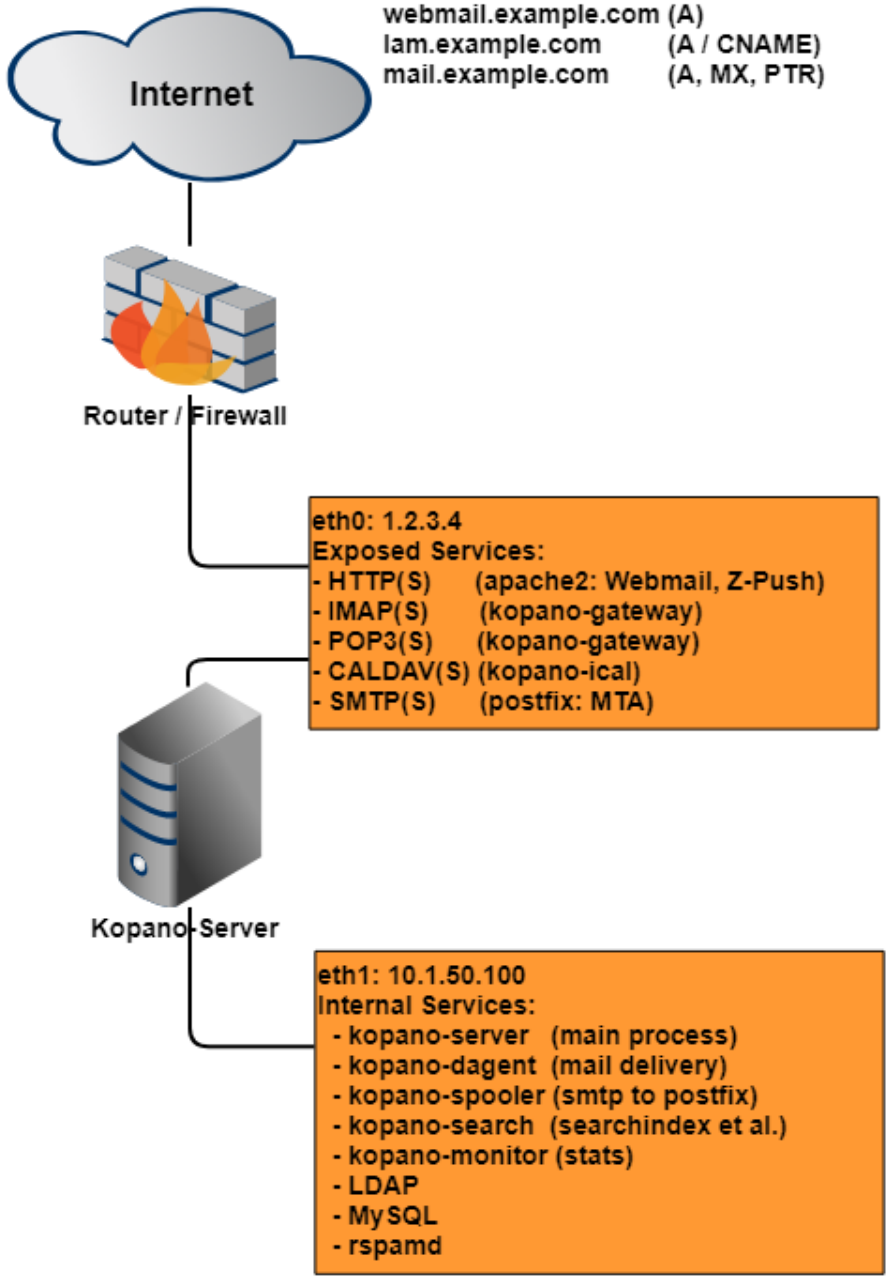
Serversetup multi-tenant Kopano+Postfix+SASL+rspamd+openLDAP (Debian 10 buster)

Kopano Groupware and dependent infrastructure setup guidelines.



Disclosure

I am not affiliated with Kopano or any other company mentioned in this how-to.

This how-to is written to the best of my knowledge, but as always: if this blows up your server, eats your cat or steals your lover, you're on your own.

Preface

This is the new and reworked setup guide for a complete virtual-user, multi-tenant, LDAP-backed Kopano Setup based on Kopano 8.x and Debian10. Spamassassin has been dropped in favor of rspamd and postfix has been augmented to use DNS-based blacklisting based on multiple online blacklist providers. LDAP configuration has been expanded with the suggested optimizations taken from the Kopano documentation

For LDAP Management I went with ldap-account-manager (LAM from hereon), which includes its own version of phpLDAPadmin for its "Tree View feature". I adapted the templates I created formerly which works alright. Although LAM supports Kopano accounts you need the paid 'Pro' version, and I'm not sure if it fully supports multi-tenant setups (correct me if I'm wrong).

Links:

- https://www.ldap-account-manager.org/
- https://documentation.kopano.io/

Migration from Zarafa

- https://documentation.kopano.io/kopano_migration_manual/zcp_migration.html

Updated templates for LAM

- modify_kopanoUser.xml
- create_kopanoUser.xml
- create_kopanoCompany.xml

Since I often get asked which **FREE mail (groupware) alternatives** exist, here is a non-complete list of viable alternatives:

- https://sogo.nu/ (frontend only, needs manual MTA / IMAP / User management integration)
- https://mailcow.email/ (Docker-ized, Sogo-based, fully-featured multi-tenant groupware)
- https://www.zimbra.com/open-source-email-overview/
- https://www.open-xchange.com/ (frontend only, needs manual MTA / IMAP integration)

All of those will probably need some tinkering in one way or another, but all of them reportedly provide open groupware implementations.

A more integrated, albeit not completely free-as-in-speech alternative is the UCS server with the Kopano App:

- https://www.univention.com/products/ucs/ - UCS Core Edition
- https://www.univention.com/products/univention-app-center/app-catalog/kopano-core/

Throughout the guide only lines in config files that need to be changed are shown.
This guide assumes:

- general linux knowledge: basic commands and system administration know-how
- basic mysql knowledge
- basic ldap knowledge
- basic apache knowledge
- intermediate postfix / rspamd knowledge

The Server will be directly exposed to the internet on the ports shown in the diagram above. Therefore another interface is assigned to the server that will handle internal connections that need not be seen from the internet and are only used for local or server internal communication.

- This is NOT a copy-and-paste how-to!
- The domain 'example.com' is used, replace all instances with your own domain (in LDAP too, like: dc=example,dc=com).
- The external interface is assumed to have the IP 1.2.3.4. Replace with your own external interface address.
- The internal interface is assumed to have the IP 10.1.50.100. Replace with your own internal interface address.
- The password 'secret' will be used as a placeholder for all passwords. Replace with DIFFERENT, STRONG passwords accordingly!
- This setup assumes a local MySQL database. If you use a remote, dedicated database you will need to change the database settings and grants in various config files or the database itself.

Minimum system requirements

The server sizing below is deducted from my personal operational experience with Zarafa/Kopano servers in SMB environments.

For official guidelines consult the Kopano docs:

- https://documentation.kopano.io/kopanocore_administrator_manual/installing.html#hardware-recommendations

For a satisfying user experience, provision at least the following server resources

- 2x CPU (or 4x vCPU on virtualized/hosted systems)
- 8GB RAM
- 50GB disk space (preferrably "fast" storage, e.g. SAS/SSD)

Usually a hosted server in the ~50$ / ~40€ price range with SSD storage should do. For on-premise deployments a HP MicroServer could be a cost-efficient alternative.

For production setups a second (offsite) server for backup storage / failover is highly recommended.

DNS configuration

When it comes to mailing, **DNS records play an important role** in successfully delivering your email.

