Grommunio

Install on debian

Préliminaires

1. Mettre à jour le système:

```
apt update && apt dist-upgrade
```

2. Configurer les locales:

```
dpkg-reconfigure locales
```

(choisir FR-BE.UTF8)

3. Documenter le hostname dans /etc/hosts et /etc/hostname
4. Faire un ping de l'adresse ::1

Installation du repository

1. Installer les paquets nécessaires:

```
apt install gnupg2 curl
```

2. Télécharger la clé et l'insérer:

```
wget -O- https://download.grommunio.com/RPM-GPG-KEY-grommunio | gpg --
dearmor -o /usr/share/keyrings/grommunio.gpg
```

3. Créer le repository:

```
#For Ubuntu
#echo deb [signed-by=/usr/share/keyrings/grommunio.gpg]
https://download.grommunio.com/community/Ubuntu_22.04 >
/etc/apt/sources.list.d/grommunio.list
#For Debian Community
#echo deb [signed-by=/usr/share/keyrings/grommunio.gpg]
https://download.grommunio.com/community/Debian_11 Debian_11 main >
/etc/apt/sources.list.d/grommunio.list
#For Debian paid support
echo deb [signed-by=/usr/share/keyrings/grommunio.gpg]
https://a:b@download.grommunio.com/supported/Debian_11 Debian_11 main >
/etc/apt/sources.list.d/grommunio.list
```

4. Mettre à jour les dépôts:

```
apt update
```

Installation d'un certificat letsencrypt

1. installer les paquets:

```
apt install python3-certbot-nginx certbot nginx
```

2. demander un certificat:

```
certbot -d **Domaine** --nginx -m noc@makeitsimple.be --agree-tos
```

3. Faire une commande CRON pour vérifier la validité du certificat:

```
crontab -e
```

Exemple:

```
# m h  dom mon dow   command
10 10 10,20,30 * * certbot renew
```

Installation de Grommunio

1. Installer les paquets:

```
apt install mariadb-server mariadb-client gromox grommunio-common
```

2. Créer les groupes nécessaires:

```
addgroup gromox
addgroup grommunio
```

3. Créer un fichier de configuration nginx:

```
nano /etc/grommunio-common/nginx/ssl_certificate.conf
```

et indiquez les certificats letsencrypt fraichement créés:

ssl_certificate.conf

```
ssl_certificate /etc/letsencrypt/live/domaine/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/domaine/privkey.pem;
```

4. Créer la base de donnée Grommunio:

```
mysql -u root -p
```

```
CREATE DATABASE `grommunio`;
GRANT ALL ON `grommunio`.* TO 'grommunio'@'localhost' IDENTIFIED BY
'password';
flush privileges;
```

```
quit;
```

5. Editer le fichier mysql_adaptor de Grommunio:

```
nano /etc/gromox/mysql_adaptor.cfg
```

mysql_adaptor.cfg

```
mysql_username=grommunio
mysql_password=password
mysql_dbname=grommunio
schema_upgrade=host:DOMAINE
```

6. Créer les tables:

```
gromox-dbop -C
```

7. Editer le fichier de config gromox:

```
nano /etc/gromox/http.cfg
```

http.cfg

```
listen_port=10080
listen_ssl_port=10443
http_support_ssl=yes
http_certificate_path=/etc/letsencrypt/live/domaine/fullchain.pem
http_private_key_path=/etc/letsencrypt/live/domaine/privkey.pem
```

8. Autoriser l'utilisateur gromox à accéder aux certificats:

```
usermod -G ssl-cert -a gromox
chgrp -R ssl-cert /etc/letsencrypt/live/
chgrp -R ssl-cert /etc/letsencrypt/archive/
chmod 770 -R /etc/letsencrypt/live/
chmod 770 -R /etc/letsencrypt/archive/
```

