sure that users can't log in as daemons or other system users.
# Note that denying root logins is hardcoded to dovecot binary and can't
# be done even if first_valid_uid is set to 0.
#first_valid_uid = 500
#last_valid_uid = 0

# Valid GID range for users, defaults to non-root/wheel. Users having
# non-valid GID as primary group ID aren't allowed to log in. If user
# belongs to supplementary groups with non-valid GIDs, those groups are
# not set.
#first_valid_gid = 1
#last_valid_gid = 0

# Maximum number of running mail processes. When this limit is reached,
# new users aren't allowed to log in.
#max_mail_processes = 1024

# Set max. process size in megabytes. Most of the memory goes to mmap()ing
# files, so it shouldn't harm much even if this limit is set pretty high.
#mail_process_size = 256

# Maximum allowed length for mail keyword name. It's only forced when trying
# to create new keywords.
#mail_max_keyword_length = 50

# Default umask to use for mail files and directories.
#umask = 0077

# ':' separated list of directories under which chrooting is allowed for mail
# processes (ie. /var/mail will allow chrooting to /var/mail/foo/bar too).
# This setting doesn't affect login_chroot or auth_chroot variables.
# WARNING: Never add directories here which local users can modify, that
# may lead to root exploit. Usually this should be done only if you don't
# allow shell access for users. See doc/configuration.txt for more information.
#valid_chroot_dirs =

# Default chroot directory for mail processes. This can be overridden for
# specific users in user database by giving /./ in user's home directory
# (eg. /home/./user chroots into /home). Note that usually there is no real
# need to do chrooting, Dovecot doesn't allow users to access files outside
# their mail directory anyway.
#mail_chroot =

##
## Mailbox handling optimizations
##

# Space-separated list of fields to initially save into cache file. Currently
# these fields are allowed:
#
#  flags, date.sent, date.received, size.virtual, size.physical
#  mime.parts, imap.body, imap.bodystructure
#
# Different IMAP clients work in different ways, so they benefit from
# different cached fields. Some do not benefit from them at all. Caching more
# than necessary generates useless disk I/O, so you don't want to do that
# either.
#
# Dovecot attempts to automatically figure out what client wants and it keeps
# only that. However the first few times a mailbox is opened, Dovecot hasn't
# yet figured out what client needs, so it may not perform optimally. If you
# know what fields the majority of your clients need, it may be useful to set
# these fields by hand. If client doesn't actually use them, Dovecot will
# eventually drop them.
#
# Usually you should just leave this field alone. The potential benefits are
# typically unnoticeable.
#mail_cache_fields =

# Space-separated list of fields that Dovecot should never save to cache file.
# Useful if you want to save disk space at the cost of more I/O when the fields
# needed.
#mail_never_cache_fields =

# The minimum number of mails in a mailbox before updates are done to cache
# file. This allows optimizing Dovecot's behavior to do less disk writes at
# the cost of more disk reads.
#mail_cache_min_mail_count = 0

# When IDLE command is running, mailbox is checked once in a while to see if
# there are any new mails or other changes. This setting defines the minimum
# time to wait between those checks. Dovecot is however able to use dnotify
# and inotify with Linux to reply immediately after the change occurs.
#mailbox_idle_check_interval = 30

# Save mails with CR+LF instead of plain LF. This makes sending those mails
# take less CPU, especially with sendfile() syscall with Linux and FreeBSD.
# But it also creates a bit more disk I/O which may just make it slower.
# Also note that if other software reads the mboxes/maildirs, they may handle
# the extra CRs wrong and cause problems.
#mail_save_crlf = no

##
## Maildir-specific settings
##

# By default LIST command returns all entries in maildir beginning with dot.
# Enabling this option makes Dovecot return only entries which are directories.
# This is done by stat()ing each entry, so it causes more disk I/O.
# (For systems setting struct dirent->d_type, this check is free and it's
# done always regardless of this setting)
#maildir_stat_dirs = no

# Copy mail to another folders using hard links. This is much faster than
# actually copying the file. This is problematic only if something modifies
# the mail in one folder but doesn't want it modified in the others. I don't
# know any MUA which would modify mail files directly. IMAP protocol also
# requires that the mails don't change, so it would be problematic in any case.
# If you care about performance, enable it.
################uncomented and set as yes##########################
maildir_copy_with_hardlinks = yes