You **will need** at least:

- A valid A record pointing to your server (e.g. "mail.example.com")
- A valid PTR record (reverse DNS) which resolves your server IP back to "mail.example.com"
- A valid MX record pointing to your server

You most probably also want the following records set up:

- A **SPF** entry to specify which servers are allowed to send mail for your domain
- A **DKIM** record to verify that emails actually where sent from your server (discussed later in the rspamd setup)
- A **CAA** record to tie your webserver certificates to your certificate vendor

In your public DNS configuration, configure the following domains to point at your server (as always, replace example.com with your mail domain):

```
mail.example.com
webmail.example.com (can be a CNAME to mail.example.com)
lam.example.com (can be a CNAME to mail.example.com)
```

Also don't forget to set up the corresponding MX records!

Server

- Debian 10 minimal install from: https://www.debian.org/CD/netinst/
- Kopano 8.x and zpush (now included in debian repo)
- MariaDB 10.1, openLDAP 2.4, postfix
- LAM 7.0, downloaded manually from the LAM homepage: https://www.ldap-account-manager.org/lamcms/releases
- rspamd, from the official rspamd repository: https://rspamd.com/downloads.html

Certificates

For this setup two different certificates are needed. One for the Webserver (webmail.example.com) and one for the mail server (mail.example.com). It is advised to use official certificates or LetsEncrypt. However for testing purposes you can create your own ones with openssl.

The following example locations are used for certificate paths throughout the guide:

- /etc/ssl/private/mail.example.com/cert.pem
- /etc/ssl/private/mail.example.com/privkey.pem
- /etc/ssl/private/mail.example.com/chain.pem
- /etc/ssl/private/mail.example.com/fullchain.pem

Replace those according to the location on your system.

If you plan to use LetsEncrypt I recommend the "**dehydrated**" LE client. It's lightweight, reliable and easy to handle:

- https://github.com/lukas2511/dehydrated

Server preparation

Set sysctl values

/etc/sysctl.d/99-sysctl.conf

```
fs.aio-max-nr = 655360
kernel.domainname = mail.example.com
net.core.optmem_max = 40960
net.core.rmem_default = 16777216
net.core.rmem_max = 16777216
net.core.wmem_default = 16777216
net.core.wmem_max = 16777216
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.all.rp_filter=1
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.rp_filter=1
net.ipv4.tcp_keepalive_intvl = 10
net.ipv4.tcp_keepalive_probes = 3
net.ipv4.tcp_keepalive_time = 60
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_syncookies=1
net.ipv4.tcp_wmem = 4096 65536 16777216
vm.swappiness = 0

# if needed, reserve ports in high port range
# e.g. this would reserve all ports below 30000:
#net.ipv4.ip_local_port_range=30000 65535
```

Update bootloader

Edit grub config file:

/etc/default/grub

```
GRUB_CMDLINE_LINUX="transparent_hugepage=never apparmor=0"
```

Update grub:

```
~# grub-mkconfig -o /boot/grub/grub.cfg
```

Reboot the server.

Install firewall, fail2ban and ulogd

Nowadays every server that is exposed to the internet should be protected by a firewall and/or automated defense tools, that block unwanted traffic and bruteforce attacks. Since linux comes with the iptables firewall per default, we only need to configure it. There are several tools that help with this task (e. g.: firewalld, ufw etc.), but in this example we use `firehol` and `fail2ban` to set up our perimeter. Firewall traffic logs will get logged to `ulogd` for later inspection if needed.

```
~# apt install firehol fail2ban ulogd2
```

Edit the fail2ban configuration:

/etc/fail2ban/jail.conf

```
# only changed config parameters listed here, leave rest at default!

banaction = iptables-ipset-proto6
banaction_allports = iptables-ipset-proto6-allports

[sshd]
port    = 22
logpath = %(sshd_log)s
backend = %(sshd_backend)s
```

/etc/fail2ban/jail.d/defaults-debian.conf

```
[sshd]
enabled = true
```

Edit the firehol configuration:

/etc/firehol/firehol.conf

```
version 6

# replace 'eth0' with the name of your public interface
# eth0
IFACE_ETH0="eth0"               # public network interface
IFACE_ETH0_IP="1.2.3.4"         # IP address
IFACE_ETH0_NET="1.2.3.4/XX"     # IP address with netmask, e.g. 1.2.3.4/28

# eth1
# enable only if your server has a second NIC with a private IP address
IFACE_ETH1="eth1"               # private network interface
IFACE_ETH1_IP="10.1.50.100"     # IP address
IFACE_ETH1_NET="10.1.50.100/24" # IP address with netmask


# public interface eth0
interface4 "${IFACE_ETH0}" eth0_pub dst "${IFACE_ETH0_IP}"
  policy              drop
  protection          strong
#  server ICMP         accept  # only enable if you want to be able to ping the server
  server http         accept
  server https        accept
#  server imap         accept  # unencrypted connections shouldn't be used anymore, enable only for legacy
clients
  server imaps        accept
#  server pop3         accept  # unencrypted connections shouldn't be used anymore, enable only for legacy
clients
  server pop3s        accept
  server smtp         accept
  server smtps        accept
  server ssh          accept
  server submission   accept
  client all          accept   # allow all outgoing traffic
  # prevent logging of dropped unmatched traffic
  # https://firehol.org/faq/#firehol-logging
  server anystateless nolog  drop

# eth1
# enable only if your server has a second NIC with a private IP address
# we assume that the private network is safe
# if this is not the case, remove "policy accept" and add your own rules!
#interface4 "${IFACE_ETH1}" eth1_lan dst "${IFACE_ETH1_IP}"
#  policy accept

# fail2ban
postprocess -warn /usr/bin/fail2ban-client reload || return 1
```

/etc/firehol/firehol-defaults.conf

```
# only changed config parameters listed here, leave other values at their default!

WAIT_FOR_IFACE="eth0"
FIREHOL_LOG_MODE="NFLOG"
```

Enable and start the services:

```
~# systemctl enable fail2ban.service
~# systemctl enable ulogd2.service
~# systemctl enable firehol.service

~# systemctl start fail2ban.service
~# systemctl start ulogd2.service
```

Test the firewall setup:

```
# To test if the firewall setup works as expected, first test firehol with the 'try' option
# This will automatically reset the firewall in case of a misconfiguration after 10s
~# firehol try

# When everything works as expected, manually disable it again and start it via the systemd unit
~# firehol stop
~# systemctl start firehol.service
```

Now reboot the server to ensure the firewall works as expected.

Install needed packages and utilities for Kopano

> Debian is very noisy and wants to configure postfix and ldap during the installation of the package. Refuse to do so or just let it create an empty configuration. You will set up these programs manually later.