9. Activer les services:

```
systemctl enable --now gromox-event gromox-timer gromox-http
```

10. Si vous faites

```
curl -kv https://localhost:10443/
```

, un code 404 doit apparaître dans la réponse

11. Petit workaround pour les applications grommunio qui souhaitent ouvrir un socket dans /run/php-fpm

```
echo "d       /run/php-fpm 0755 www-data gromox - -" >
/etc/tmpfiles.d/run-php-fpm.conf
systemd-tmpfiles --create
```

12. Installer le paquet grommunio-web:

```
apt install grommunio-web
```

13. Supprimer le fichier de base de nginx et redémarrer le service

```
rm /etc/nginx/sites-enabled/default
service nginx restart
```

14. Vérifier que cette page est accessible:

```
curl -kv https://localhost:10443/web/robots.txt
```

Configurer les accès IMAP / POP3

1. Activer la synchronisation:

```
systemctl enable --now gromox-midb gromox-zcore
```

2. Editer le fichier de configuration imap:

```
nano /etc/gromox/imap.cfg
```

avec une config similaire:

imap.cfg

```
listen_ssl_port=993
imap_support_starttls=true
imap_certificate_path=/etc/letsencrypt/live/domaine/fullchain.pem
imap_private_key_path=/etc/letsencrypt/live/domaine/privkey.pem
imap_force_starttls=true
```

3. Editer le fichier de configuration pop3:

```
nano /etc/gromox/pop3.cfg
```

avec une config similaire:

pop3.cfg

```
listen_ssl_port=995
pop3_support_stls=true
pop3_certificate_path=/etc/letsencrypt/live/domaine/fullchain.pem
pop3_private_key_path=/etc/letsencrypt/live/domaine/privkey.pem
```

```
pop3_force_stls=true
```

4. Lancer les services:

```
systemctl enable --now gromox-imap gromox-pop3
```

5. Quelques tests:

```
curl -kv imaps://localhost/
curl -kv pop3s://localhost/
```

Install Grommunio Admin

1. Installer les paquets:

```
apt install grommunio-admin-api grommunio-admin-web
```

2. Editer le fichier de config:

```
nano /etc/grommunio-admin-api/conf.d/database.yaml
```

et y spécifier les paramètres mysql

database.yaml

```
DB:
  host: 'localhost'
  user: 'grommunio'
  pass: 'password'
  database: 'grommunio'
```

3. Générer un mot de passe admin:

```
grommunio-admin passwd
```

4. Faire un lien symbolique avec la configuration ssl:

```
ln -s /etc/grommunio-common/nginx/ssl_certificate.conf /etc/grommunio-admin-common/nginx-ssl.conf
```

5. Activer le service:

```
systemctl enable --now grommunio-admin-api
```

6. Modifier les droits pour autoriser l'upload du fichier licence:

```
chown :gromox /etc/grommunio-admin-common/license/
chmod 775 /etc/grommunio-admin-common/license/
```

Configuration autodiscover

1. Créer le fichier autodiscover:

```
nano /etc/gromox/autodiscover.ini
```

Avec ce contenu

autodiscover.ini

```
[database]
host = localhost
username = 'grommunio'
password = 'password'
dbname = 'grommunio'

[exchange]
hostname = mail2.creapix.eu
;mapihttp = 1

[default]
timezone = 'Europe/Brussels'

[system]

[http-proxy]
/var/lib/gromox/user = mail2.creapix.eu
/var/lib/gromox/domain = mail2.creapix.eu
```

Redis

1. Installer le paquet:

```
apt install redis
```

2. Créer un répertoire et lui donner les droits nécessaires:

```
mkdir -p /var/lib/redis/default/
chown redis:redis * -R /var/lib/redis/
```

3. Créer un script de démarrage:

```
systemctl edit redis@grommunio.service --full --force
```

et coller ceci dedans:

```
[Unit]
Description=Redis instance: %i
```

```
After=network.target
PartOf=redis.target

[Service]
Type=notify
User=redis
Group=redis
PrivateTmp=true
PIDFile=/run/redis/%i.pid
ExecStart=/usr/bin/redis-server /etc/redis/%i.conf
LimitNOFILE=10240
Restart=on-failure

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

4. Redémarrer/activer/désactiver les services:

