Kopano Migration Manual

Release 7.2.1

Kopano BV

Jun 01, 2018

Contents

This document, the Kopano Migration Manual, describes how to migrate to Kopano using various scenarios.

CHAPTER 1

Introduction

Migrating to Kopano is performed by either:

- Using the Zarafa2Kopano script for migrations of Zarafa to Kopano
- Importing `.pst` files with the kopano-pst-import utility (planned for Q3/2016) for any other Outlook or PST-capable groupware solution
- Direct import from MS Exchange to Kopano (Toolset available from Kopano professional Services)
- Importing via IMAP and CalDAV

1.1 Professional Services Toolset / Migration Tool

Professional Services can provide you with a Migration Tool that performs a migration by opening each user's store in the data source (Outlook PST format), traveling over the MAPI folder structure migrating each item and folder. This means that all data from the data source will be migrated, including mail, attachments, appointments, tasks, contacts etc. The Tool will also check if the item/folder already exists at the destination (a Kopano Server or a `.pst` file) to avoid duplicates. The tool will not create users within Kopano, it assumes that the users to which the administrator wants to migrate data to, already exist at the Kopano Core. In the future, Kopano will provide a Toolchain to import with the same import capabilities - based on an extra provided package by Kopano.

1.1.1 Exchange to Kopano Migration

The Kopano Migration Tool can migrate multiple stores from one MS Exchange server to one Kopano Server.

CHAPTER 2

IMAP to Kopano Migration

`kopano-migration-imap` is based on well known imapsync, and is a utility for easily migrating from about any IMAP server to Kopano. It synchronizes all folders and items from the existing IMAP inbox to a given mailbox availble from the Kopano IMAP gateway.

The syntax to use kopano-migration-imap is as followed:

```
kopano-migration-imap --noauthmd5  --host1 old-imap.example.com --user1 userold --
↪password1 passw1  -sep2 / --prefix2 "" --host2 kopano.example.com --user2␣
↪usernew --password2 passw2
```

Description of the extra options: - The option –noauthmd5 forces authentication with plain password, which the Kopano gateway only supports - The option –sep2 / is to define the used separator for Kopano - The option –prefix2 “” is to define the basic starting folder - The option –allowsizemismatch is necessary when migrating from google/gmail accounts

Important: If you don’t know all user password in Kopano, you can change in the gateway.cfg the server_socket to *file:///var/run/kopano/server.sock*. This change allows users to login without a password. Make sure to switch this option back to http://localhost:236/kopano after the migration! Note: The password is plain text and is readable, to change this, it is possible to use ssl, when you put in extra options (Read: man imapsync)

Important: When installing *kopano-migration-imap* on RHEL the EPEL repositories are needed to fullfil all dependencies.

Please refer to https://imapsync.lamiral.info/doc/TUTORIAL_Unix.html for detailed usage instructions.

CHAPTER 3

PST to Kopano Migration

kopano-migration-pst can import multiple .pst files to the stores of the users on the Kopano Server. There are many ways to retrieve the .pst files with the users store. You might work with .pst files stored on clients separately, centrally stored on a server or use a groupware solution.

Microsoft has made a very useful tool, called “ExMerge”, and since MS Exchange 2010 SP1, the commandlet New-MailboxExportRequest allows exports of Mailboxes to PST format. These .pst files can be imported into an Exchange mail server, and more importantly, allows one to export an Exchange mail server to .pst files.

The Migration Tool is then used in the next stage to import these .pst files onto the Kopano Server.

```
$ kopano-migration-pst --help
Usage: kopano-pst PATH -u NAME
Options:
-h, --help show this help message and exit
-c FILE, --config=FILE load settings from FILE
-s SOCKET, --server-socket=SOCKET connect to server SOCKET
-k FILE, --ssl-key=FILE SSL key file
-p PASS, --ssl-pass=PASS SSL key password
-U NAME, --auth-user=NAME login as user
-P PASS, --auth-pass=PASS login with password
-u NAME, --user=NAME run program for specific user
-f NAME, --folder=NAME run program for specific folder
-l LEVEL, --log-level=LEVEL set log level (CRITICAL, ERROR, WARNING,
INFO, DEBUG)
--stats list folders for PATH
--index list items for PATH
--import-root=PATH list items for PATH
```

To import a PST file, you can run the following command:

```
$ kopano-migration-pst backup.pst -u user1
```

Since this is a command line tool, it gives the administrator the flexibility to script a batched import for multiple stores.

CHAPTER 4

Drop-in Migration from Zarafa

Since Kopano Core is a fork from ZCP, it is relatively easy to migrate to Kopano Core. Depending on the version that has been used in ZCP, not many changes are necessary to be done to run the latest version of Kopano Core.