First, add the rspamd repo as described here: https://rspamd.com/downloads.html

Then install all needed packages for the mailserver:

```
~# apt install postfix postfix-ldap postfix-pcre sasl2-bin apache2 mariadb-server-10.1 php-mapi \
               kopano-core kopano-dagent kopano-gateway kopano-ical kopano-monitor kopano-search kopano-server
kopano-spooler \
               kopano-webapp-apache2 kopano-webapp-plugin-desktopnotifications kopano-webapp-plugin-spell \
               kopano-webapp-plugin-titlecounter kopano-utils kopano-backup \
               z-push z-push-backend-kopano z-push-common z-push-ipc-memcached

# for LetsEncrypt integration also install:
~# apt install dehydrated dehydrated-apache2
```

Enable apache modules, disable the default site, generate dh_params and restart the webserver:

```
~# a2dissite 000-default.conf
~# a2enmod expires
~# a2enmod headers
~# a2enmod http2
~# a2enmod php7.3
~# a2enmod rewrite
~# a2enmod setenvif
~# a2enmod ssl

# for LetsEncrypt integration also enable:
~# a2enconf dehydrated

# generate a custom dh_params file
# WARNING: this may take some time!
~# openssl dhparam -out /etc/apache2/dhparams_4096.pem 4096
~# cp -a /etc/apache2/dhparams_4096.pem /etc/postfix/dhparams_4096.pem
~# chmod 0600 /etc/apache2/dhparams_4096.pem /etc/postfix/dhparams_4096.pem

~# systemctl restart apache2.service
```

Install python dependencies for automatic ham/spam handling (used later in the guide):

```
~# apt install python3-pip
~# pip3 install wheel
~# pip3 install inotify
```

Database (MariaDB/MySQL) Setup

> Please configure MariaDB / MySQL in a sane way if you are setting up a production system.
>
> The **default mariaDB configuration is not fit for production use** and it is strongly recommended that you change it accordingly!
>
> Docs: https://documentation.kopano.io/kopanocore_administrator_manual/performance_tuning.html

Create database, users and permissions for Kopano:

```
~# mysql -uroot -p
mysql> create database kopano CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;;
mysql> grant all on kopano.* to 'kopano'@'localhost' identified by 'secret';
mysql> flush privileges;
```

LDAP Setup

Probably the part that generates the most trouble for sysadmins. First we install Open-LDAP.

⚠️ During installation: create your LDAP base DN "`dc=example,dc=com`"!

⚠️ Take not of the password password of the "`admin`" user!

Install the LDAP server and tools:

```
~# apt install slapd ldap-utils
```

For easier LDAP management we will utilize `ldapvi`:

```
~# apt install ldapvi
```

LDAP server configuration

First adjust the default-file:

/etc/default/slapd

```
SLAPD_SERVICES="ldap:/// ldapi:///"
```

Import the Kopano schema into LDAP and restart the server

```
zcat /usr/share/doc/kopano-server/kopano.ldif.gz | ldapadd -H ldapi:/// -Y EXTERNAL
systemctl restart slapd.service
```

Configure LAM (ldap-account-manager)

Download the latest LAM debian package from here: https://www.ldap-account-manager.org/lamcms/releases (Version 7.0 at time of writing)

```
# Probably we re missing some dependencies.
# First install will therefor produce errors
~# dpkg -i ldap-account-manager_7.0-1_all.deb

# Automatically install missing dependencies
~# apt -f install

# Now redo the installation, it will work now that all dependencies are satisfied
~# dpkg -i ldap-account-manager_7.0-1_all.deb
```

Configure the webserver for LAM

LAM comes with a default apache config that we can use:

```
~# ln -s /etc/ldap-account-manager/apache.conf /etc/apache2/conf-available/ldap-account-manager.conf
~# rm -f /etc/apache2/conf-enabled/ldap-account-manager.conf
```

Create an apache vhost for the domain "lam.example.com".

⚠️ Don't forget to replace the domain name and the path to the certificates accordingly!

/etc/apache2/sites-available/lam.conf

```
<VirtualHost 1.2.3.4:80>

  ServerName      lam.example.com
  ServerAdmin     webmaster@example.com

  RewriteEngine   On
  RewriteCond     %{HTTPS} off
  RewriteRule     ^(.*)$   https://%{SERVER_NAME}%{REQUEST_URI} [L,R]

</VirtualHost>

<VirtualHost 1.2.3.4:443>
  ServerName              lam.example.com
  ServerAdmin             webmaster@example.com


  SSLEngine               on
  SSLCertificateFile      /etc/ssl/private/lam.example.com/cert.pem
  SSLCertificateKeyFile   /etc/ssl/private/lam.example.com/privkey.pem
  SSLCertificateChainFile /etc/ssl/private/lam.example.com/chain.pem
  SSLCACertificateFile    /etc/ssl/private/lam.example.com/fullchain.pem
  SSLProtocol             all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
  SSLHonorCipherOrder     on
  SSLCompression          off
  SSLSessionTickets       off
  SSLOpenSSLConfCmd       DHParameters "/etc/apache2/dhparams_4096.pem"
  SSLCipherSuite          ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-
POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256

  Protocols               h2 h2c http/1.1

  Include                 /etc/apache2/conf-available/ldap-account-manager.conf

  ErrorLog  /var/log/apache2/lam-error.log
  CustomLog /var/log/apache2/lam-access.log combined

</VirtualHost>
```

In "`/etc/apache2/conf-available/ldap-account-manager.conf`" replace the first line:

```
# comment out this line:
#Alias /lam /usr/share/ldap-account-manager

# add this instead:
DocumentRoot /usr/share/ldap-account-manager
```

Enable the vhost and restart apache:

```
~# a2ensite lam.conf
~# apachectl configtest
~# systemctl restart apache2.service
```

LAM should now load (with a valid certificate) in your browser with this domain: "`https://lam.example.com`"

⚠️ Before you continue, make sure you change the default LAM account passwords for its configuration:

- In the top-right, click "LAM configuration"

- Set new passwords for both "general settings" and "server profiles" (the default password is "`lam`")

Now set up a login profile for your LDAP server running at localhost under "server profiles".

After you logged in with the "admin" account, create a new object of type '**inetOrgPerson**' called '**kopanoservice**':

- Click "Tree View" in the top right
- Select top level entry "dc=example,dc=com"
- Click "Create child entry"
- Select the "Default" template
- Choose "inetOrgPerson" from select box click proceed
- Fill in the fields **cn** (kopanoservice), **sn** (kopanoservice) and **userPassword** (SSHA password hash)

This will be the user which Kopano and postfix will use to access LDAP.

If done correctly LAM should display a dn of '**cn=kopanoservice,dc=example,dc=com**' for the user.

Install Kopano LAM templates

Download the templates that are attached to this page (see at the top for download links) and put them into the following folders:

- copy **create_kopanoUser.xml** and **create_KopanoCompany.xml** into **/usr/share/ldap-account-manager/templates/3rdParty/pla/templates/creation/**
- copy **modify_KopanoUser.xml** into **/usr/share/ldap-account-manager/templates/3rdParty/pla/templates/modification/**

Create a Company, User and Alias

With the templates in place, log into LAM again and:

- create a company (Kopano Company)
- create a user in the newly created company (Kopano User Account)
- edit the user, add new attribute 'KopanoAlias'
    - assign at least one email alias to the user

This user and the mailalias will be needed later for testing the functionality of the server.

Edit LDAP permissions

Use ldapvi to edit the LDAP configuration. the LDAP server must be restarted after every edit to read the new configuration.

```
~# ldapvi -h ldapi:// -Y EXTERNAL -b cn=config
```

Since the following changes are intrusive and might render the server unusable if done in a wrong way, **be sure to make a backup of the running config and your database** before you edit the LDAP settings!

Search for the entry "**olcDatabase={1}mdb,cn=config**" (there should be a number in front of it). Add the following lines at the end of the entry.

```
olcDbIndex: entryCSN eq
olcDbIndex: entryUUID eq
olcDbIndex: objectClass eq
olcDbIndex: cn pres,eq,sub
olcDbIndex: gidNumber eq
olcDbIndex: mail pres,eq,sub
olcDbIndex: memberUid eq
olcDbIndex: ou eq
olcDbIndex: uid eq
olcDbIndex: uidNumber eq
olcDbIndex: uniqueMember eq
olcDbIndex: kopanoAccount eq,pres
olcDbIndex: kopanoAliases eq
olcDbIndex: kopanoViewPrivilege eq
olcDbIndex: sn pres,eq,sub
olcDbIndex: givenName pres,eq,sub
```

Save the config and restart the LDAP server.

Now we will change the default access policy (ACL) of the LDAP server.

