Setup basic mailserver with Postfix + Dovecot + Sieve for Virtualmail

.. on ubunut 18.04 server

pastebin...

source

very good guide that explains alot

```
apt install pwgen
# usefull to create safe passwords
apt install mariadb-server
# we need this to store our accounts and domains database
apt install postfix postfix-mysql
```

choose to configure postfix as **Internet Site** and enter fully qualified domain name of the server, the name entered here must not match any email domains you want to handle later (since we are configuring our server to process virtual mailboxes and not canonical domains).

```
apt install apache2 php7.2 swaks mutt certbot dovecot-mysql dovecot-pop3d
dovecot-imapd dovecot-managesieved dovecot-lmtpd adminer ca-certificates

rm /etc/apache2/sites-enabled/*

cat > /etc/apache2/sites-available/webmail-http.conf <<EOF
<VirtualHost *:80>
  ServerName mail.yourdomain.ch
  DocumentRoot /var/www/webmail/pub
</VirtualHost>
EOF
a2ensite webmail-http
systemctl reload apache2
mkdir -p /var/www/webmail/pub
chown www-data.www-data /var/www/webmail/pub
echo "it works" > /var/www/webmail/pub/test.txt
```

now try http://mail.yourdomain.ch/test.txt in a web-browser, it should show “it works!”

```
certbot certonly --webroot --webroot-path /var/www/webmail/pub -d
mail.yourdomain.ch
cat > /etc/apache2/sites-available/webmail-https.conf <<EOF
<VirtualHost *:443>
 ServerName mail.yourdomain.ch
 DocumentRoot /var/www/webmail/pub
 SSLEngine on
 SSLCertificateFile /etc/letsencrypt/live/mail.yourdomain.ch/fullchain.pem
```

```
 SSLCertificateKeyFile /etc/letsencrypt/live/mail.yourdomain.ch/privkey.pem
 Alias /adminer /usr/share/adminer/adminer
</VirtualHost>
EOF
a2ensite webmail-https
a2enmod ssl
systemctl restart apache2
```

to auto-forward all non-https traffic except for the certbot renewal stuff to https, add this to your `/etc/apache2/sites-available/webmail-http.conf` file before the closing `VirtualHost` tag:

```
RewriteEngine On
RewriteCond %{REQUEST_URI} !.well-known/acme-challenge
RewriteRule ^(.*)$ https://%{SERVER_NAME}/ [R=301,L]
```

```
a2enmod rewrite
systemctl restart apache2
```

to make sure letsencrypt will restart all our servers once the ssh keys change, we need to add this:

```
cat > /etc/letsencrypt/renewal-hooks/post/restart_services.sh <<EOF
#!/bin/bash
service postfix restart
service dovecot reload
service apache2 reload
EOF
chmod +x /etc/letsencrypt/renewal-hooks/post/restart_services.sh
```

setup mysql database:

create pw:

```
pwgen -s1 30 2
```

note down the two passwords and start the mysql client

```
mysql
```

in the mysql client console enter the following mysql commands:

```
  CREATE DATABASE mailserver;
  grant all on mailserver.* to 'mailadmin'@'localhost' identified by '<first
password goes here>';
  grant select on mailserver.* to 'mailserver'@'127.0.0.1' identified by
'<second password goes here>';
  USE mailserver;
  CREATE TABLE IF NOT EXISTS `domains` (
 `id` int(11) NOT NULL auto_increment,
```

```
 `name` varchar(50) NOT NULL,
 PRIMARY KEY (`id`)
 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

 CREATE TABLE IF NOT EXISTS `users` (
 `id` int(11) NOT NULL auto_increment,
 `domain_id` int(11) NOT NULL,
 `email` varchar(100) NOT NULL,
 `password` varchar(150) NOT NULL,
 `quota` int(11) NOT NULL DEFAULT 0,
 PRIMARY KEY (`id`),
 UNIQUE KEY `email` (`email`),
 FOREIGN KEY (domain_id) REFERENCES domains(id) ON DELETE CASCADE
 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

 CREATE TABLE IF NOT EXISTS `aliases` (
 `id` int(11) NOT NULL auto_increment,
 `domain_id` int(11) NOT NULL,
 `source` varchar(100) NOT NULL,
 `destination` varchar(100) NOT NULL,
 PRIMARY KEY (`id`),
 FOREIGN KEY (domain_id) REFERENCES domains(id) ON DELETE CASCADE
 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

exit
```