##
## mbox-specific settings
##

# Which locking methods to use for locking mbox. There are four available:
#  dotlock: Create <mailbox>.lock file. This is the oldest and most NFS-safe
#           solution. If you want to use /var/mail/ like directory, the users
#           will need write access to that directory.
#  fcntl  : Use this if possible. Works with NFS too if lockd is used.
#  flock  : May not exist in all systems. Doesn't work with NFS.
#  lockf  : May not exist in all systems. Doesn't work with NFS.
#
# You can use multiple locking methods; if you do the order they're declared
# in is important to avoid deadlocks if other MTAs/MUAs are using multiple
# locking methods as well. Some operating systems don't allow using some of
# them simultaneously.
#mbox_read_locks = fcntl
#mbox_write_locks = dotlock fcntl

# Maximum time in seconds to wait for lock (all of them) before aborting.
#mbox_lock_timeout = 300

# If dotlock exists but the mailbox isn't modified in any way, override the
# lock file after this many seconds.
#mbox_dotlock_change_timeout = 120

# When mbox changes unexpectedly we have to fully read it to find out what
# changed. If the mbox is large this can take a long time. Since the change
# is usually just a newly appended mail, it'd be faster to simply read the
# new mails. If this setting is enabled, Dovecot does this but still safely
# fallbacks to re-reading the whole mbox file whenever something in mbox isn't
# how it's expected to be. The only real downside to this setting is that if
# some other MUA changes message flags, Dovecot doesn't notice it immediately.
# Note that a full sync is done with SELECT, EXAMINE, EXPUNGE and CHECK
# commands.
#mbox_dirty_syncs = yes

# Like mbox_dirty_syncs, but don't do full syncs even with SELECT, EXAMINE,
# EXPUNGE or CHECK commands. If this is set, mbox_dirty_syncs is ignored.
#mbox_very_dirty_syncs = no

# Delay writing mbox headers until doing a full write sync (EXPUNGE and CHECK
# commands and when closing the mailbox). This is especially useful for POP3
# where clients often delete all mails. The downside is that our changes
# aren't immediately visible to other MUAs.
#mbox_lazy_writes = yes

# If mbox size is smaller than this (in kilobytes), don't write index files.
# If an index file already exists it's still read, just not updated.
#mbox_min_index_size = 0

##
## dbox-specific settings
##

# Maximum dbox file size in kilobytes until it's rotated.
#dbox_rotate_size = 2048

# Minimum dbox file size in kilobytes before it's rotated
# (overrides dbox_rotate_days)
#dbox_rotate_min_size = 16

# Maximum dbox file age in days until it's rotated. Day always begins from
# midnight, so 1 = today, 2 = yesterday, etc. 0 = check disabled.
#dbox_rotate_days = 0

##
## IMAP specific settings
##

protocol imap {
  # Login executable location.
  #login_executable = /usr/libexec/dovecot/imap-login

  # IMAP executable location. Changing this allows you to execute other
  # binaries before the imap process is executed.
  #
  # This would write rawlogs into ~/dovecot.rawlog/ directory:
  #   mail_executable = /usr/libexec/dovecot/rawlog /usr/libexec/dovecot/imap
  #
```

```
# This would attach gdb into the imap process and write backtraces into
# /tmp/gdbhelper.* files:
#   mail_executable = /usr/libexec/dovecot/gdbhelper /usr/libexec/dovecot/imap
#
#mail_executable = /usr/libexec/dovecot/imap

# Maximum IMAP command line length in bytes. Some clients generate very long
# command lines with huge mailboxes, so you may need to raise this if you get
# "Too long argument" or "IMAP command line too large" errors often.
#imap_max_line_length = 65536

# Support for dynamically loadable plugins. mail_plugins is a space separated
# list of plugins to load.
#mail_plugins =
#mail_plugin_dir = /usr/lib/dovecot/imap

# Send IMAP capabilities in greeting message. This makes it unnecessary for
# clients to request it with CAPABILITY command, so it saves one round-trip.
# Many clients however don't understand it and ask the CAPABILITY anyway.
#login_greeting_capability = no