The desired setup looks like this:

- The "admin" user has full administrative access to the server (as per default)
- The "kopanoservice" account has read access to the full LDAP tree
- One or more domain admin accounts only have access to their respective domain subtree ("ou" - "organizational unit" in LDAP terms) to create and manage user accounts

Be careful when you edit the configuration! Editing the LDIF in ldapvi can be tricky at times. If in doubt, check the manual: http://www.lichteblau.com/ldapvi/manual/

- Every entry needs a unique number in curly brackets. Order entries by incrementing numbers.
- Ordering is important - LDAP ACLs are evaluated in order and you can block access to your users accidentally if the permissions are evaluated in the wrong way
- No whitespace at the beginning or the end of lines! (hint: use "**:set list**" to spot them more easily)

Edit the config to set the ACLs:

```
~# ldapvi -h ldapi:// -Y EXTERNAL -b cn=config
```

Again, search for the entry "**olcDatabase={1}mdb,cn=config**" (there should be a number in front of it). Replace all "**olcAccess**" entries with something like this:

```
olcAccess: {0}to attrs=userPassword,shadowLastChange by dn="cn=<MAILDOMAINADMIN>,ou=<MAILDOMAIN>,dc=example,
dc=com" write by * none break
olcAccess: {1}to attrs=userPassword by self write by anonymous auth by * none
olcAccess: {2}to attrs=shadowLastChange by self write by * read
olcAccess: {3}to dn="dc=example,dc=com" by * read by * auth
olcAccess: {4}to dn.subtree="dc=example,dc=com" by dn="cn=kopanoservice,dc=example,dc=com" read by * none break
olcAccess: {5}to dn.subtree="ou=<MAILDOMAIN>,dc=example,dc=com" by dn="cn=<MAILDOMAINADMIN>,ou=<MAILDOMAIN>,
dc=example,dc=com" write by * none break
olcAccess: {6}to dn.subtree="dc=example,dc=com" by dn="cn=admin,dc=example,dc=com" manage break
olcAccess: {7}to * by * none
```

You can duplicate the lines "**{0}**", "**{5}**" and "**{6}**" for every mail domain you host to delegate access to multiple mail admins. Don't forget to increment the numbers if you add more domains later.

The entry "**{5}**" allows subtree access to a delegated mail domain admin. Replace "<MAILDOMAIN>" with the mail domain and "<MAILDOMAINADMIN>" with the user account of the mail domain admin.

Save the config and restart the LDAP server.

Now test if the permissions work as expected:

- "admin" can access / edit everything
- "kopanoservice" has read access to the whole tree
- mail admins can access and edit their mail domain and edit entries within it

If everything works as expected, continue to set up the kopano server components.

⚠️ **If this doesn't work, do not continue with the setup, as both kopano and postfix depend on working LDAP permissions and further configuration steps will fail!**

Kopano Server Setup

Be sure to read the admin guide, this will cover only what needs to be changed so Kopano will work in this specific setup. All Kopano server configuration files are stored in '**/etc/kopano**'.

- https://documentation.kopano.io/kopanocore_administrator_manual/

kopano-server.conf and ldap.conf

The option '**enable_hosted_kopano**' will switch on **multi-tenancy**. If you don't want this feature set it to 'false'.

The init script is broken in kopano server (debian bug and another debian bug), so we have to create our own systemd unit instead:

/etc/systemd/system/kopano-server.service

```
[Unit]
Description=Kopano Server
Before=kopano-dagent.service kopano-gateway.service kopano-ical.service kopano-monitor.service kopano-search.
service kopano-spamd.service kopano-spooler.service
After=network.target postfix.service

[Service]
Type=forking
AmbientCapabilities=CAP_NET_BIND_SERVICE
Environment=SERVER_CONFIG="/etc/kopano/server.cfg"
Environment=SERVER_OPTS=""
Environment=KOPANO_USERSCRIPT_LOCALE="C"
ExecStart=/usr/sbin/kopano-server -c ${SERVER_CONFIG}
ExecStop=/bin/kill -s TERM $MAINPID
PIDFile=/run/kopano/server.pid
TimeoutStopSec=60
Restart=always
User=kopano
Group=kopano
RuntimeDirectory=kopano
RuntimeDirectoryMode=2755

PrivateTmp=yes
PrivateDevices=yes
ProtectHome=yes
ReadOnlyDirectories=/
ReadWriteDirectories=-/var/lib/kopano
ReadWriteDirectories=-/var/log/kopano
ReadWriteDirectories=-/run/kopano
NoNewPrivileges=true
ProtectSystem=full

[Install]
WantedBy=multi-user.target
```

Afterwards, remove the current init script from systemd, reload the config, and re-enable it:

```
~# systemctl disable kopano-server.service
~# systemctl daemon-reload
~# systemctl enable kopano-server.service
```

⚠️ The "`createuser`" script references a wrong path in to the binary "`kscriptrun`" which breaks mailstore creation when running "`kopano-admin --sync`". Let's fix it. Open the script and change it:

/usr/lib/kopano/userscripts/createuser

```
# replace the last line in the script with this:
exec "/usr/lib/x86_64-linux-gnu/kopano/kscriptrun" /usr/lib/kopano/userscripts/createuser.d /etc/kopano
/userscripts/createuser.d
```

Now configure the kopano server:

/etc/kopano/server.cfg

```
server_listen           = 127.0.0.1:236
server_pipe_name        = /var/run/kopano/server.sock
server_pipe_priority    = /var/run/kopano/prio.sock
server_name             = kopanoserver
server_hostname         = mail.example.com
database_engine         = mysql
local_admin_users       = root kopano
system_email_address    = postmaster@example.com
pid_file                = /run/kopano/server.pid
log_method              = file
log_file                = /var/log/kopano/server.log
log_level               = 6
log_timestamp           = yes
log_buffer_size         = 0
mysql_host              = localhost
mysql_user              = kopano
mysql_password          = secret
mysql_socket            = /run/mysqld/mysqld.sock
mysql_database          = kopano
attachment_storage      = files
attachment_files_fsync  = yes
attachment_path         = /var/lib/kopano/attachments
attachment_compression  = 6
server_listen_tls       =
user_plugin             = ldap
user_plugin_config      = /etc/kopano/ldap.cfg
storename_format        = %f_%c
loginname_format        = %u

# set to 'true'  for  multi  tenancy!
enable_hosted_kopano    = false

# enable all 'features' per default
# possible values: imap pop3 mobile outlook webapp
disabled_features =
```

Copy the openldap configuration template to ldap.conf before you start editing.

We use the LDAP builtin "entryUUID" value as the unique identifier for users. The default "uidNumber" only works for posix accounts.