create a password hash and add a test user (using adminer). to create the hash i used

```
dovecot pw -s SHA256-CRYPT
```

now let's configure postfix:

```
cat > /etc/postfix/mysql-virtual-mailbox-domains.cf <<EOF
user = mailserver
password = <second password goes here>
hosts = 127.0.0.1
dbname = mailserver
query = SELECT 1 FROM domains WHERE name='%s'
EOF

postconf virtual_mailbox_domains=mysql:/etc/postfix/mysql-virtual-mailbox-
domains.cf

#postconf virtual_mailbox_base=/var/vmail
#postconf virtual_uid_maps=static:5000
#postconf virtual_gid_maps=static:5000
```

postconf adds stuff to your main.cf file and also reloads postfix, so we can now test with our test domain if it works:

```
postmap -q yourdomain.ch mysql:/etc/postfix/mysql-virtual-mailbox-domains.cf
```

this shouwld now show 1 as an answer

now add virtual mailboxes:

```
cat > /etc/postfix/mysql-virtual-mailbox-maps.cf <<EOF
user = mailserver
password = <second password goes here>
hosts = 127.0.0.1
dbname = mailserver
query = SELECT 1 FROM users WHERE email='%s'
EOF

postconf virtual_mailbox_maps=mysql:/etc/postfix/mysql-virtual-mailbox-
maps.cf

postmap -q john@yourdomain.ch mysql:/etc/postfix/mysql-virtual-mailbox-
maps.cf


cat > /etc/postfix/mysql-virtual-alias-maps.cf <<EOF
user = mailserver
password = <second password goes here>
hosts = 127.0.0.1
dbname = mailserver
query = SELECT destination FROM aliases WHERE source='%s'
EOF

postconf virtual_alias_maps=mysql:/etc/postfix/mysql-virtual-alias-maps.cf

postmap -q forwarded@mydomain.ch mysql:/etc/postfix/mysql-virtual-alias-
maps.cf
#this should return the target email address
```

next up is dovecot

create a user who will own all the mail stored on your server:

```
groupadd -g 5000 vmail
useradd -g vmail -u 5000 vmail -d /var/vmail -m
```

all the configs are stored in `/etc/dovecot/conf.d`

edit `10-auth.conf` and make sure that:

```
auth_mechanisms = plain login
```

and that the sql backend at the end of this file is the only enabled backend:

```
!include auth-sql.conf.ext
```

next is `auth-sql.conf.ext`

make sure the userdb section looks like this (it already did on ubuntu)

```
userdb {
   driver = sql
   args = /etc/dovecot/dovecot-sql.conf.ext
}
```

next is `10-mail.conf` set the mail location to

```
mail_location = maildir:~/Maildir
```

and enable the quota plugin

```
mail_plugins = quota
```

in the file `10-master.conf` edit the `service auth` section and uncomment the `Postfix smtp-auth` settings.. they should look like this:

```
  # Postfix smtp-auth
  unix_listener /var/spool/postfix/private/auth {
    mode = 0666
    user = postfix
    group = postfix
  }
```

further more, find the `service lmtp` section and edit it so it looks like this:

```
service lmtp {
  unix_listener /var/spool/postfix/private/dovecot-lmtp {
    group = postfix
    mode = 0600
    user = postfix
  }
}
```