# Workarounds for various client bugs:
#   delay-newmail:
#     Send EXISTS/RECENT new mail notifications only when replying to NOOP
#     and CHECK commands. Some clients ignore them otherwise, for example
#     OSX Mail. Outlook Express breaks more badly though, without this it
#     may show user "Message no longer in server" errors. Note that OE6 still
#     breaks even with this workaround if synchronization is set to
#     "Headers Only".
#   outlook-idle:
#     Outlook and Outlook Express never abort IDLE command, so if no mail
#     arrives in half a hour, Dovecot closes the connection. This is still
#     fine, except Outlook doesn't connect back so you don't see if new mail
#     arrives.
#   netscape-eoh:
#     Netscape 4.x breaks if message headers don't end with the empty "end of
#     headers" line. Normally all messages have this, but setting this
#     workaround makes sure that Netscape never breaks by adding the line if
#     it doesn't exist. This is done only for FETCH BODY[HEADER.FIELDS..]
#     commands. Note that RFC says this shouldn't be done.
#   tb-extra-mailbox-sep:
#     With mbox storage a mailbox can contain either mails or submailboxes,
#     but not both. Thunderbird separates these two by forcing server to
#     accept '/' suffix in mailbox names in subscriptions list.
# The list is space-separated.
#imap_client_workarounds = outlook-idle
```

```
}

##
## POP3 specific settings
##

protocol pop3 {
  # Login executable location.
########################uncomented###################################
  login_executable = /usr/local/libexec/dovecot/pop3-login

  # POP3 executable location
#########################uncomented############################
mail_executable = /usr/local/libexec/dovecot/pop3

  # Don't try to set mails non-recent or seen with POP3 sessions. This is
  # mostly intended to reduce disk I/O. With maildir it doesn't move files
  # from new/ to cur/, with mbox it doesn't write Status-header.
  #pop3_no_flag_updates = no

  # Support LAST command which exists in old POP3 specs, but has been removed
  # from new ones. Some clients still wish to use this though. Enabling this
  # makes RSET command clear all \Seen flags from messages.
  #pop3_enable_last = no

  # If mail has X-UIDL header, use it as the mail's UIDL.
  #pop3_reuse_xuidl = no

  # Keep the mailbox locked for the entire POP3 session.
  #pop3_lock_session = no

  # POP3 UIDL format to use. You can use following variables:
  #
  #  %v - Mailbox UIDVALIDITY
  #  %u - Mail UID
  #  %m - MD5 sum of the mailbox headers in hex (mbox only)
  #  %f - filename (maildir only)
  #
  # If you want UIDL compatibility with other POP3 servers, use:
  #  UW's ipop3d         : %08Xv%08Xu
  #  Courier version 0   : %f
  #  Courier version 1   : %u
  #  Courier version 2   : %v-%u
  #  Cyrus (<= 2.1.3)    : %u
  #  Cyrus (>= 2.1.4)    : %v.%u
  #  Older Dovecots      : %v.%u
  #  tpop3d             : %Mf
  #
  # Note that Outlook 2003 seems to have problems with %v.%u format which was
  # Dovecot's default, so if you're building a new server it would be a good
  # idea to change this. %08Xu%08Xv should be pretty fail-safe.
  #
  # NOTE: Nowadays this is required to be set explicitly, since the old
  # default was bad but it couldn't be changed without breaking existing
  # installations. %08Xu%08Xv will be the new default, so use it for new
  # installations.
  #
  pop3_uidl_format = %08Xv%08Xu

  # POP3 logout format string:
  #  %t - number of TOP commands
  #  %p - number of bytes sent to client as a result of TOP command
  #  %r - number of RETR commands
  #  %b - number of bytes sent to client as a result of RETR command
  #  %d - number of deleted messages
  #  %m - number of messages (before deletion)
  #  %s - mailbox size in bytes (before deletion)
  #pop3_logout_format = top=%t/%p, retr=%r/%b, del=%d/%m, size=%s

  # Support for dynamically loadable plugins. mail_plugins is a space separated
  # list of plugins to load.
  #mail_plugins =
########################uncomented##################################
mail_plugin_dir = /usr/lib/dovecot/pop3

  # Workarounds for various client bugs:
  #   outlook-no-nuls:
  #     Outlook and Outlook Express hang if mails contain NUL characters.
  #     This setting replaces them with 0x80 character.
  #   oe-ns-eoh:
  #     Outlook Express and Netscape Mail breaks if end of headers-line is
  #     missing. This option simply sends it if it's missing.
  # The list is space-separated.
  #pop3_client_workarounds =
}

##
## LDA specific settings
##

protocol lda {
  # Address to use when sending rejection mails.
  postmaster_address = postmaster@example.com