/etc/kopano/ldap.cfg

```
!include /usr/share/kopano/ldap.openldap.cfg
#!include /usr/share/kopano/ldap.active-directory.cfg

# LDAP host name/IP address
ldap_host = localhost

# LDAP port
# Optional, default = 389
# Use 636 for ldaps
ldap_port = 389

# LDAP search base
# specify this if you get weird LDAP authentication errors
# https://forum.kopano.io/topic/533/ad-and-kopano-not-working-after-migration-form-zarafa/2
#ldap_search_base = dc=example,dc=com

# The DN of the user to bind as for normal operations (not used for
# authentication if ldap_authentication_method is set to "bind".
# When empty, uses anonymous binding.
# The userPassword attribute must be readable for this user if the
# ldap_authentication_method option is set to password.
ldap_bind_user = cn=kopanoservice,dc=example,dc=com

# LDAP bind password
ldap_bind_passwd = secret

# When an object (user/group/company) is changed, this attribute will also change:
# Active directory: uSNChanged
# LDAP: modifyTimestamp
ldap_last_modification_attribute = modifyTimestamp

##########
# Object settings

# attribute name which is/(should: was) used in ldap_user_search_filter
ldap_user_type_attribute_value = kopano-user
ldap_group_type_attribute_value = kopano-group

##########
# User settings

ldap_user_search_filter = (objectClass=kopano-user)
ldap_user_unique_attribute = entryUUID

##########
# Group settings
ldap_group_search_filter = (objectClass=kopano-group)

##########
# Company settings
ldap_company_search_filter = (objectClass=kopano-company)

##########
# Quota settings
ldap_quota_multiplier = 1048576
```

Kopano-dagent

This is the service which talks to postfix and delivers mails to Kopano postboxes. The dagent must be enabled in Kopano's default file. We adjust some values for spam management here that will later tie in nicely with rspamd:

/etc/default/kopano

```
DAGENT_ENABLED=yes
DAGENT_CONFIG=/etc/kopano/dagent.cfg
DAGENT_OPTS="-d"
```

/etc/kopano/dagent.cfg

```
log_timestamp         = yes
log_buffer_size       = 0
lmtp_listen           = 127.0.0.1:2003
spam_header_name      = X-Spam-Status
spam_header_value     = Yes
```

Kopano-gateway

This service talks pops and imaps to the outside world. Unencrypted protocols should no longer be used. Enable them only if you need to support legacy clients.

/etc/kopano/gateway.cfg

```
log_method             = file
log_level              = 4
log_file               = /var/log/kopano/gateway.log
log_timestamp          = yes
log_buffer_size        = 0
#pop3_listen            = *:110
pop3s_listen           = *:995
#imap_listen            = *:143
imaps_listen           = *:993
server_socket          = http://localhost:236/
server_hostname        = example.com
imap_max_messagesize   = 20M
imap_max_fail_commands = 5
ssl_private_key_file   = /etc/ssl/private/mail.example.com/privkey.pem
ssl_certificate_file   = /etc/ssl/private/mail.example.com/fullchain.pem
ssl_protocols          = TLSv1.1 TLSv1.2
ssl_prefer_server_ciphers = yes
```

Kopano-ical

This is the iCal / CALDAV Server. The preferred method is CALDAV though.

/etc/kopano/ical.cfg

```
log_method           = file
log_level            = 4
log_file             = /var/log/kopano/ical.log
log_timestamp        = yes
log_buffer_size      = 0
server_timezone      = Europe/Berlin
ssl_private_key_file = /etc/ssl/private/mail.example.com/privkey.pem
ssl_certificate_file = /etc/ssl/private/mail.example.com/fullchain.pem
ssl_protocols        = TLSv1.1 TLSv1.2
ssl_prefer_server_ciphers = yes
```

Kopano-search

The search index which is utilized by Kopano for fulltext search in the mailstores. There are multiple options for indexing mail attachments under the "ATTACHMENT INDEX SETTINGS" paragraph. As they are pretty self-explanatory and heavily dependent of the use-case, I wont give any recommendations for them.

/etc/kopano/search.cfg

```
log_method          = file
log_level           = 4
log_file            = /var/log/kopano/search.log
log_timestamp       = yes
term_cache_size     = 256M
index_processes     = 2
index_drafts        = no
index_junk          = yes
index_attachments   = yes
index_attachment_max_size = 50M
```

Kopano-spamd

This service moves messages marked as spam (by moving it into the Junk folder) into a folder in the file system so rspamd can pick it up for spam training.

We will integrate rspamd later on, just configure the following settings for now:

/etc/kopano/spamd.cfg

```
run_as_user         = kopano
run_as_group        = _rspamd
log_method          = file
log_level           = 5
log_file            = /var/log/kopano/spamd.log
log_timestamp       = 1
sa_group            = _rspamd
header_tag          = X-Spam-Status
```

Then create the folders and update groups and folder permissions:

```
gpasswd -a kopano _rspamd
mkdir -p /var/lib/kopano/spamd/{spam,ham}
chown -R kopano:_rspamd /var/lib/kopano/spamd/{spam,ham}
chmod -R ug+rwX /var/lib/kopano/spamd/{spam,ham}
```

Kopano-spooler

This service delivers mail to postfix to be sent into the intertubes. Luckily nothing must be adjusted here except the user and group settings.

/etc/kopano/spooler.cfg

```
log_method          = file
log_level           = 4
log_file            = /var/log/kopano/spooler.log
log_timestamp       = yes
log_buffer_size     = 0
enable_dsn          = yes
```

Now enable and start all services:

```
~# systemctl enable kopano-server.service
~# systemctl enable kopano-dagent.service
~# systemctl enable kopano-gateway.service
~# systemctl enable kopano-ical.service
~# systemctl enable kopano-monitor.service
~# systemctl enable kopano-search.service
~# systemctl enable kopano-spamd.service
~# systemctl enable kopano-spooler.service

~# systemctl start kopano-server.service
~# systemctl start kopano-dagent.service
~# systemctl start kopano-gateway.service
~# systemctl start kopano-ical.service
~# systemctl start kopano-monitor.service
~# systemctl start kopano-search.service
~# systemctl start kopano-spamd.service
~# systemctl start kopano-spooler.service
```

Kopano Webapp

Create an apache vhost for the domain "webapp.example.com".

⚠️ Don't forget to replace the domain name and the path to the certificates accordingly!

/etc/apache2/sites-enabled # cat kopano.conf

```
<VirtualHost 1.2.3.4:80>

  ServerName      webmail.example.com
  ServerAdmin     webmaster@example.com

  RewriteEngine   On
  RewriteCond     %{HTTPS} off
  RewriteRule     ^(.*)$   https://%{SERVER_NAME}%{REQUEST_URI} [L,R]

</VirtualHost>

<VirtualHost 1.2.3.4:443>
  ServerName              webmail.example.com
  ServerAdmin             webmaster@example.com


  SSLEngine               on
  SSLCertificateFile      /etc/ssl/private/webmail.example.com/cert.pem
  SSLCertificateKeyFile   /etc/ssl/private/webmail.example.com/privkey.pem
  SSLCertificateChainFile /etc/ssl/private/webmail.example.com/chain.pem
  SSLCACertificateFile    /etc/ssl/private/webmail.example.com/fullchain.pem
  SSLProtocol             all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
  SSLHonorCipherOrder     on
  SSLCompression          off
  SSLSessionTickets       off
  SSLOpenSSLConfCmd       DHParameters "/etc/apache2/dhparams_4096.pem"
  SSLCipherSuite          ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-
POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256

  Protocols               h2 h2c http/1.1

  Include /etc/apache2/sites-available/kopano-webapp.conf
  Include /etc/apache2/conf-available/z-push.conf
  Include /etc/apache2/conf-available/z-push-autodiscover.conf

  ErrorLog  /var/log/apache2/kopano-error.log
  CustomLog /var/log/apache2/kopano-access.log combined

</VirtualHost>
```

Edit the kopano webapp include file:

/etc/apache2/sites-available/kopano-webapp.conf

```
# comment out this line:
#Alias /webapp /usr/share/kopano-webapp

# add this instead:
DocumentRoot /usr/share/kopano-webapp
```

Enable the vhost and restart apache:

```
~# a2ensite kopano.conf
~# apachectl configtest
~# systemctl restart apache2.service
```

The kopano webapp now should load (with a valid certificate) in your browser with this domain: "`https://webmail.example.com`".

rspamd

rpamd has become the swiss armyknife when it comes to spam filtering, virus scanning and so on. We will also install "inotify-spamlearn" so that mails that are tagged as "Junk" are automatically sent to rspamd for spam training.