4.1 Disclaimer & Preface

- We provide this information as-is, please understand each step you are taking first. The migration is at your own risk and please execute the steps only if you understand what you are doing.
- These instructions are intended as pointers and not as a definitive configuration, as every installed environment can be set up differently, based on the actions that have been done with the existing installation in the past.
- Please keep in mind that some steps in this guide can take time depending on the size and complexity of your environment.

Kopano Core is a fork of ZCP 7.2.3. Migrating from ZCP to Kopano Core means that some features and components are replaced and some are dropped If you are migrating from ZCP 7.1 to KC 8.x, please take the following into account before upgrading:

- Backups are not compatible. We recommend our users to make a MySQL backup before starting the migration procedure.
- Kopano introduces a new search component with a new index format. Indexes have to be rebuilt, which can take a considerable amount of time.
- Kopano Core is MAPI based. WebApp, Z-Push and other MAPI based clients (such as Zarafa MAPI client) can be used.
  - Note: not all versions of the Zarafa MAPI Outlook client is compatible with Kopano Core. Make sure all clients are upgraded to the latest version before migrating your Zarafa server to Kopano
- Optional, but highly recommended steps:
  - Run the daemon as an unprivileged user. Please note this means that the attachements directory needs to be ‘chowned’. This can take quite some time.
  - The socket location has changed to implement support for systemd (also required to run daemons as unpriviledged users.
- Also note: you can use the ZCP LDAP scheme with its attributes. Keep in mind that you might need to migrate the schema if new features require schema changes in the future.

4.2 Manual ZCP to Kopano migration

4.2.1 Backup of configuration files

In any case, we recommend you to do a backup of your configuration files - this helps later in making sure the right paths are used. Making a backup can easily be done by executing:

```
cp -R /etc/zarafa /etc/zarafa.bak
```

4.2.2 Removal of ZCP packages

As a first step in the manual migration, it is recommended to remove all existing ZCP packages. This operation depends on the distribution that is used.

For Debian-based distributions (Debian and Ubuntu) we recommend the following command to remove all ZCP packages:

```
dpkg -l | egrep '(mapi|gsoap|libical|vmime|zarafa)' | awk '{ print $2 }' | xargs␣
↪apt-get remove
```

For SUSE-based distributions we recommend the following command to remove all ZCP packages:

```
rpm -qa | egrep '(mapi|gsoap|libical|vmime|zarafa)' | awk '{ print $1 }' | xargs␣
↪zypper rm
```

For Red Hat-based distributions we recommend the following command to remove all ZCP packages:

```
rpm -qa | egrep '(mapi|gsoap|libical|vmime|zarafa)' | awk '{ print $1 }' | xargs␣
↪yum remove
```

Also note: the github version of python-zarafa is not compatible with python-kopano so remove any previous version of it (usually located in for example /usr/local/lib/python2.7/dist-packages/zarafa)

4.2.3 Installation of Kopano packages

After the packages have been removed, it is time to install the new kopano packages. This can be done by either installing the repositories provided by Kopano or by downloading and installing the packages directly from the Downloads section provided by Kopano.

Note: Please be aware that special characters in the username or password like ”!” can cause issues

Use the following examples to configure repositories for your operating system:

Debian 8 - /etc/apt/sources.list.d/kopano.list:

```
deb https://<username>:<password>@download.kopano.io/supported/core:/final/Debian_
↪8.0/ ./
```

For Debian-based distributions, we recommend to import the key as well:

```
curl https://<username>:<password>@download.kopano.io/supported/core:/final/Debian_
↪8.0/Release.key | apt-key add -
```

And for Red Hat Enterprise Linux 7 - /etc/yum.repos.d/kopano.repo:

```
[Kopano-Core]
name=Kopano Final release builds
type=rpm-md
baseurl=https://<username>:<password>@download.kopano.io/supported/core:/final/
↪RHEL_7/
gpgcheck=0
gpgkey=
enabled=1
```

SUSE Linux Enterprise Server 12 - /etc/zypp/repos.d/Kopano.repo:

```
[Kopano-Core]
enabled=1
autorefresh=0
baseurl=https://<username>:<password>@download.kopano.io/supported/core:/final/SLE_
↪12/
type=rpm-md
```

You can manually pick the packages you want to install, or install the meta package 'kopano-server-packages'. The meta package will install: (alphabetical order):

- kopano-backup
- kopano-dagent
- kopano-gateway
- kopano-ical
- kopano-monitor
- kopano-presence
- kopano-search
- kopano-server
- kopano-spooler
- kopano-utils

The installation can be done by the native package manager utilities, e.g.