restart dovecot

```
systemctl restart dovecot
```

next up is `10-ssl.conf`

```
ssl = required
```

and

```
ssl_cert =
```

```
</etc/letseXwiN2Np9MI2DyEaBSHDAYPLTQsQkHAncrypt/live/mail.yourdomain.ch/full
chain.pem
ssl_key = </etc/letsencrypt/live/mail.yourdomain.ch/privkey.pem
```

```
cat >> /etc/dovecot/dovecot-sql.conf.ext <<EOF
driver = mysql
connect = host=127.0.0.1 dbname=mailserver user=mailserver password=<second
password goes here>
user_query = SELECT email as user, \\
  concat('*:bytes=', quota) AS quota_rule, \\
  '/var/vmail/%d/%n' AS home, \\
  5000 AS uid, 5000 AS gid \\
  FROM users WHERE email='%u';
password_query = SELECT password FROM users WHERE email='%u'
EOF
```

```
cat >> /etc/dovecot/conf.d/90-quota.conf <<EOF

plugin {
  quota = maildir:User quota

  quota_status_success = DUNNO
  quota_status_nouser = DUNNO
  quota_status_overquota = "552 5.2.2 Mailbox is full and cannot receive any
more emails"
}

service quota-status {
  executable = /usr/lib/dovecot/quota-status -p postfix
  unix_listener /var/spool/postfix/private/quota-status {
    user = postfix
  }
}

plugin {
   quota_warning = storage=95%% quota-warning 95 %u
   quota_warning2 = storage=80%% quota-warning 80 %u
   quota_warning3 = -storage=100%% quota-warning below %u
}
service quota-warning {
   executable = script /usr/local/bin/quota-warning.sh
   unix_listener quota-warning {
     group = dovecot
     mode = 0660
   }
 }
EOF
```

/usr/local/bin/quota-warning.sh

```
#!/bin/sh
PERCENT=$1
USER=$2
cat << EOF | /usr/lib/dovecot/dovecot-lda -d $USER -o
"plugin/quota=maildir:User quota:noenforcing"
From: postmaster@webmail.example.org
Subject: Quota warning - $PERCENT% reached

Your mailbox can only store a limited amount of emails.
Currently it is $PERCENT% full. If you reach 100% then
new emails cannot be stored. Thanks for your understanding.
EOF
```

```
chmod 755 /usr/local/bin/quota-warning.sh

systemctl restart dovecot.service

postconf "smtpd_recipient_restrictions = reject_unauth_destination
check_policy_service unix:private/quota-status"

chown root:root /etc/dovecot/dovecot-sql.conf.ext
chmod go= /etc/dovecot/dovecot-sql.conf.ext
```

tell postfix to send email via lmtp to dovecot:

```
postconf virtual_transport=lmtp:unix:private/dovecot-lmtp
```

enable sieve plugin on lmtp protocol (this is where we get mails from postfix, so it's where we want to route them through sieve rules)

edit `/etc/dovecot/conf.d/20-lmtp.conf` and look for `mail_plugins` in the `protocol lmtp` section (at the very end in the default config).. edit the line like this:

```
mail_plugins = $mail_plugins sieve
```

pending issue i had to allow the world to read and write to `/var/run/dovecot/quota-warning` because the vmail user was used to access it and adding the vmail user to the dovecot group did not help to solve the issue.. not sure if this is really a good idea, i hope for some good feedback on this later on, so far i did:

```
chmod a+w /var/run/dovecot/quota-warning
chmod a+r /var/run/dovecot/quota-warning
```

now we can do some testing with `swaks` to send emails and `mutt` to read them over imap:

```
swaks --to myuser@yourdomain.ch --server mail.yourdomain.ch
mutt -f imaps://myuser@yourdomain.ch@mail.yourdomain.ch
```

postfix smtp(d) config

```
postconf smtpd_sasl_type=dovecot
postconf smtpd_sasl_path=private/auth
postconf smtpd_sasl_auth_enable=yes
postconf smtpd_tls_security_level=may
postconf smtpd_tls_auth_only=yes
postconf
smtpd_tls_cert_file=/etc/letsencrypt/live/mail.yourdomain.ch/fullchain.pem
postconf
smtpd_tls_key_file=/etc/letsencrypt/live/mail.yourdomain.ch/privkey.pem
postconf smtp_tls_security_level=may
```

to enable submission service (port 587 for sending emails from clients) edit `/etc/postfix/master.cf` and uncomment the lines for the submission service. I left the restrictions commented out, because i don't want any furhter restrictions for my clients besides the need to authenticate.

```
submission inet n       -       y       -       -       smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_tls_auth_only=yes
  -o smtpd_reject_unlisted_recipient=no
#  -o smtpd_client_restrictions=$mua_client_restrictions
#  -o smtpd_helo_restrictions=$mua_helo_restrictions
#  -o smtpd_sender_restrictions=$mua_sender_restrictions
#  -o smtpd_recipient_restrictions=
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
```

```
systemctl restart postfix
```

additional postfix settings

mail size limit

i think 10MB is just too small, so i allowed 30mb instead

```
postconf message_size_limit=31457280
```

regex based virtual aliases

i've added this as well to my config. however, I called the config file /etc/postfix/regex_aliases.map so

i can create mappings for different domains and different purposes.

