

[Logiciel](#)

Msmtp : envoyer un courriel par smtp en ligne de commande

Le fonctionnement du courrier électronique est basé sur l'utilisation d'une boîte à lettres électronique.

Lors de l'envoi d'un email, le message est acheminé de serveur en serveur jusqu'au serveur de messagerie du destinataire.

Le message est envoyé au serveur de courrier électronique chargé du transport (nommé MTA pour Mail Transport Agent), jusqu'au MTA du destinataire.

[Sur le web, les MTA communiquent entre-eux grâce au protocole SMTP et sont appelés serveurs SMTP \(mais aussi serveur de courrier sortant\).](#)



Le serveur MTA du destinataire délivre alors le courrier au serveur de courrier électronique entrant (nommé MDA pour Mail Delivery Agent), qui stocke alors le courrier en attendant que l'utilisateur vienne le relever.

La relève du courrier se fait grâce à un logiciel appelé MUA (Mail User Agent). On parle de client de messagerie (par exemple Mozilla Thunderbird, Microsoft Outlook, Eudora Mail, Incredimail, ...).

Pré-requis

Installation

1. Installez msmtp,msmtp-mta,bsd-mailx

```
...@...:~$ sudo apt install {msmtp,msmtp-mta,bsd-mailx}
```

2. Vérifiez que sendmail est présent sur votre système :

```
...@...:~$ sudo whereis sendmail
sendmail: /usr/sbin/sendmail /usr/lib/sendmail /usr/share/sendmail
/usr/share/man/man8/sendmail.8.gz
```

3. Vérifiez que sendmail est lié à msmtp :

```
...@...:~$ sudo ls -la /usr/sbin/sendmail
lrwxrwxrwx 1 root root 12 oct. 11 2019 /usr/sbin/sendmail ->
../bin/msmtp
```

→ Les 2 logiciels sont bien associés.

Configuration

Le paramétrage des options se fait dans le fichier **/etc/msmtpc** (pour tous les utilisateurs ou **~/.msmtpc** (pour un utilisateur)

Commandes du fichier msmtpc

Exemple de fichier msmtpc :

[/etc/msmtpc](#)

```
# Valeurs par défaut pour tous les comptes.
defaults
auth            on
tls             on
tls_starttls   on
tls_trust_file /etc/ssl/certs/ca-certificates.crt
logfile        /var/log/msmtp

# Exemple pour un compte Gmail
account        gmail
auth           plain
host           smtp.gmail.com
port           587
from           username@gmail.com
user           username
password       XXXXXXXXXXXX

# Exemple pour un compte GMX
account        gmx
host           mail.gmx.com
port           587
from           username@gmx.com
user           username@gmx.com
password       XXXXXXXXXXXX
tls_nocertcheck

# Exemple pour un compte OVH
account        ovh
host           XXXXXX.ovh.net
port           465
from           username@example.com
user           username@example.com
password       XXXXXXXXXXXX

# Exemple pour un compte Infomaniak
```

```
account      infomaniak
host         mail.infomaniak.com
port         587
from         username@example.com
user         username@example.com
password     XXXXXXXXXXXX

# Exemple pour un serveur de test MailHog
account      mailhog
auth         off
tls          off
tls_starttls off
host         localhost
port         1024
from         username@example.com

# Définir le compte par défaut
account default : gmail
```

Modèle exemple de fichier :

[/etc/msmtprc](#)

```
# nom du compte (le fichier de configuration accepte plusieurs
# configurations de serveurs SMTP)
account default
# serveur SMTP utilisé
host smtp.monfai.fr
# Adresse de l'expéditeur
from prenom.nom@monfai.fr
# si pas d'authentification
auth off
# si authentification
#auth on
# expéditeur du mail
#user prenom.nom@fai.fr
# mot de passe en clair dans le fichier de configuration
password *****
```

Exemple pour gmail

Exemple de fichier de configuration utilisateur ~/.msmtprc

Ce fichier se concentre sur le protocole TLS et l'authentification.

Les fonctionnalités non utilisées ici incluent la journalisation, les délais d'expiration, les proxy SOCKS, les paramètres TLS, les paramètres de notification d'état de livraison (DSN), etc.

- **Adresse du serveur SMTP Gmail** : smtp.gmail.com.