```
systemctl daemon-reload
systemctl disable --now redis
systemctl enable --now redis@grommunio.service
systemctl status redis@grommunio.service
```

Grommunio Sync

Grommunio-sync permet de synchroniser avec les smartphones via la technologie EAS (activesync).

1. Installer le paquet:

```
apt install grommunio-sync
```

2. Workaround suite à une erreur de paquet, copier le fichier dans la bonne location

```
ln -s  /etc/php7/fpm/php-fpm.d/pool-grommunio-sync.conf
/etc/php/7.4/fpm/pool.d
```

3. Donner les droits au folder de logs:

```
chmod 770 /var/log/grommunio-sync
```

4. Redémarrer le service:

```
service php7.4-fpm restart
```

Postfix

Grommunio-Delivery ne peut envoyer les mails qu'en interne. Pour relayer les mails vers l'extérieur, nous avons besoin de Postfix.

1. Installer les paquets:

```
apt install postfix postfix-mysql
```

2. Modifier le port de Grommunio-smtp en port 24:

```
echo "listen_port = 24" > /etc/gromox/smtp.cfg
```

3. Editer un fichier postfix permettant d'atteindre les alias des users grommunio:

```
nano /etc/postfix/g-alias.cf
```

contenu:

g-alias.cf

```
user = grommunio
password = password
hosts = 127.0.0.1
dbname = grommunio
query = SELECT mainname FROM aliases WHERE aliasname='%s'
```

4. Editer un fichier postfix permettant d'atteindre les domaines connus:

```
nano /etc/postfix/g-virt.cf
```

Contenu:

g-virt.cf

```
user = grommunio
password = password
hosts = 127.0.0.1
dbname = grommunio
query = SELECT 1 FROM domains WHERE domain_status=0 AND
domainname='%s'
```

5. Quelques commandes pour configurer postfix:

```
postconf -e virtual_alias_maps=mysql:/etc/postfix/g-alias.cf
postconf -e virtual_mailbox_domains=mysql:/etc/postfix/g-virt.cf
postconf -e virtual_transport="smtp:[localhost]:24"
postconf -e mynetworks="127.0.0.0/8 [::1]/128"
postconf -e smtpd_banner='$myhostname ESMTP'
postconf -e inet_interfaces="all"
postconf -e
smtpd_helo_restrictions="permit_mynetworks,permit_sasl_authenticated,re
ject_invalid_hostname,reject_non_fqdn_hostname"
postconf -e
```

```
smtpd_sender_restrictions="reject_non_fqdn_sender,permit_sasl_authenticated,permit_mynetworks,reject_unknown_sender_domain,reject_unknown_reverse_client_hostname,reject_unknown_client_hostname"
postconf -e smtpd_recipient_restrictions="permit_sasl_authenticated,permit_mynetworks,reject_unknown_recipient_domain,reject_non_fqdn_hostname,reject_non_fqdn_sender,reject_non_fqdn_recipient,reject_unauth_destination,reject_unauth_pipelining"
postconf -e message_size_limit=20480000

postconf -e smtpd_use_tls=yes
postconf -e smtpd_tls_key_file=/etc/letsencrypt/live/domaine/privkey.pem
postconf -e smtpd_tls_cert_file=/etc/letsencrypt/live/domaine/fullchain.pem
postconf -e smtpd_tls_session_cache_database=btree:${data_directory}/smtpd_scache

postconf -e smtp_use_tls=yes
postconf -e smtp_tls_key_file=/etc/letsencrypt/live/domaine/privkey.pem
postconf -e smtp_tls_cert_file=/etc/letsencrypt/live/domaine/fullchain.pem
postconf -e smtp_tls_session_cache_database=btree:${data_directory}/smtp_scache
postconf -e smtp_tls_security_level=may
postconf -e smtp_tls_note_starttls_offer=yes
postconf -e smtp_tls_enforce_peername=no
postconf -e myhostname=mail1.creapix.eu
```

6. Redémarrer/Activer les services:

```
systemctl enable --now gromox-delivery gromox-delivery-queue postfix
systemctl restart gromox-delivery-queue postfix
```

Installer rspamd

1. Installer le paquet nécessaire:

```
apt install lsb-release
```

2. Ajouter la clé gpg du dépot rspamd:

```
wget -O- https://rspamd.com/apt-stable/gpg.key | gpg --dearmor | tee /usr/share/keyrings/rspamd.gpg
```

3. Ajouter le dépot:

```
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/rspamd.gpg] http://rspamd.com/apt-stable/ $(lsb_release -cs) main" | tee /etc/apt/sources.list.d/rspamd.list
```

```
echo "deb-src [arch=amd64 signed-by=/usr/share/keyrings/rspamd.gpg]
http://rspamd.com/apt-stable/ $(lsb_release -cs) main"  | tee -a
/etc/apt/sources.list.d/rspamd.list
```

4. Mettre à jour & installer rspamd:

```
apt update
apt install rspamd --no-install-recommends
```

5. Modification du port d'écoute:

```
nano /etc/rspamd/local.d/worker-normal.inc
```

Contenu

```
bind_socket = "127.0.0.1:11333";
```

6. Modification du port milter

```
nano /etc/rspamd/local.d/worker-proxy.inc
```

Contenu:

worker-proxy.inc

```
bind_socket = "127.0.0.1:11332";
milter = yes;
timeout = 120s;
upstream "local" {
  default = yes;
  self_scan = yes;
}
```

7. Nous allons à présent créer un code pour l'interface de gestion rspamd:

```
rspamadm pw --encrypt -p Password
```

Garder ce code pour l'étape suivante

8. Editer le fichier contenant le mot de passe:

```
nano /etc/rspamd/local.d/worker-controller.inc
```

Contenu:

```
password = "lecode-encrypté"
```

9. Editer le fichier classifier-bayes:

```
nano /etc/rspamd/local.d/classifier-bayes.conf
```

Contenu:

classifier-bayes.conf

```
servers = "127.0.0.1";
backend = "redis";
autolearn = true;
```

10. Editer le fichier milter_headers:

```
nano /etc/rspamd/local.d/milter_headers.conf
```

Contenu:

milter_headers.conf

```
extended_spam_headers = true;
skip_local = false;
skip_authenticated = false;
use = ["spam-header"];
routines {
   spam-header {
#     header = "X-Spam-Flag";
#     value = "Yes";
     header = "X-Spam";
     value = "Yes";
   }
}
```

11. Editer le fichier redis.conf:

```
nano /etc/rspamd/local.d/redis.conf
```

Contenu:

redis.conf

```
servers = "127.0.0.1";
```

12. Redémarrer rspamd:

```
systemctl restart rspamd
```

13. Ajouter un reverse proxy dans nginx:

```
nano /usr/share/grommunio-admin-common/nginx.d/antispam.conf
```

Contenu:

antispam.conf

```
location ^~ /rspamd/ {
    proxy_pass http://127.0.0.1:11334/;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}
```

14. Lier rspamd à postfix:

```
postconf smtpd_milters=inet:127.0.0.1:11332
postconf non_smtpd_milters=inet:127.0.0.1:11332
postconf milter_protocol=6
postconf -e "milter_default_action = accept"
service postfix restart
```

15. Pour tester la config:

```
Pour tester la config
rspamadm configtest
rspamadm configdump
```

1. Création de deux règles pour whitelister les domaines ou les ips. Editer le fichier multimap.conf

```
nano /etc/rspamd/local.d/multimap.conf
```

Et coller le contenu suivant

multimap.conf

```
WHITELIST_SENDER_DOMAIN {
      type = "from";
      filter = "email:domain";
      map = "/var/lib/rspamd/whitelist.sender.domain.map";
      score = -10.0
}

WHITELIST_SENDER_IP {
      type = "ip";
      prefilter = "true";
      map = "/var/lib/rspamd/whitelist.sender.ip.map";
      #action = "accept";
      score = -10.0
}