  # Hostname to use in various parts of sent mails, eg. in Message-Id.
  # Default is the system's real hostname.
  #hostname =

  # Support for dynamically loadable plugins. mail_plugins is a space separated
  # list of plugins to load.
  #mail_plugins =
  #mail_plugin_dir = /usr/lib/dovecot/lda

  # Binary to use for sending mails.
  #sendmail_path = /usr/lib/sendmail

  # UNIX socket path to master authentication server to find users.
  #auth_socket_path = /var/run/dovecot/auth-master
}

##
## Authentication processes
##

# Executable location
##########################uncomented#####################################
auth_executable = /usr/local/libexec/dovecot/dovecot-auth

# Set max. process size in megabytes.
##############################uncomented################################
auth_process_size = 256

# Authentication cache size in kilobytes. 0 means it's disabled.
# Note that bsdauth, PAM and vpopmail require cache_key to be set for caching
# to be used. Also note that currently auth cache doesn't work very well if
# you're using multiple passdbs with same usernames in them.
#auth_cache_size = 0
# Time to live in seconds for cached data. After this many seconds the cached
# record is no longer used, *except* if the main database lookup returns
# internal failure.
#auth_cache_ttl = 3600

# Space separated list of realms for SASL authentication mechanisms that need
# them. You can leave it empty if you don't want to support multiple realms.
# Many clients simply use the first one listed here, so keep the default realm
# first.
#auth_realms =
# Default realm/domain to use if none was specified. This is used for both
# SASL realms and appending @domain to username in plaintext logins.
#auth_default_realm =

# List of allowed characters in username. If the user-given username contains
# a character not listed in here, the login automatically fails. This is just
# an extra check to make sure user can't exploit any potential quote escaping
# vulnerabilities with SQL/LDAP databases. If you want to allow all characters,
# set this value to empty.
###############################uncomented###################################
#
auth_username_chars =
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890.-
_@

# Username character translations before it's looked up from databases. The
# value contains series of from -> to characters. For example "#@/@" means
# that '#' and '/' characters are translated to '@'.
#auth_username_translation =

# Username formatting before it's looked up from databases. You can use
# the standard variables here, eg. %Lu would lowercase the username, %n would
# drop away the domain if it was given, or "%n-AT-%d" would change the '@' into
# "-AT-". This translation is done after auth_username_translation changes.
#auth_username_format =

# If you want to allow master users to log in by specifying the master
# username within the normal username string (ie. not using SASL mechanism's
# support for it), you can specify the separator character here. The format
# is then <username><separator><master username>. UW-IMAP uses "*" as the
# separator, so that could be a good choice.
#auth_master_user_separator =

# Username to use for users logging in with ANONYMOUS SASL mechanism
#auth_anonymous_username = anonymous

# More verbose logging. Useful for figuring out why authentication isn't
# working.
###############################uncomented##################################
auth_verbose = yes

# Even more verbose logging for debugging purposes. Shows for example SQL
# queries.
#auth_debug = no

# In case of password mismatches, log the passwords and used scheme so the
# problem can be debugged. Requires auth_debug=yes to be set.
############################uncomentes and set to yes#####################
auth_debug_passwords = yes

# Maximum number of dovecot-auth worker processes. They're used to execute
# blocking passdb and userdb queries (eg. MySQL and PAM). They're
# automatically created and destroyed as needed.
#auth_worker_max_count = 30

# Kerberos keytab to use for the GSSAPI mechanism. Will use the system
# default (usually /etc/krb5.keytab) if not specified.
#auth_krb5_keytab =

auth default {
  # Space separated list of wanted authentication mechanisms:
  #   plain login digest-md5 cram-md5 ntlm rpa apop anonymous gssapi
  mechanisms = plain
  #
  # Password database is used to verify user's password (and nothing more).
  # You can have multiple passdbs and userdbs. This is useful if you want to
  # allow both system users (/etc/passwd) and virtual users to login without
  # duplicating the system users into virtual database.
  #
  # http://wiki.dovecot.org/Authentication
  #
  # By adding master=yes setting inside a passdb you make the passdb a list
  # of "master users", who can log in as anyone else. Unless you're using PAM,
  # you probably still want the destination user to be looked up from passdb
  # that it really exists. This can be done by adding pass=yes setting to the
  # master passdb.
  #
  # http://wiki.dovecot.org/MasterPassword