Debian / Ubuntu:

```
apt-get install kopano-server-packages
```

SUSE Linux Enterprise Server:

```
zypper install kopano-server-packages
```

Red Hat / CentOS:

```
yum install kopano-server-packages
```

4.2.4 Configuration file takeover

With the installation of Kopano Core, the Kopano default configuration directory /etc/kopano is created. There are now 2 approaches in adapting configuration to Kopano: a) making a diff of the configuration files from /etc/zarafa to /etc/kopano, effectively migrating only the wanted configuration directives to Kopano or b) removing the per-default installed configuration files from Kopano, copying the existing Zarafa configuration files and making smaller changes to it

We do recommend a), as this is the cleanest approach in migration, and the default values for most entries (such as socket locations, etc.) are already in there. There is yet one exception, which is in case of using the LDAP backend: The default schema of Kopano has the “kopano” schema prefix, whereas the default schema of Zarafa uses the “zarafa” prefix for attributes. If the installation is to be migrated to the new Kopano LDAP schema as well, then of course also the default configuration files from Kopano can be uses as configuration basis.

When using b) all configuration settings can be taken over, just by effectively removing the kopano configuration files first, and then copying over the Zarafa configuration files, which is done by executing:

```
cp -a /etc/kopano{,-orig}; rm -rf /etc/kopano/*; cp -R /etc/zarafa/* /etc/kopano/
```

Please note, that there might be (depending on the list of Zarafa components installed before) the neccesity to fix certain path symlinks, such as /etc/zarafa/webapp originally pointing to /usr/share/zarafa-webapp which in Kopano is installed (per default) as /usr/share/kopano-webapp. A copy of the original files is kept in the folder /etc/kopano-orig for easier comparison.

Most daemons are now started per default as user “kopano”. Even if (which is the default) the repspective run_as_* directives are commented out, they are started per default as this unprivileged user. If desired, this can be changed to any other user, though this is not recommended, as permission changes would be required (if not “root”) with every update.

Default run-as daemon settings in Kopano Core:

```
# run_as_user = kopano
# run_as_group = kopano
```

Additionally, in Kopano Core the default socket locations have changed to the directory “/var/run/kopano”. Please check in the configuration files that the directives server_socket and pid_file are located under /var/run/kopano.

4.2.5 Renamed configuration directives

If the migration of the configuration files have been done by copying the original Zarafa configuration files, some minor attributes need to be renamed, as the original attributes by their name do not exist in Kopano Core. These are: enable_distributed_zarafa, enable_hosted_zarafa from server.cfg.

4.2.6 Move of attachment directory

Moving attachments from one directory to another is not technically required, they can still be pointed to the existing Zarafa directory. For sanity reasons, we though still recommend to move the directory contents under the new Kopano attachment location, if no special configuration is used. This allows out-of-the-box configurations also from security mechanisms like AppArmor or SELinux to allow access to the directories as they are intended to.

In default configurations, the attachment directory is /var/lib/zarafa/attachments, with Kopano the default attachment path is to be found at /var/lib/kopano/attachments. There are multiple possibilities to move the attachments to the new directory. If the directory is mounted in as a separate disk or network based mount (NFS, iSCSI block device, etc.) it is just required to change the corresponding mount point of the directory via the normal OS methods (mostly /etc/fstab).

If the attachments are in the default location, they can be easily moved by executing:

```
mv /var/lib/zarafa/attachments/* /var/lib/kopano/attachments/
```

If both directories reside on the same mountpoint the move operation will be executed quite quick, regardless of the size of the attachment store. Please make sure server.cfg has the correct attachment path specified with “attachment_path”.

4.2.7 Filesystem Permissions

With ZCP 7.2.2 the default user and group “zarafa” became the default for non-privileged usage. With Kopano Core, the default user and group are named “kopano”. All files and locations that have been owned by zarafa require therefore to be re-owned by the kopano user and group.

Please take care that the following directories have the correct permissions set:

- /etc/kopano (+ config files, + ssl (and optionally kerberos) files)
- /var/lib/kopano (+ subdirectories like attachments)
- /var/log/kopano (+ files located in this directory)
- /tmp
- Kopano attachment directory (configured with "attachment_path" in server.cfg)
- Kopano search directory (configured with "index_path" in search.cfg)

The correct permissions of config files for unprivileged users are 640 and root:kopano (owner (root): read and write, kopano (services): read, other: none). With these permissions the highest security for confidential credentials are maintained while the kopano services are able to read (but not write) their configuration.