BLACKLIST_SENDER_DOMAIN {
      type = "from";
      filter = "email:domain";
      map = "/var/lib/rspamd/blacklist.sender.domain.map";
      score = 10.0
```

```
}

BLACKLIST_SENDER_IP {
      type = "ip";
      prefilter = "true";
      map = "/var/lib/rspamd/blacklist.ip.map";
      action = "reject";
}
```

2. Créer un répertoire pour les signatures dkim:

```
mkdir /var/lib/rspamd/dkim
chown -R _rspamd:_rspamd /var/lib/rspamd/dkim
```

Signer DKIM

1. Créer un répertoire pour stocker les clés:

```
mkdir /var/lib/rspamd/dkim/
```

2. Créer un fichier dkim_signing

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

Avec ceci:

dkim_signing.conf

```
# If false, messages with empty envelope from are not signed
# If false, messages with empty envelope from are not signed
allow_envfrom_empty = true;

# If true, envelope/header domain mismatch is ignored
allow_hdrfrom_mismatch = false;

# If true, multiple from headers are allowed (but only first is
used)
allow_hdrfrom_multiple = false;

# If true, username does not need to contain matching domain
allow_username_mismatch = false;

# Default path to key, can include '$domain' and '$selector'
variables
path = "/var/lib/rspamd/dkim/$domain.$selector.key";

# Default selector to use
selector = "dkim";
```

```
# If false, messages from authenticated users are not selected for
signing
sign_authenticated = true;

# If false, messages from local networks are not selected for
signing
sign_local = true;

# Map file of IP addresses/subnets to consider for signing
# sign_networks = "/some/file"; # or url

# Symbol to add when message is signed
symbol = "DKIM_SIGNED";

# Whether to fallback to global config
try_fallback = true;

# Domain to use for DKIM signing: can be "header" (MIME From),
"envelope" (SMTP From) or "auth" (SMTP username)
use_domain = "header";

# Domain to use for DKIM signing when sender is in sign_networks
("header"/"envelope"/"auth")
#use_domain_sign_networks = "header";

# Domain to use for DKIM signing when sender is a local IP
("header"/"envelope"/"auth")
#use_domain_sign_local = "header";

# Whether to normalise domains to eSLD
use_esld = true;

# Whether to get keys from Redis
use_redis = true;

# Hash for DKIM keys in Redis
key_prefix = "DKIM_KEYS";

# If `true` get pubkey from DNS record and check if it matches
private key
check_pubkey = false;
# Set to `false` if you want to skip signing if public and private
keys mismatch
allow_pubkey_mismatch = true;
```

Signer ARC

1. Créer un fichier arc.conf

```
nano /etc/rspamd/local.d/arc.conf
```

Avec ceci:

arc.conf

```
# If false, messages with empty envelope from are not signed
allow_envfrom_empty = true;
# If true, envelope/header domain mismatch is ignored
allow_hdrfrom_mismatch = false;
# If true, multiple from headers are allowed (but only first is
used)
allow_hdrfrom_multiple = false;
# If true, username does not need to contain matching domain
allow_username_mismatch = false;
# Default path to key, can include '$domain' and '$selector'
variables
path = "${DBDIR}/dkim/$domain.$selector.key";
# Default selector to use
selector = "dkim";
# If false, messages from authenticated users are not selected for
signing
sign_authenticated = true;
# If false, messages from local networks are not selected for
signing
sign_local = true;
# Symbol to add when message is signed
symbol_signed = "ARC_SIGNED";
# Whether to fallback to global config
try_fallback = true;
# Domain to use for ARC signing: can be "header" or "envelope"
use_domain = "header";
# Whether to normalise domains to eSLD
use_esld = true;
# Whether to get keys from Redis
use_redis = false;
# Hash for ARC keys in Redis
key_prefix = "ARC_KEYS";
# map of domains -> names of selectors (since rspamd 1.5.3)
#selector_map = "/etc/rspamd/arc_selectors.map";
# map of domains -> paths to keys (since rspamd 1.5.3)
#path_map = "/etc/rspamd/arc_paths.map";
# map of trusted domains. Symbol ARC_ALLOW_TRUSTED is added to
messages
# with valid ARC chains from these domains. A failed DMARC result
is removed/ignored.
# whitelisted_signers_map = ["example.org", "example.com"]