  # Users can be temporarily disabled by adding a passdb with deny=yes.
  # If the user is found from that database, authentication will fail.
  # The deny passdb should always be specified before others, so it gets
  # checked first. Here's an example:

  #passdb passwd-file {
    # File contains a list of usernames, one per line
    #args = /etc/dovecot.deny
    #deny = yes
  #}

  # PAM authentication. Preferred nowadays by most systems.
  # Note that PAM can only be used to verify if user's password is correct,
  # so it can't be used as userdb. If you don't want to use a separate user
  # database (passwd usually), you can use static userdb.
  # REMEMBER: You'll need /etc/pam.d/dovecot file created for PAM
  # authentication to actually work.
  passdb pam {
    #  [session=yes] [setcred=yes] [cache_key=<key>] [<service name>]
    #
    # session=yes makes Dovecot open and immediately close PAM session. Some
    # PAM plugins need this to work, such as pam_mkhomedir.
    #
    # setcred=yes makes Dovecot establish PAM credentials if some PAM plugins
    # need that. They aren't ever deleted though, so this isn't enabled by
    # default.
    #
    # cache_key can be used to enable authentication caching for PAM
    # (auth_cache_size also needs to be set). It isn't enabled by default
    # because PAM modules can do all kinds of checks besides checking password,
    # such as checking IP address. Dovecot can't know about these checks
    # without some help. cache_key is simply a list of variables (see
    # doc/variables.txt) which must match for the cached data to be used.
    # Here are some examples:
    #   %u - Username must match. Probably sufficient for most uses.
    #   %u%r - Username and remote IP address must match.
    #   %u%s - Username and service (ie. IMAP, POP3) must match.
    #
    # If service name is "*", it means the authenticating service name
    # is used, eg. pop3 or imap (/etc/pam.d/pop3, /etc/pam.d/imap).
    #
    # Some examples:
    #   args = session=yes *
    #   args = cache_key=%u dovecot
    #args = dovecot
  }

  # /etc/passwd or similar, using getpwnam()
  # In many systems nowadays this uses Name Service Switch, which is
  # configured in /etc/nsswitch.conf.



######################uncomented and args
added##########################
passdb passwd {
args = /etc/passwd
  }
```

```
  # /etc/shadow or similiar, using getspnam(). Deprecated by PAM nowadays.
  passdb shadow {
args = /etc/shadow
}

  # BSD authentication. Used by at least OpenBSD.
  #passdb bsdauth {
    # [cache_key=<key>] - See cache_key in PAM for explanation.
    #args =
  #}

  # passwd-like file with specified location
  #passdb passwd-file {
    # Path for passwd-file
    #args =
  #}

  # checkpassword executable authentication
  # NOTE: You will probably want to use "userdb prefetch" with this.
###############uncomented the whole block and /etc/passwd added#############
#passdb checkpassword {
    # Path for checkpassword binary
 #   args = /etc/passwd
 # }

  # SQL database
  #passdb sql {
    # Path for SQL configuration file, see doc/dovecot-sql.conf for example
    #args =
  #}

  # LDAP database
 # passdb ldap {
    # Path for LDAP configuration file, see doc/dovecot-ldap.conf for example
    #args =
 # }

  # vpopmail authentication
  #passdb vpopmail {
    # [cache_key=<key>] - See cache_key in PAM for explanation.
    #args =
  #}

  #
  # User database specifies where mails are located and what user/group IDs
  # own them. For single-UID configuration use "static".
  #
  # http://wiki.dovecot.org/Authentication
  # http://wiki.dovecot.org/VirtualUsers
  #

  # /etc/passwd or similar, using getpwnam()
  # In many systems nowadays this uses Name Service Switch, which is
  # configured in /etc/nsswitch.conf.
###################uncomented whole block and args added###############
  userdb passwd {
args = /etc/passwd
args = mail = maildir:/home/%u/Maildir
}

  # passwd-like file with specified location
  #userdb passwd-file {
    # Path for passwd-file
    #args =
  #}

  # static settings generated from template
  #userdb static {
    # Template for settings. Can return anything a userdb could normally
    # return, eg.: uid, gid, home, mail, nice
    #
    # A few examples:
    #
    #  args = uid=500 gid=500 home=/var/mail/%u
    #  args = uid=500 gid=500 home=/home/%u mail=mbox:/home/%u/mail nice=10
    #
    #args =
  #}