install roundcube

i went with the latest roundcube because the versioni that came with ubuntu at the time of writing did not have the new elastic gui which i really wanted.

download the lastest verison of Roundcube and save the tar file to a temporary directory on your server

```
cd /tmp
wget
https://github.com/roundcube/roundcubemail/releases/download/1.4.1/roundcube
mail-1.4.1-complete.tar.gz
```

```
cd /var/www/webmail
tar xvf /tmp/roundcubemail-1.4.1-complete.tar.gz
mv roundcubemail-1.4.1/{.,}* pub/
rmdir roundcubemail-1.4.1/
cd pub
```

install dependencies according to INSTALL instructions

```
apt install php7.2-json php7.2-xml php7.2-mbstring php7.2-zip php7.2-intl
php7.2-gd
```

further more i had to set the `date.timezone = Europe/Zurich` in the php.ini and reload apache to apply the changes

follow the INSTALL instructions.. for version 1.4.1 i did this:

```
  chown -R www-data.www-data logs temp
  pwgen -s1 30
```

take note of the password created with pwgen and start the mysql client

```
mysql
```

inside the mysql console run these commands to create the database:

```
CREATE DATABASE roundcubemail /*!40101 CHARACTER SET utf8 COLLATE
utf8_general_ci */;
GRANT ALL PRIVILEGES ON roundcubemail.* TO roundcube@localhost
    IDENTIFIED BY '<password goes here>';
quit
```

```
mysql roundcubemail < SQL/mysql.initial.sql
```

continue in the web-based installer found at https://mail.yourdomain.ch/installer

here are the things i've changed:

- imap port changed to 993
- imap server set to

```
ssl://mail.yourdomain.ch
```

- smtp server set to

```
tls://mail.yourdomain.ch
```

- set database password to the one you noted down before
- add at least `managesieve` and `password` plugin

move install directory out of www root.. we can always symlink to it later if we need the installer again

```
cd /var/www/webmail/pub/
mv installer ../
```

configure the plugins:

create a new password for a db admin user which we need for the password plugin:

```
pwgen -s1 30
```

create the db admin user for roundcube:

```
mysql
```

in the mysql console add the user:

```
grant all on mailserver.* to 'rcpasswd'@'localhost' identified by '<password
goes here>';
quit
```

now write our config for the password plugin:

```
cat > /var/www/webmail/pub/plugins/password/config.inc.php <<EOF
<?php
\$config['password_driver'] = 'sql';
\$config['password_confirm_current'] = true;
\$config['password_minimum_length'] = 7;
\$config['password_force_save'] = false;
\$config['password_algorithm'] = 'dovecot';
\$config['password_dovecotpw'] = '/usr/bin/doveadm pw -s SHA256-CRYPT';
\$config['password_dovecotpw_method'] = 'SHA256-CRYPT';
\$config['password_dovecotpw_with_method'] = true;
\$config['password_db_dsn'] = 'mysql://rcpasswd:<password goes
here>@localhost/mailserver';
\$config['password_query'] = "UPDATE users SET password=%D WHERE email=%u";
```

```
?>
EOF
```

the managesieve plugin is simpler, just provide the hostname of the sieve server and we're golden:

```
cat > /var/www/webmail/pub/plugins/managesieve/config.inc.php <<EOF
<?php
\$config['managesieve_host'] = 'localhost';
?>
EOF
```

client auto configuration

postponed.. tutorial

proxmox mail gateway

i decided on using proxmox mail gateway (PMG) for spam and virus filtering rather than setting up rspamd or any other mail filter solution i have to maintain myself. i've tested PMG in the past and it yielded a pretty good detection rate. my ultimate goal is that i don't need to spend too much time dealing with spam filters, they should be there and just do their job.. PMG did just that during my tests using some catchall domains to gather as much spam as i could :)

i've installed PMG onto another Virtual Machine as i host a virtual host myself. if you have to pay alot of money for a vps and you already have one for your mailserver, you can also run PMG inside a LXC container, more details on the installation can be found in the admin guide.

so i've downloaded the latest ISO from the Proxmox webpage and installed my VM using this image. the installation is very easy and very fast.

i set a public ip with a hostname filter.yourdomain.ch.

after the installation is complete, you can access the web-interface on https://filter.yourdomain.ch:8006

your root password is also your login for the web-interface. i did disable ssh password login and i've blocked all ports except 22 and 25 from the outside world in my firewall, so nobody can access the web-interface and brute-force my password.

once you're logged in to the web-interface, do the basic setup. first make sure the dns and time settings are correct. you can change those by clicking on “Configuration” in the left column.

once this is all set, go ahead and click on the “Mail Proxy” settings.