The configuration can be quiet daunting at first, but is actually structured very well:

- ⚠️ Do not touch any config files in /etc/rspamd directly
- Instead put your own config files in either "**local.d**" or "**override.d**" folders
  - Explanation: https://rspamd.com/doc/faq.html#what-are-the-locald-and-overrided-directories
- Configuration results can be dumped at any time with the command "`rspamadm configdump`"

First, install rspamd:

```
apt install rspamd redis-server
```

Create a password for rspamd (take note of the displayed hash string, we need it later!)

```
~# rspamadm pw
```

Configure redis:

/etc/redis/redis.conf

```
requirepass secret      # <-- put a really long random string here!
maxmemory   256M        # set redis memory limit to something sane. depends on your environment and the load
that you are expecting
```

Afterwards restart redis:

```
~# systemct restart redis.service
```

Configure rspamd:

- For Antivirus integration with Virustotal you will need a free API key, obtainable by signing up to their service. If you don't want this, skip the "**antivirus.conf**" file
- For DKIM integration you will have to set up fitting DNS records. If you don't want DKIM, skip the "**dkim_signing.conf**" file.

Configuration files in local.d

List of all needed configuration files in **/etc/rspamd/local.d**:

antivirus.conf

```
# local.d/antivirus.conf
virustotal {
  # Obtained from Virustotal
  apikey = "<YOUR-VIRUSTOTAL-API-KEY>";
}
```

classifier-bayes.conf

```
backend = "redis";
new_schema = true;
expire = 315360000;     # 10 years
min_learns = 1;
store_tokens = true;
signatures = true;

# enable autolearn here if you want it:
autolearn = false;
autolearn {
  spam_threshold = 6.0; # When to learn spam (score >= threshold)
  ham_threshold = -0.5; # When to learn ham (score <= threshold)
  check_balance = true; # Check spam and ham balance
  min_balance = 0.9;    # Keep diff for spam/ham learns for at least this value
}
```

Greylisting isn't widely accepted anymore since it usually leads to mail delays and bots tend to dodge it more often these days.

If you still want it, set it to "`true`".

greylist.conf

```
enabled = false;
```

milter_headers.conf

```
authenticated_headers = ["authentication-results"];
use = ["x-spamd-result","x-spam-level","x-spamd-bar", "my-x-spam-score","x-spam-status", "authentication-
results"];

# add X-Spam-Score header (like SA does)
# Source: https://groups.google.com/forum/#!topic/rspamd/fEdbnG0J18I
custom {
   my-x-spam-score = <<EOD
     return function(task, common_meta)
       local sc = common_meta['metric_score'] or task:get_metric_score()
       -- return no error
       return nil,
       -- header(s) to add
       {['X-Spam-Score'] = string.format('%.2f', sc[1])},
       -- header(s) to remove
       {['X-Spam-Score'] = 1},
       -- metadata to store
       {}
```

```
   end
EOD;
}
```

mx_check.conf

```
enabled = true;
```

options.inc

```
local_addrs = "127.0.0.0/8, ::1";
```

phishing.conf

```
openphish_enabled = true;
phishtank_enabled = true;
```

redis.conf

```
servers = "127.0.0.1";
password = "<REDIS-PASSWORD>";     # the really long string we defined in as 'requirepass' in redis.conf
#disabled_modules = ["ratelimit"]; # List of modules that should not use redis from this section
timeout = 3s;
```

spamtrap.conf

```
enabled = true;
action = "no action";
score = 1.0;
learn_fuzzy = true;
learn_spam = true;
map = file://$LOCAL_CONFDIR/maps.d/spamtrap.map;
```

url_reputation.conf

```
enabled = true;
```

url_tags.conf

```
enabled = true;
```

worker-controller.inc

```
count = 2;
enable_password = "$2$rI8h.....";  # pw strng generated with 'rspamadm pw'
```

worker-fuzzy.inc

```
count = 2;
backend = "redis";
expire = 90d;
allow_update = ["127.0.0.1", "::1"];
```

worker-proxy.inc

```
spam_header = "X-Spam-Status";
```

Configuration files in override.d

List of all needed configuration files in **/etc/rspamd/override.d**:

milter_headers.conf

```
extended_spam_headers = true;
```

Setting up DKIM

Generate a new key with rspamdadm:

```
rspamadm dkim_keygen -b 2048 -s mail -k /var/lib/rspamd/dkim/mailexamplecom.key
```

DO NOT COPY THIS DNS ENTRY - it is just an example! USE the output of the command above!

The command will display a BIND DNS server compatible string like this:

```
mail._domainkey IN TXT ( "v=DKIM1; k=rsa; "

"p=MIIBIjANBgkqhkiG9w0BAQEXVAOCAQ8AMIIBCgKCAQEAwUhRCoEHrziLPw8IQmguUSpppl0sqQ8wX2lKlylBl0P5dqSKp+WVFUn8x6F4EJcxu
usV3anlgVH97xaSrX/KxM/08agJyd8l9FDwil/sBQGut4uvO/RXGztn0t81ziXXmjW9REQi73QkXA/umH4pNOzG
/PW3aIuYmAzs8OsfFGMlGqp3S2WhW47Sx42zEPPWJ+9Sew2qhtDTG8CS2"
        "AGaF5kWJD87DnV2McuRZTNcGcPLr
/Oa9I+acDl6d+lJxWb626tHitmrYRbVYJuAfSf6glcx64IdNokJOfzxHBlIcYP4HbTNrnP+oo9Yhir7fnd4Cs06ishAdGA9Olt0PKpl2wIDAQAB"
) ;
```

Now add the displayed output to your DNS server as a TXT record

If you have a BIND DNS server (or compatible DNS administration panel) you can copy-paste the displayed string and it will be picked up as a valid configuration automatically.

Several DNS administration panels like to break up longer strings into multiple lines or just silently discard parts of the string after a certain length.

Make sure the data is picked up by the DNS server correctly and/or contact your DNS support if you run into trouble.

Test if the entry is resolved in DNS correctly:

```
# install 'host' utility
~# apt install bind9-host

# test DKIM TXT DNS entry
~# host -t TXT mail._domainkey.example.net

# result:
mail._domainkey.example.com descriptive text "v=DKIM1; k=rsa; "
"p=p=MIIBIjANBgkqhkiG9w0BAQEXVAOCAQ8AMIIBCgKCAQEAwUhRCoEHrziLPw8IQmguUSpppl0sqQ8wX2lKlylBl0P5dqSKp+WVFUn8x6F4EJc
xuusV3anlgVH97xaSrX/KxM/08agJyd8l9FDwil/sBQGut4uvO/RXGztn0t81ziXXmjW9REQi73QkXA/umH4pNOzG
/PW3aIuYmAzs8OsfFGMlGqp3S2WhW47Sx42zEPPWJ+9Sew2qhtDTG8CS2" "TAGaF5kWJD87DnV2McuRZTNcGcPLr
/Oa9I+acDl6d+lJxWb626tHitmrYRbVYJuAfSf6glcx64IdNokJOfzxHBlIcYP4HbTNrnP+oo9Yhir7fnd4Cs06ishAdGA9Olt0PKpl2wIDAQAB"

# the result must display AS A SINGLE LINE with no linebreaks, starting with '"v=DKIM1; k=rsa; "'!
```

Now configure DKIM signing in rspamd:

/etc/rspamd/local.d/dkim_signing.conf

```
use_domain = "header";

# Enable DKIM signing for alias sender addresses
allow_username_mismatch = true;

domain {
  example.com {
    path = "/var/lib/rspamd/dkim/mailexamplecom.key";
    selector = "mail";
  }
}
```

```
# for ARC, just copy the DKIM config:
~# cp -a /etc/rspamd/local.d/dkim_signing.conf /etc/rspamd/local.d/arc.conf
```

Now restart rspamd:

```
~# systemctl restart rspamd.service
```

Use a tool like http://www.appmaildev.com/en/dkim to verify the DKIM keys work as expected