# From version 1.8.4, Rspamd uses a different set of sign_headers
for ARC:
sign_headers = "(o)from:(o)sender:(o)reply-
```

```
to:(o)subject:(o)date:(o)message-id:(o)to:(o)cc:(o)mime-
version:(o)content-type:(o)content-transfer-encoding:resent-
to:resent-cc:resent-from:resent-sender:resent-message-id:(o)in-
reply-to:(o)r>
```

Grommunio-dav (WIP)

```
apt install grommunio-dav
ln -s /etc/php7/fpm/php-fpm.d/pool-grommunio-dav.conf
/etc/php/7.4/fpm/pool.d
chown root:grodav /var/lib/grommunio-dav/
```

Tips & tricks

Activer le port submission

```
postconf -M submission/inet="submission inet n - n - - smtpd"
postconf -P submission/inet/syslog_name="postfix/submission"
postconf -P submission/inet/smtpd_tls_security_level=encrypt
postconf -P submission/inet/smtpd_sasl_auth_enable=yes
postconf -P
submission/inet/smtpd_relay_restrictions=permit_sasl_authenticated,reject
postconf -P submission/inet/milter_macro_daemon_name=ORIGINATING
systemctl restart postfix
```

Activer ssl pour admin

```
nano /usr/share/grommunio-admin-common/nginx-ssl.conf
```

1. commenter la ligne qui fait de nouveau appel à nginx-ssl.conf
2. ajouter include /etc/grommunio-common/nginx/ssl_*.conf;

Ensuite il faut décommenter la dernière ligne dans /etc/nginx/conf.d/grommunio-admin.conf

Autoriser le relay pour les users authentifiés

nano /etc/postfix/master.cf -o
smtpd_recipient_restrictions=reject_non_fqdn_recipient,reject_unknown_recipient_domain,permit_sasl_authenticated,reject

Remplacer reject par reject_unauth_destination

Backup / Transfert

```
systemctl restart gromox-http
systemctl restart gromox-midb

mysqldump --single-transaction --routines --triggers --events --add-drop-database grommunio > grommunio-mysql-backup.sql

rsync -avzh --progress root@46.105.80.174:/var/lib/gromox/domain/ /var/lib/gromox/domain/
rsync -avzh --progress root@46.105.80.174:/var/lib/gromox/user/ /var/lib/gromox/user/


mysql - u root -p
```

```
SET autocommit=0 ; source grommunio-mysql-backup.sql ; COMMIT ;
```

Activer le debugging

```
echo http_debug=1 >> /etc/gromox/http.cfg
systemctl restart gromox-http
journalctl -fu gromox-http
```

Lister les utilisateurs

```
echo "select username,maildir from users where maildir<>'';" | mysql -N grommunio
```

Nettoyer les comptes users

```
for i in /var/lib/gromox/user/*/*; do /usr/libexec/gromox/cleaner -v -d "$i"; done
```

Workaround pour les confirmations de lecture envoyées à invalid@invalid

1. ```
   nano /etc/postfix/main.cf
   ```

   Et rajouter invalid dans mydestination

   ```
   mydestination = $myhostname, localhost.$mydomain, localhost, invalid
   ```

2. ```
   nano /etc/aliases
   ```

Et rajouter invalid: postmaster

3. ```
   newaliases
   service postfix restart
   ```

Workaround pour le "upstream sent too big header while reading response header from upstream"

```
nano /usr/share/grommunio-common/nginx/locations.d/grommunio-web.conf
```

grommunio-web.conf

```
fastcgi_buffers 16 16k;
fastcgi_buffer_size 32k;
```
```