- under Relaying enter your mailserver's ip as “Default Relay”. this is the ip to which we want to relay incoming mail from the internet after it passes all the filters.
- leave port 25
- i've disabled MX lookups, not sure why they should be needed here.

- under “Relay Domains” enter all your domains you want to accept emails for on your mailserver
- in the Options tab i've enabled “Verify Receivers” which will verify that the receiver address is actually valid before accepting the email. i've set it to “Yes(550)” to work with my above postfix setup.
- i have disabled Greylisting as this delays mail delivery significantly and that's a bit of a pain if you wait for account confirmation emails or booking confirmations etc. i'll re-enable it if the spam detection rate is too low.
- also in Options, i have enabled DNSBL and i've entered the following two blacklists to query: `b.barracudacentral.org,zen.spamhaus.org`. please note that you need to register your dns servers at barracudacentral prior to using them and spamhaus asks you to rsync their database to your own dns if you have a high volume server.. i don't :)
- there is no need to configure any transports. this is only needed if you want to route incoming mails for different domains or addresses to different servers.
- in the networks tab, you can add the network or ip of your mailserver, in case it is not in the same subnet as your filter.. if it's in the same subnet there is no need to add anything here, as the same subnet is allowed to relay through PMG by default.
- since we will be relaying our outgoing emails from our mailserver via PMG as well, we will use PMG's DKIM signing function. to enable this, in the DKIM tab you need to **first add a new selector** before you can enable DKIM .. that's a bit confusing. as selector i've entered the current date like `20200101` you can be more creative if you want to. i then enabled DKIM and checked the box to sign all outgoing traffic. like this there is no need to add each of your domains separately to the DKIM domains.
- for DKIM to work you need to add a TXT entry to your domian's DNS record. you can click on “View DNS Record” to get a copy-paste snippet to paste right into your bind zone file if you're using bind as your name-server.
- once all these settings where done, i had to login to the filter via ssh and **manually restart postfix**. otherwise postfix would bind port 25 to 127.0.0.1 only. i guess rebooting the entire filter would fix this issue as well.

tag and deliver spam instead of quarantine

I'm not sure I or my users would be happy with waiting for reports to find out why a recently sent mail did not reach them. after all it's always a good feeling if you can tell someone on the phone that you didn't find their email in your spam folder either, to convince them that they might have had a typo in your email address :) ..

First you need to make sure that spam is no longer quarantined but instead marked and forwarded.

you can either mark an email by modifying its subject or by adding a header element.. i don't like changing the visible part of the email message, so i opted to go for an additional header field that marks spam.

to create it, go to the `Mail Filter→Action Objects` page in the PMG web-interface and add a new action object of the `Header Attribute` type. start the header atribute with `x-` and choose something meaningful.. i went for `x-spam-mail:yes`

next we need to make sure that all spam mail is tagged with this header field instead of quarantined.

in the PMG web interface click on `Mail Filter` in the left column. you will now see a list of all active

or inactive mail filters. by selecting a filter, you can then see on the right hand side column which actions will be executed. you can simply remove the Quarantine action from the active ones and then drag and drop the newly created action object to the active actions or click the + sign to add it.

now on to your postfix mail server.. we need to add a global sieve rule to dovecot that will move spam into a spam folder.. edit `/etc/dovecot/conf.d/90-sieve.conf` and look for “sieve_after” templates.. add a new line after those that looks like this:

```
sieve_after = /etc/dovecot/sieve-after
```

now create the sieve-after directory:

```
mkdir /etc/dovecot/sieve-after
```

all filters found in this directory will be executed AFTER each user's own filters. so a user can create his own filters to whitelist spam in our case.

```
cat > /etc/dovecot/sieve-after/spam-to-folder.sieve <<EOF
require ["fileinto","mailbox"];

if header :contains "x-spam-mail" "yes" {
 fileinto :create "INBOX.Junk";
 stop;
}
EOF
```