- **Nom SMTP Gmail** : Votre nom complet.
- **Identifiant SMTP Gmail** : votre adresse Gmail complète (par exemple vous@gmail.com)
- **Mot de passe SMTP Gmail** : le mot de passe que vous utilisez pour vous connecter à Gmail.
- **Port SMTP Gmail (TLS)** : 587.
- **Port SMTP Gmail (SSL)** : 465.

~/msmtp.rc

```
# Valeurs par défaut pour tous les comptes suivants.
defaults

# Utiliser le port 587 au lieu du port SMTP 25.
port 587

# Toujours utiliser TLS.
tls on

# Définir une liste d'autorités de certification de confiance pour
# TLS.
tls_certcheck on
tls_trust_file /etc/ssl/certs/ca-certificates.crt

# Service Gmail
#####
account gmail

# Serveur SMTP
host smtp.gmail.com

# Adresse de l'expéditeur du mail
from *****@gmail.com

# Authentification. Le mot de passe est donné en utilisant l'une
# des cinq méthodes, voir ci-dessous.
auth on
user *****@gmail.com

# Mot de passe, enregistré directement dans ce fichier.
# Assurez-vous que ce fichier ne peut être lu que par vous-même.
password *****

# Fichier journal
logfile /var/log/msmtp.log

# Compte par défaut
account default : gmail (10)
```

defaults

Valeurs par défaut pour tous les comptes qui suivent.

```
; port 587
```

port d'envoi pour les mails : utiliser le port 587 au lieu du port SMTP 25.

```
; tls on
```

```
tls_starttls on
```

Toujours utiliser TLS (Connexion chiffrée)

La connexion entre votre ordinateur et le serveur distant est chiffrée, ce qui signifie que si une personne arrive à intercepter les données, elles seront illisibles, car elles nécessitent une sorte de "mot de passe" connu de votre ordinateur et du serveur pour être lues.

```
; tls_certcheck on
```

```
tls_trust_file /etc/ssl/certs/ca-certificates.crt
```

permettent d'indiquer ou sont situés les certificats par défaut de votre OS, ici c'est le chemin par défaut sous Debian, donc compatible Ubuntu.

Définit une liste d'autorités de certification approuvées pour TLS.

Par défaut, on utilise les paramètres système, mais vous pouvez sélectionner votre propre fichier.

```
#tls_trust_file /etc/ssl/certs/ca-certificates.crt
```

Si vous sélectionnez votre propre fichier, vous devez également utiliser la commande **tls_crl_file** pour vérifier les certificats révoqués, mais malheureusement, obtenir des listes de révocation et les tenir à jour n'est pas simple.

```
#tls_crl_file ~/.tls-crls
```

```
account gmail
```

Compte Gmail que vous utiliserez pour envoyer vos mails. Le fichier de configuration accepte plusieurs configurations de serveurs SMTP.

```
; host smtp.gmail.com
```

Nom d'hôte du serveur SMTP

; Au lieu de `tls_trust_file/tls_crl_file`,

vous pouvez utiliser `tls_fingerprint` pour épingler un seul certificat.

Vous devez mettre à jour l'empreinte digitale lorsque le certificat du serveur change, mais un attaquant ne peut pas vous inciter à accepter un certificat frauduleux.

Obtenez l'empreinte digitale avec

```
# $ msmtp --serverinfo --tls --tls-certcheck=off --  
host=smtplib.freemial.example  
#tls_fingerprint  
00:11:22:33:44:55:66:77:88:99:AA:BB:CC:DD:EE:FF:00:11:22:33
```

; from prenom.nom@gmail.com

origine du message, adresse de l'expéditeur

Authentification

```
auth on  
user prenom.nom@gmail.com
```

ces 2 paramètres activent l'authentification de l'expéditeur du mail. Absolument nécessaire pour sécuriser l'envoi de mails.