Integration with kopano-spamd

Install and configure inotify-spamlearn as described here: https://github.com/bkram/inotify-spamlearn

If you have everything set up, add the config file:

/etc/kopano/inotify-spamlearn.cfg

```
[paths]
# path to look for spam emls
spam_dir = /var/lib/kopano/spamd/spam
# path to look for ham emls
ham_dir = /var/lib/kopano/spamd/ham
```

```
[spam]
# command to pipe the spam into
spamcmd = /usr/bin/rspamc learn_spam
# command to pipe the ham into
hamcmd = /usr/bin/rspamc learn_ham

[mode]
# should we delete emls after processing
delete = True
# should we scan and process existing spam/ham in the watch directories
scan = True
# after scanning and processing of existing spam/ham exit the script. (crontab mode)
oneshot = False

[logging]
# levels possible are DEBUG INFO WARN ERROR CRITICAL
loglevel = INFO
# empty means log to console, use journalctl to read the logs when run from systemd service
logfile = /var/log/kopano/learnspam.log
```

Edit the systemd unit and change the "Group" to "_rspamd":

/etc/systemd/system/inotify-spamlearn.service

```
[Unit]
Description=Inotify Spamlearn
After=kopano-spamd.service

[Service]
User=kopano
Group=_rspamd
Type=simple
ExecStart= /usr/local/sbin/inotify-spamlearn.py
ExecReload=/bin/kill -HUP $MAINPID

[Install]
WantedBy=multi-user.target
```

Enable and start the service

```
~# systemctl daemon-reload
~# systemctl enable inotify-spamlearn.service
~# systemctl start inotify-spamlearn.service
```

postfix and sasl

postfix must be configured to do LDAP user lookups which is done via the config files **ldap-users.cf** and **ldap-aliases.cf**. Further main.cf and master.cf need to be adjusted to scan mail for spam and deliver them to dagent via lmtp in the end. To authenticate users who are sending mails from external addresses we will use SASL with the remote IMAP (rimap) method, which basically performs a IMAP-login to check if the supplied password is correct.

postfix

This configuration contains restrictive DNSBL postscreen filtering. This means that postfix will refuse to accept the mail in the first place and immediately send a 5** error code. This also means the mail won't ever be considered as delivered or show up in your mailqueue. Be sure to understand what this means and please read the manual!

This postfix config also contains enhanced recipient restrictions and mailq timeouts. If you do not know what they do you shouldn't use them.

/etc/postfix/main.cf

```
compatibility_level     = 2
smtpd_banner            = $myhostname ESMTP NO UCE
sendmail_path           = /usr/sbin/sendmail
newaliases_path         = /usr/bin/newaliases
mailq_path              = /usr/bin/mailq
myhostname              = example.com
biff                    = no
append_dot_mydomain     = no
mynetworks              = 127.0.0.0/8 [::1]/128 1.2.3.4
recipient_delimiter     = +
owner_request_special   = no
inet_interfaces         = all
inet_protocols          = ipv4
alias_maps              = hash:/etc/aliases

myorigin                = $myhostname
mydestination           = localhost


# virtual maps
virtual_mailbox_domains = mail.example.com mail2.example.com
virtual_mailbox_maps    = ldap:/etc/postfix/ldap-users.cf
virtual_alias_maps      = ldap:/etc/postfix/ldap-aliases.cf
virtual_transport       = lmtp:127.0.0.1:2003

# prevent leaking valid e-mail addresses
disable_vrfy_command    = yes

#sasl
smtpd_sasl_path                 = smtpd
smtpd_sasl_auth_enable          = yes
broken_sasl_auth_clients        = yes

# TLS encryption - server
smtpd_tls_security_level        = may
smtpd_tls_auth_only             = no
smtpd_tls_cert_file             = /etc/ssl/private/mail.example.com/fullchain.pem
smtpd_tls_key_file              = /etc/ssl/private/mail.example.com/privkey.pem
smtpd_tls_loglevel              = 0
smtpd_tls_received_header       = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source               = dev:/dev/urandom
smtpd_tls_dh1024_param_file     = /etc/postfix/dhparams_4096.pem
smtpd_tls_session_cache_database        = btree:${data_directory}/smtpd_scache
smtpd_tls_mandatory_exclude_ciphers     = aNULL, eNULL, EXPORT, DES, RC4, MD5, PSK, aECDH, EDH-DSS-DES-CBC3-
SHA, EDH-RSA-DES-CDC3-SHA, KRB5-DE5, CBC3-SHA
smtpd_tls_mandatory_protocols           = !SSLv2, !SSLv3
smtpd_tls_protocols                     = !SSLv2 !SSLv3

# TLS encryption - client
smtp_tls_security_level         = may
smtp_tls_loglevel               = 1
smtp_tls_mandatory_ciphers      = high
smtp_tls_mandatory_protocols    = !SSLv2, !SSLv3
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

### Before-220 tests
postscreen_dnsbl_reply_map      = pcre:/etc/postfix/postscreen_dnsbl_reply_map.pcre
postscreen_blacklist_action     = drop
postscreen_dnsbl_action         = enforce
postscreen_greet_action         = enforce
postscreen_dnsbl_threshold      = 4
postscreen_dnsbl_sites =
        zen.spamhaus.org*3
        b.barracudacentral.org*2
        bl.spameatingmonkey.net*2
        z.mailspike.net*2
        bl.mailspike.net
```

```
        bl.spamcop.net
        dnsbl.sorbs.net
        psbl.surriel.com
        swl.spamhaus.org*-4
postscreen_whitelist_interfaces = $mynetworks, static:all

# spam filter and DKIM signatures via rspamd
smtpd_milters         = inet:localhost:11332
non_smtpd_milters     = inet:localhost:11332
milter_protocol       = 6
milter_mail_macros    = i {mail_addr} {client_addr} {client_name} {auth_authen}
milter_default_action = accept

# check incoming mail for 'stuff'
smtpd_recipient_restrictions =
        permit_mynetworks,
        permit_sasl_authenticated,
        reject_unknown_recipient_domain,
        reject_non_fqdn_recipient,
        reject_unauth_destination,
        reject_unauth_pipelining,
        reject_invalid_hostname

smtpd_data_restrictions =
        reject_unauth_pipelining

# client restrictions
smtpd_client_restrictions =
        permit_mynetworks,
        permit_auth_destination,
        permit_sasl_authenticated,

# anybody out there?
smtpd_helo_restrictions =
        permit_mynetworks,
        reject_invalid_hostname

# who may send
smtpd_sender_restrictions =
        permit_mynetworks,
        permit_sasl_authenticated,
        reject_unknown_sender_domain,

# mail reject codes
unknown_address_reject_code             = 550
unknown_client_reject_code              = 550
unknown_hostname_reject_code            = 554
unknown_local_recipient_reject_code     = 550
unknown_relay_recipient_reject_code     = 554
unknown_virtual_alias_reject_code       = 550
unknown_virtual_mailbox_reject_code     = 550

# deferred mail intervals

queue_run_delay                 = 900
# (default: 300 seconds; before Postfix 2.4: 1000s)
# How often the queue manager scans the queue for deferred mail.

# (default: 300 seconds; before Postfix 2.4: 1000s)
# The minimal amount of time a message won't be looked at, and the minimal amount of time to stay away from a
"dead" destination.
minimal_backoff_time            = 450

# (default: 4000 seconds)
# The maximal amount of time a message won't be looked at after a delivery failure.
maximal_backoff_time            = 1800

# (default: 5 days)
# How long a message stays in the queue before it is sent back as undeliverable. Specify 0 for mail that should
be returned immediately after the first unsuccessful delivery attempt.
maximal_queue_lifetime          = 14
```

```
# (default: 5 days, available with Postfix version 2.1 and later)
# How long a MAILER-DAEMON message stays in the queue before it is considered undeliverable. Specify 0 for mail
that should be tried only once.
bounce_queue_lifetime           = 14

# (default: 20000)
# The size of many in-memory queue manager data structures. Among others, this parameter limits the size of the
short-term, in-memory list of "dead" destinations. Destinations that don't fit the list are not added.
qmgr_message_recipient_limit    = 1000000

# mail size
message_size_limit              = 21360000
```