now compile the sieve filter:

```
sievec /etc/dovecot/sieve-after/spam-to-folder.sieve
```

lastly restart dovecot to re-read the config we altered bove

```
systemctl restart dovecot
```

to test, send an email from outside to your mail account with the following line in the body:

```
XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X
```

future improvements

Bayesian learning by moving mail to junk folder

in order for proxmox's Bayesian filter to learn and start working you need to provide spam and ham mails. to do this, one needs to copy spam that passed the filter to a temporary location on the proxmox mail gateway and then tell the Bayesian filter that this is spam. it needs at least 200 spam and ham mails to start working. ideally, we should be able to do this by simply moving an email to the spam folder on our imap account and the email should then be submitted to the bayesian filter of PMG automatically..

some information i will need to do this:

- the command on PMG to submit a ham or spam to the learning system is

```
sa-learn --spam filename
```

- a solution on how to use imap sieve filters to trigger a bash script when a mail is moved into a specific folder can be found in the all mighty ISPmail tutorial

adjustments to postfix settings on our mailserver

we can now limit access for incoming mails so that postfix only accepts connections from our mailfilter. to do this, edit the `smtpd` line in `/etc/postfix/master.cf` and add the following option:

```
smtp      inet  n       -       y       -       -       smtpd
  -o smtpd_client_restrictions=permit_mynetworks,reject
```

don't forget to restart postfix

further more we can configure our mailserver to send all its mails through our proxmox gateway to allow proxmox to track outgoing mails and scan them for viruses as well. to do that we can set the `relayhost` accordingly:

```
postconf relayhost=filter.yourdomain.ch:26
```

note port 26, that's because proxmox mail gatway distinguishes between incoming and outgoing mail by accepting them on different smtp ports. by default port 25 is for incoming and port 26 for outgoing mail.

greylisting

by default PMG uses greylisting. this means, that every email coming from a new sender address will first be rejected for a duration of a couple of minutes. i think 3 minutes is the actual greylist timeout on PMG. however, the delay that occurrs in reality will be dependent also on the sending mail server's retry interval.

you can see all attempts that where blocked by geylisting if you go to the tracking center and check the “Include Greylist” search option, then click search.

enterprise vs. free

PMG is free open source software with an optional enterprise subscription. For a private person, the enterprise license is too expensive, but if you use this setup for a production server in a copmany, you might want to consider getting the enterprise subscription, because besides support, it also offers access to an enterprise repository, which contains heavily tested packages whereas the non-subscription repo is kind of the “beta tester” repo, meaning, a new package gets to the no-sub repo

first and only after a certain amount of time of successful distribution through the no-sub repos will it enter the enterprise repo. So you either contribute by testing new packages or by paying money :)

by default, PMG comes with the enterprise repo pre-configured which means you won't be able to update if you don't have a subscription. if you want to use the free repo, you need to change your apt configuration:

```
rm /etc/apt/sources.list.d/pmg-enterprise.list
echo "deb http://download.proxmox.com/debian/pmg buster pmg-no-subscription"
>> /etc/apt/sources.list.d/pmg-no-sub.list
```

as a non-enterprise user you will have to run updates from the command line, while enterprise users can do it from the web-ui IIRC (i am a poor private user ;))

Create new users

to manage user accounts, login to adminer to edit your `mailserver` database.

first make sure the domain is added by checking the `domains` table. if the domain does not exist, add it.

make note of the domain id of the domain you want to add a new user for.

to add a new forwarding, make a new entry in the `aliases` table, make sure you don't forget to fill in the `domain_id` field

to add a new user account, make a new entry in the users table. make sure you don't forget to fill in the `domain_id` field. leave `quota` empty or 0 in order to not enforce a mail box quota. if you want to set a quota, enter the amount of space the mailbox is allowed to use **in bytes**. for the `password` field, you need to generate a password hash on the command line of your mailserver. use the command

```
dovecot pw -s SHA256-CRYPT
```

and then enter the new password. It will return a SHA hash which you can then enter in the password field.

lastly, to activate the new mail user account, send an email to that address.

From:
http://wiki.psuter.ch/ - **pswiki**

Permanent link:
http://wiki.psuter.ch/doku.php?id=setup_basic_mailserver_with_postfix_dovecot_sieve&rev=1616220049

Last update: **20.03.2021 07:00**