  # SQL database
  #userdb sql {
    # Path for SQL configuration file, see doc/dovecot-sql.conf for example
    #args =
  #}

  # LDAP database
  #userdb ldap {
    # Path for LDAP configuration file, see doc/dovecot-ldap.conf for example
    #args =
  #}

  # vpopmail
```

```
#userdb vpopmail {
#}

# "prefetch" user database means that the passdb already provided the
# needed information and there's no need to do a separate userdb lookup.
# This can be made to work with SQL and LDAP databases, see their example
# configuration files for more information how to do it.
# http://wiki.dovecot.org/AuthSpecials
#userdb prefetch {
#}

# User to use for the process. This user needs access to only user and
# password databases, nothing else. Only shadow and pam authentication
# requires roots, so use something else if possible. Note that passwd
# authentication with BSDs internally accesses shadow files, which also
# requires roots. Note that this user is NOT used to access mails.
# That user is specified by userdb above.
user = root

# Directory where to chroot the process. Most authentication backends don't
# work if this is set, and there's no point chrooting if auth_user is root.
# Note that valid_chroot_dirs isn't needed to use this setting.
#chroot =

# Number of authentication processes to create
#count = 1

# Require a valid SSL client certificate or the authentication fails.
#ssl_require_client_cert = no

# Take the username from client's SSL certificate, using X509_NAME_oneline()
# which typically uses subject's Distinguished Name.
#ssl_username_from_cert = no

# It's possible to export the authentication interface to other programs:
#socket listen {
  #master {
    # Master socket is typically used to give Dovecot's local delivery
    # agent access to userdb so it can find mailbox locations. It can
    # however also be used to disturb regular user authentications.
    # WARNING: Giving untrusted users access to master socket may be a
    # security risk, don't give too wide permissions to it!
    #path = /var/run/dovecot/auth-master
    #mode = 0600
    # Default user/group is the one who started dovecot-auth (root)
    #user =
      #group =
    #}
    #client {
      # The client socket is generally safe to export to everyone. Typical use
      # is to export it to your SMTP server so it can do SMTP AUTH lookups
      # using it.
      #path = /var/run/dovecot/auth-client
      #mode = 0660
    #}
  #}
}

# If you wish to use another authentication server than dovecot-auth, you can
# use connect sockets. They assumed to be already running, Dovecot's master
# process only tries to connect to them. They don't need any other settings
# than the path for the master socket, as the configuration is done elsewhere.
# Note that the client sockets must exist in the login_dir.
#auth external {
#  socket connect {
#    master {
#      path = /var/run/dovecot/auth-master
#    }
#  }
#}

##
## Dictionary server settings
##

# Dictionary can be used by some plugins to store key=value lists.
# Currently this is only used by dict quota backend. The dictionary can be
# used either directly or though a dictionary server. The following dict block
# maps dictionary names to URIs when the server is used. These can then be
# referenced using URIs in format "proxy:<name>".

dict {
  #quota = mysql:/etc/dovecot-dict-quota.conf
}

##
## Plugin settings
##

plugin {
  # Here you can give some extra environment variables to mail processes.
  # This is mostly meant for passing parameters to plugins. %variable
  # expansion is done for all values.

  # Quota plugin. Multiple backends are supported:
  #   dirsize: Find and sum all the files found from mail directory
  #   dict: Keep quota stored in dictionary (eg. SQL)
  #   maildir: Maildir++ quota
  #   fs: Read-only support for filesystem quota
  #quota = maildir

  # ACL plugin. vfile backend reads ACLs from "dovecot-acl" file from maildir
  # directory. You can also optionally give a global ACL directory path where
  # ACLs are applied to all users' mailboxes. The global ACL directory contains
  # one file for each mailbox, eg. INBOX or sub.mailbox.
  #acl = vfile:/etc/dovecot-acls

  # Convert plugin. If set, specifies the source storage path which is
  # converted to destination storage (default_mail_env).
  #convert_mail = mbox:%h/mail

  # Trash plugin. When saving a message would make user go over quota, this
  # plugin automatically deletes the oldest mails from configured mailboxes
  # until the message can be saved within quota limits. The configuration file
  # is a text file where each line is in format: <priority> <mailbox name>
  # Mails are first deleted in lowest -> highest priority number order
  #trash = /etc/dovecot-trash.conf
}
```