By example these rights can be made by executing:

```
find /etc/kopano -type f -iname "*.cfg" -exec chown root:kopano {} \;
find /etc/kopano -type f -iname "*.cfg" -exec chmod 640 {} \;
chmod 755 /etc/kopano

chown -R root:kopano /etc/kopano/ssl
chmod 750 /etc/kopano/ssl
chmod 640 /etc/kopano/ssl/*
```

An example of changing ownership from zarafa to kopano in the example case of attachments can be done by executing (To attachments write access is required by the user "kopano"):

```
find /var/lib/kopano/attachments -user zarafa -exec chown kopano {} \;
find /var/lib/kopano/attachments -group zarafa -exec chown :kopano {} \;
```

Note: Please note that ZCP 7.1 Search index-files are not compatible with Kopano Core. ZCP 7.2 Search index files are compatible and require to be made available with correct path and permissions (same as with attachments).

4.2.8 LDAP schema

Kopano Core is capable to use the zarafa schema, however it is highly recommended that you import and use the kopano-schema: With new schema updates made available to Kopano, these are then correctly referenced to the Kopano schema.

If you want to continue to use the old Zarafa schema you can reuse the 'ldap.cfg' and 'ldap.propmap.cfg' from your old Zarafa installation.

4.3 Multi-server information

4.3.1 LDAP data transplantation to new schema

On OpenLDAP-based systems it is quite easily possible to migrate also the LDAP-based attribute data to the new Kopano LDAP-schema. To execute these changes a downtime to the LDAP master is required though.

Note: This operation should be done with care, we recommend to test the procedure for your specific case as well. Wrong commands can result in the loss of your directory data. The mentioned operations will make your

LDAP temporarily unavailable, so any other services using that LDAP also for authentication will therefore not be able to login.

The following commands execute essentially the following:

- Shut down of the LDAP service in a consistent state
- Output the LDAP database information, replace any occurence of “zarafa” in an attribute to “kopano”
- Dump of the existing LDAP database
- Import of saved LDAP dataset
- Re-index of LDAP based attributes (for performance)

```
<shutdown of LDAP service>
slapcat | sed  's#^zarafa#kopano#' > zarafa2kopano.ldif
<removal of existing LDAP DB, most likely /var/lib/ldap>
slapadd << zarafa2kopano.ldif
```

After these steps have been made and the schema attributes for Kopano schema have also been marked as to be indexed (either slapd.conf or via configuration schema), it is recommended to update the LDAP index by executing:

```
slapindex
```

After these changes have been made, the LDAP server can be restarted again, and any potential replication slave can be re-replicated by removing the LDAP-DN-tree database and restarting LDAP after schema installation.

On other systems this task can be a bit tedious, one possibility is to continue with the Zarafa LDAP schema and adapting to (very seldom happening schema update changes) or alternatively installing the new schema and transplanting the data on attribute basis.

An alternative to the downtime-required approach above, it is possible to migrate the attribute from the Zarafa schema to the Kopano schema. Please note, that before executing this below script, we strongly advise you to backup your LDAP directory (for example with slapcat).

This approach works for any LDAPv3-based Installation, including Active Directory, as long as both schemas exist in the directory.

```
#!/bin/bash

LDAP_HOST="localhost"
LDAP_USER="cn=root,dc=kopano,dc=com"
LDAP_PASSWORD="ThisIsMySecretPassword"
LDAP_LDIF="/tmp/zarafa2kopano.ldif"

grep ^attributetype /usr/share/doc/packages/zarafa/zarafa.schema -A1 | grep NAME |␣
↪sed -e 's#.*NAME..##g' -e 's#.$##' | while read line; do
       new_attrib="`echo $line | sed 's#zarafa#kopano#'`"
       ldapsearch -o ldif-wrap=no -LLL -x $line=* dn $line | sed "s#$line: \(.*\)
↪#changetype: modify\nadd: $new_attrib\n$new_attrib: \1\n-\ndelete: $line\n$line:␣
↪\1\n-#" >> "$LDAP_LDIF".step2_attribute_data
done


# We now have the LDIF, we now need to extend the objects with the new objectClass.
grep ^dn "$LDAP_LDIF".step2_attribute_data | sort | uniq | sed 's#dn: ##' | while␣
↪read dn; do ldapsearch -o ldif-wrap=no -LLL -x -s base -b "$dn" objectclass |␣
↪egrep '(^dn|zarafa)' | sed -e '/^objectClass/  s#zarafa#kopano#' | sed 's
↪#objectClass:\(.*\)#changetype: modify\nadd: objectClass\nobjectClass:\1\n#';␣
↪done >> "$LDAP_LDIF".step1_objectclass_extend
ldapmodify -h "$LDAP_HOST" -D "$LDAP_USER" -w "$LDAP_PASSWORD" -f "$LDAP_LDIF".
↪step1_objectclass_extend
```

```
# We are done with the LDIF, now lets do it.
ldapmodify -h "$LDAP_HOST" -D "$LDAP_USER" -w "$LDAP_PASSWORD" -f "$LDAP_LDIF".step2_attribute_data

grep ^dn "$LDAP_LDIF".step2_attribute_data | sort | uniq | sed 's#dn: ##' | while read dn; do ldapsearch -o ldif-wrap=no -LLL -x -s base -b "$dn" objectclass | egrep '(^dn|zarafa)' | sed 's#objectClass:\(.*\)#changetype: modify\ndelete: objectClass\nobjectClass:\1\n#'; done >> "$LDAP_LDIF".step3_objectclass_remove
ldapmodify -h "$LDAP_HOST" -D "$LDAP_USER" -w "$LDAP_PASSWORD" -f "$LDAP_LDIF".step3_objectclass_remove

# Clean up
rm -f "$LDAP_LDIF.step1_objectclass_extend"
rm -f "$LDAP_LDIF.step2_attribute_data"
rm -f "$LDAP_LDIF.step3_objectclass_remove"
```