/etc/postfix/postscreen_dnsbl_reply_map.pcre

```
# We will be rejecting much mail which is listed in multiple DNSBLs.
# We're not proud of some of the lists we are using, thus have given
# them lower scores in postscreen_dnsbl_sites listing. So this checks
# the DNSBL name postscreen(8) gets from dnsblog(8), and if it's not
# one of our Tier 1 DNSBL sites, it changes what the sender will see:

!/^zen\.spamhaus\.org$/         multiple DNS-based blocklists
```

master.cf

The master transport file. Transport decisions are made here, in our case:
internet > postscreen (DNSBL) -> smpt:postfix -> rspamd -> smtp:postfix -> lmtp:dagent

> **Notice:** lmtp must not run chrooted (look at the fifth column), otherwise it won't be able to talk to dagent!

/etc/postfix/master.cf

```
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (no)    (never) (100)
# ==========================================================================
#smtp      inet  n       -       y       -       -       smtpd
smtp      inet  n       -       y       -       1       postscreen
smtpd     pass  -       -       y       -       -       smtpd
dnsblog   unix  -       -       y       -       0       dnsblog
tlsproxy  unix  -       -       y       -       0       tlsproxy
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=$mua_sender_restrictions
  -o smtpd_recipient_restrictions=
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
smtps     inet  n       -       y       -       -       smtpd
  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_reject_unlisted_recipient=no
  -o smtpd_client_restrictions=$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
#628       inet  n       -       y       -       -       qmqpd
pickup    unix  n       -       y       60      1       pickup
cleanup   unix  n       -       y       -       0       cleanup
qmgr      unix  n       -       n       300     1       qmgr
#qmgr     unix  n       -       n       300     1       oqmgr
tlsmgr    unix  -       -       y       1000?   1       tlsmgr
rewrite   unix  -       -       y       -       -       trivial-rewrite
bounce    unix  -       -       y       -       0       bounce
defer     unix  -       -       y       -       0       bounce
trace     unix  -       -       y       -       0       bounce
verify    unix  -       -       y       -       1       verify
flush     unix  n       -       y       1000?   0       flush
proxymap  unix  -       -       n       -       -       proxymap
proxywrite unix -       -       n       -       1       proxymap
smtp      unix  -       -       y       -       -       smtp
relay     unix  -       -       y       -       -       smtp
        -o syslog_name=postfix/$service_name
#       -o smtp_helo_timeout=5 -o smtp_connect_timeout=5
showq     unix  n       -       y       -       -       showq
error     unix  -       -       y       -       -       error
retry     unix  -       -       y       -       -       error
discard   unix  -       -       y       -       -       discard
local     unix  -       n       n       -       -       local
virtual   unix  -       n       n       -       -       virtual
lmtp      unix  -       -       y       -       -       lmtp
anvil     unix  -       -       y       -       1       anvil
scache    unix  -       -       y       -       1       scache
#
# maildrop. See the Postfix MAILDROP_README file for details.
# Also specify in main.cf: maildrop_destination_recipient_limit=1
#
maildrop  unix  -       n       n       -       -       pipe
  flags=DRhu user=vmail argv=/usr/bin/maildrop -d ${recipient}
#
# See the Postfix UUCP_README file for configuration details.
#
uucp      unix  -       n       n       -       -       pipe
  flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail ($recipient)
```

ldap users and aliases

These files control ldap user and alias lookups

/etc/postfix/ldap-users.cf

```
server_host     = localhost
search_base     = dc=example,dc=com
version         = 3
bind            = yes
bind_dn         = cn=kopanoservice,dc=example,dc=com
bind_pw         = secret
scope           = sub
query_filter    = (mail=%s)
result_attribute        = mail
```

/etc/postfix/ldap-aliases.cf

```
server_host     = localhost
search_base     = dc=example,dc=com
version         = 3
bind            = yes
bind_dn         = cn=kopanoservice,dc=example,dc=com
bind_pw         = secret
```

```
scope           = sub
query_filter    = (kopanoAliases=%s)
result_attribute        = mail
```

Test LDAP connectivity

Test your setup by issuing the following commands.

In both test cases the email address of the user should be displayed.

If this doesn't work then postfix won't be able to deliver email as it can't find email addresses or aliases in the LDAP directory!

```
~# postmap -q <insert-username-here>@example.com ldap:/etc/postfix/ldap-users.cf
~# postmap -q <insert-user-alias-here>@example.com ldap:/etc/postfix/ldap-aliases.cf
```

SASL

Sadly, the saslauthd init script is not working correctly on Debian10. We have to create our own systemd unit to make it work:

/etc/systemd/system/saslauthd.service

```
[Unit]
RequiredBy=postfix.service
Before=runlevel2.target runlevel3.target runlevel4.target runlevel5.target shutdown.target
After=local-fs.target remote-fs.target
Conflicts=shutdown.target

[Service]
User=root
Group=sasl
Type=forking
Restart=no
TimeoutSec=5min
IgnoreSIGPIPE=no
ExecStart=/usr/sbin/saslauthd -a rimap -O 127.0.0.1 -c -m /var/spool/postfix/var/run/saslauthd -r -n 0

[Install]
WantedBy=multi-user.target
```

Register the service and start it:

```
systemctl daemon-reload
systemctl enable saslauthd.service
systemctl start saslauthd.service
```

/etc/postfix/sasl/smtpd.conf

```
pwcheck_method: saslauthd
mech_list: plain login
```

Restart the services afterwards:

add postfix to sasl

```
~# gpasswd -a postfix sasl
~# mkdir -p /var/spool/postfix/var/run/saslauthd
~# chown postfix:sasl /var/spool/postfix/var/run/saslauthd
~# systemctl restart postfix
~# systemctl restart saslauthd
```

Regular Maintenance

Drop this file into '/etc/cron.d'. It's pretty self-explanatory, however you might want adjust the values and/or intervals.

/etc/cron.d/mailjobs

```
# minute (0-59),
# |     hour (0-23),
# |     |       day of the month (1-31),
# |     |       |       month of the year (1-12),
# |     |       |       |       day of the week (0-7 with 0=7=Sunday).
# |     |       |       |       |       user
# |     |       |       |       |       |       command

# sync LDAP to Kopano regularly
*/30    *       *       *       *       root    /usr/sbin/kopano-admin --sync

# purge soft-deleted items after 30 days
30      3       *       *       *       root    /usr/sbin/kopano-srvadm --purge-softdelete=30
```

⚠️ **kopano-srvadm** expects the file **/etc/kopano/admin.cfg** for some reason, otherwise it will throw an error. Just creating an empty file by that name suffices to make **kopano-srvadm** work as expected:

```
~# touch /etc/kopano/admin.cfg
```

Testing / Verifying the setup

- Send Mails via
  - WebAccess / WebApp
  - pop/pops
  - imap/imaps
  - ActiveSync ("Exchange" on mobile phones)
  - postscreen checks
  - rspamd
  - spam/ham learning
- Deliver mails
  - test local delivery to username / alias
  - test group deliveries
- exchange / activesync / mobile access

Contact

Additions, comments, critisicm? mail to: b2c[at]dest-unreachable.net