

KOLAB

URLs

- [Web Mail](#)
- [Admin Backend](#)
- [Global Ressources](#)
- [File Sharing](#)

Clients konfigurieren

Um per CALDAV oder CARDAV Daten mit dem Server zu synchronisieren sind folgende Einstellungen nötig:

| Service | URL | Name / User |
|---------------------------------------|---|--------------------|
| Perönliche Kalender / Kontakte | https://mail.moppert.de/iRony/ | lutz@moppert.de |
| Gemeinsame Kalender / Kontakte | https://mail.moppert.de/iRony/ | familie@moppert.de |
| MopBox | https://mopbox.moppert.de:8443/ | lutz@moppert.de |


Security



Siehe <https://weakdh.org/>

[Let's Encrypt certificates for mail servers and DANE](#)

Datensicherung

Der Server hat folgende Platten Aufteilung, dies folgt grob den Empfehlungen der CIS Benchmark kombiniert mit den Empfehlungen für Kolab:

| Dateisystem | Größe | 07-2019 | Eingehängt auf | Daten |
|--------------------------|-------|---------|----------------|--|
| /dev/dm-0 | 9,8G | | / | Alles andere |
| udev | 10M | 0 | /dev | Geräte-Dateien |
| tmpfs | 1,2G | 39M | /run | Laufzeitinformationen (z.B. PIDs) |
| tmpfs | 3,0G | 40k | /dev/shm | Shared Memory |
| tmpfs | 5,0M | 0 | /run/lock | Lock-Dateien |
| tmpfs | 3,0G | 0 | /sys/fs/cgroup | Ressourcen Controller |
| /dev/sda1 | 236M | 32M | /boot | Boot Images etc. |
| /dev/mapper/mail-vg-home | 9,8G | 3,6G | /home |  Persönliche Daten |
| /dev/mapper/mail-vg-tmp | 2,0G | 3,7M | /tmp | Temporäre Dateien |
| /dev/mapper/mail-vg-var | 9,8G | 1,5G | /var | Verschiedenste Daten |

| Dateisystem | Größe | 07-2019 | Eingehängt auf | Daten |
|-----------------------------|-------|---------|------------------|--|
| /dev/mapper/mail-vg-imap | 30G | 3,3G | /var/spool/imap |  E-Mail, Kalender und Kontakte |
| /dev/mapper/mail-vg-seafile | 148G | 98G | /var/www/seafile |  Daten des Cloud Speichers |
| /dev/mapper/mail-vg-mysql | 2,0G | 3,0M | /var/mysql | z..Zt. nicht genutzt |
| /dev/mapper/mail-vg-log | 2,0G | 265M | /var/log | Log Daten |
| /var/tmp | - | | bind (/tmp) | Temporäre Dateien |

Tools

- **Duplicity**: das eigentliche Backup-Programm, erstellt verschlüsselte TAR-Archive und lädt sie auf eine Zielplattform hoch
- **Duply**: Kommandozeilen Frontend für Duplicity
- **BackupNinja**: Kommandozeilen Frontend für Duplicity mit „grafischem“ Wizard
- **Deja-Dup**: Gnome Frontend für Duplicity

Links:

- <https://www.datamate.org/duplicity-duply-datensicherung-auf-die-verlass-ist/>
- <https://vitux.com/how-to-backup-files-with-duplicity-on-ubuntu/>
- <https://www.techrepublic.com/article/how-to-automate-database-backup-with-backupninja/>

Serverkonfiguration

Spamassassin

Spamassassin wird vom User amavis ausgeführt, um also einen manuellen Lernlauf zu starten, muss man als amavis angemeldet sein. Im Home-Verzeichnis dieses Users /var/lib/amavis/ liegt die Datei learn.sh, die alle E-Mail Ordner nach Spam durchsucht und aus den enthaltenen E-Mails lernt.

Getmail / Fetchmail

Getmail

Die Konfiguration der Postfächer, aus denen E-Mails abgeholt werden sollen, erfolgt in /var/spool/getmail/. Hier das Beispiel für die Abholung von lmoppert@gmx.de mit anschließendem lokalen Versand per sendmail:

lmoppert@gmx.de

```
#####
# GetMail Configuration

[retriever]
type = SimpleIMAPSSLRetriever
```

```
server = imap.gmx.net
username = lmoppert@gmx.de
password = *****

[destination]
type = MDA_external
path = /usr/sbin/sendmail
arguments = ('-f', 'lmoppert@gmx.de', '-oi', 'lutz@moppert.de')

[options]
verbose = 0
message_log = /var/spool/getmail/log/lmoppert@gmx.de.log
message_log_verbose = true
read_all = true
delete = true
delivered_to = false
received = false

#####
# vim:syn=getmailrc
```

Fetchmail



Fetchmail wurde in neueren Kolab Installationen durch getmail ersetzt, die folgenden Zeilen dienen nur als Archiv:

Um E-Mails von anderen Mail Accounts holen zu können, sind folgende Schritte nötig:

1. Das Zertifikat des Dienstes in eine Datei speichern:

```
echo "quit" | openssl s_client -connect pop.gmx.net:995 -showcerts |
sed -ne '/BEGIN/,/END/p' > /etc/ssl/certs/gmx.net.pem
```

2. Den entsprechenden Fingerprint herausfinden:

```
openssl x509 -in /etc/ssl/certs/gmx.net.pem -md5 -fingerprint
```

3. Nun noch die Zeilen in der fetchmailrc anlegen:

```
poll pop.gmx.de with proto POP3
  user 'MyID@gmx.de' there with password 'MyPASS'
  is 'Me@LOCAL.DOM' here options ssl sslproto tls1
  sslcertpath /etc/ssl/certs
  sslfingerprint 'AA:BB:77:CC:00:73:4E:EE:FF:EB:B8:C0:90:7D:46:56'
  sslcertck
```

4. Für Google muss der Zugriff von einem Server noch explizit freigeschaltet werden. Hierfür vom Server die URL <https://accounts.google.com/b/0/DisplayUnlockCaptcha> aufrufen und anmelden.

IMAP

Damit die Handys immer E-Mails senden dürfen, sollte das verschicken für angemeldete Benutzer immer erlaubt sein (die Option `permit_sasl_authenticated` ergänzen).

[/etc/postfix/main.cf](#)

```
main.cf:smtpd_recipient_restrictions = permit_mynetworks,  
permit_sasl_authenticated, reject_unauth_pipelining, reject_rbl_client  
zen.spamhaus.org, reject_non_fqdn_recipient,  
reject_invalid_helo_hostname, reject_unknown_recipient_domain,  
reject_unauth_destination, check_policy_service  
unix:private/recipient_policy_incoming, permit
```

From:
<https://wiki.moppert.de/> - Familien Wiki

Permanent link:
<https://wiki.moppert.de/doku.php?id=tech:kolab&rev=1578327118>

Last update: **2020/01/06 16:11**