This script changes all zarafa-based objects, regardless of type, including: Users, Groups, Contacts and Servers

4.3.2 Socket change in LDAP

Each server object in Kopano has a socket definition defined by the attribute zarafaFilePath/kopanoFilePath. This attribute should be replaced to the new value of the kopano-server socket, per default located at /var/run/kopano/server.sock - instead of /var/run/zarafad/server.sock (or for older installations /var/run/zarafa-server.sock).

CHAPTER 5

Legal Notice

Adobe, Acrobat, Acrobat Reader and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apache is a trademark of The Apache Software Foundation.

Apple, Mac, Macintosh, Mac OS, iOS, Safari and TrueType are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Blackberry is the trademark or registered trademark of BlackBerry Limited, the exclusive rights to which are expressly reserved. Kopano is not affiliated with, endorsed, sponsored, or otherwise authorized by BlackBerry Limited.

Collax is a trademark of Collax GmbH.

Debian is a registered trademark of Software in the Public Interest, Inc.

ECMAScript is the registered trademark of Ecma International.

Gentoo is a trademark of Gentoo Foundation, Inc.

Google, Android and Google Chrome are trademarks or registered trademarks of Google Inc.

IBM and PowerPC are trademarks of International Business Machines Corporation in the United States, other countries, or both.

MariaDB is a registered trademark of MariaDB Corporation AB.

Microsoft, Microsoft Internet Explorer, the Microsoft logo, the Microsoft Internet Explorer logo, Windows, Windows Phone, Office Outlook, Office 365, Exchange, Active Directory and the Microsoft Internet Explorer interfaces are trademarks or registered trademarks of Microsoft, Inc.

Mozilla, Firefox, Mozilla Firefox, the Mozilla logo, the Mozilla Firefox logo, and the Mozilla Firefox interfaces are trademarks or registered trademarks of Mozilla Corporation.

MySQL, InnoDB, JavaScript and Oracle are registered trademarks of Oracle Corporation Inc.

NDS and eDirectory are registered trademarks of Novell, Inc.

NGINX is a registered trademark of Nginx Inc. NGINX Plus is a registered trademark of Nginx Inc.

Opera and the Opera “O” are registered trademarks or trademarks of Opera Software AS in Norway, the European Union and other countries.

Postfix is a registered trademark of Wietse Zweitze Venema.

QMAIL is a trademark of Tencent Holdings Limited.

Red Hat, Red Hat Enterprise Linux, Fedora, RHCE and the Fedora Infinity Design logo are trademarks or registered trademarks of Red Hat, Inc. in the U.S. and other countries.

SUSE, SLES, SUSE Linux Enterprise Server, openSUSE, YaST and AppArmor are registered trademarks of SUSE LLC.

Sendmail is a trademark of Sendmail, Inc.

UNIX is a registered trademark of The Open Group.

Ubuntu and Canonical are registered trademarks of Canonical Ltd.

Univention is a trademark of Ganten Investitions GmbH.

All trademarks are property of their respective owners. Other product or company names mentioned may be trademarks or trade names of their respective owner.

Disclaimer: Although all documentation is written and compiled with care, Kopano is not responsible for direct actions or consequences derived from using this documentation, including unclear instructions or missing information not contained in these documents.