Le mot de passe est fourni en utilisant l'une de ces cinq méthodes :

1. Ajoutez le mot de passe au trousseau de clés du système et laissez msmtp l'obtenir automatiquement. Pour définir le mot de passe du trousseau de clés à l'aide de libsecret de Gnome :

```
# $ secret-tool store --label=msmtp \  
# host smtplib.freemial.example \  
# service smtp \  
# user joe.smith
```

2. Stockez le mot de passe dans un fichier chiffré et indiquez à msmtp la commande à utiliser pour le déchiffrer. Ceci se fait généralement avec GnuPG, comme dans cet exemple. Habituellement, `gpg-agent` demande une fois le mot de passe de décryptage.

```
#passwordeval gpg2 --no-tty -q -d ~/.msmtp-password.gpg
```

3. Stocker le mot de passe directement dans ce fichier. En général, ce n'est pas une bonne idée de stocker les mots de passe dans des fichiers texte. Si vous le faites quand même, assurez-vous au moins que ce fichier ne peut être lu que par vous-même.

```
password *****
```

4. Stocker le mot de passe dans ~/.netrc. Cette méthode n'est sans doute plus d'actualité.
5. Ne pas spécifier de mot de passe. Msmtplib vous le demandera alors. Cela signifie que vous devez pouvoir taper dans un terminal lorsque msmtplib s'exécute.

Pour ce tutoriel, j'ai choisi la plus facile mais la moins sécurisée, c'est-à-dire la saisie directe du mot de passe en clair dans le fichier de configuration. Je vous conseillerai plutôt une autre méthode, la 1 où la 2 qui cache le mot de passe.

```
password *****
```

méthode 3 : mot de passe en clair dans le fichier de configuration

```
; logfile /var/log/msmtplib.log
```

Fichier journal. Cette ligne permet de créer un fichier de log spécifique pour l'application msmtplib.

```
; account default
```

```
gmail
```

Définir un compte par défaut

msmtplib permet d'utiliser plusieurs comptes smtp. Vous devez en choisir un par défaut.



Si vous voulez configurer l'envoi de courriels au niveau d'un utilisateur seulement, créez le fichier **~/.msmtplibrc** dans le dossier personnel de l'utilisateur concerné pour y écrire le même contenu.

- **exemple pour smtp.free.fr :**

```
~/.msmtplibrc
```

```
# exemple de compte avec free
account free

host smtp.free.fr

from prenom.nom@monfai.fr
```

```
auth off

user prenom.nom
```

Configurations particulières

1. **gmail** : gmail impose l'utilisation d'une couche TLS :

[~/msmtprc](#)

```
account default
tls on
host smtp.gmail.com
port 587
from nom_user@gmail.com
auth on
user nom_user
password password_user
tls_starttls on
tls_certcheck on
tls_trust_file /etc/ssl/certs/ca-certificates.crt
```

La dernière ligne est spécifique aux systèmes basés sur debian. Elle pointe vers le fichier où sont stockés l'ensemble des certificats racine de confiance du système

2. **gmail non sécurisé**, Il est possible de ne pas vérifier le certificat serveur est désactivant `tls_certcheck`. Ce n'est pas recommandé car sensible à une attaque de type MiM :

[~/msmtprc](#)

```
account default
tls on
host smtp.gmail.com
port 587
from nom_user@gmail.com
auth on
user nom_user
password password_user
tls_starttls off
tls_certcheck off
```

3. **Orange sécurisé**, fonctionne avec TLS :

[~/msmtprc](#)

```
account default
```

```
host smtp.orange.fr
protocol smtp
auth login
tls on
tls_starttls off
tls_certcheck on
tls_trust_file /etc/ssl/certs/ca-certificates.crt
from mon-adresse@orange.fr
maildomain orange.fr
port 465
user mon-adresse
password mon-motdepasse
```

4. **gandi.net** (nom de domaine acheté) :

```
account default
host mail.gandi.net
from <adresse de l'émetteur>
tls on
tls_certcheck off
tls_starttls off
auth on
user <adresse Gandi/nom d'utilisateur utilisé pour se connecter à la
boîte aux lettres électronique Gandi>
password <mot de passe boîte aux lettres électronique>
```

Utilisation

1. **Test de fonctionnement** : Pour valider le bon fonctionnement de votre configuration, essayez d'envoyer un message en ligne de commande :

- en mode interactif :

```
...@...:~$ mail destinataire@domaine.fr
Subject: .....
.... (corps du message)
a moi
. (un point suivi de <key>Entrée</key> pour terminer)
Cc:
```

Suivre les indications en mode interactif. Pour terminer le mail, tapez un point en début de ligne "."

- ou en mode non interactif :

```
...@...:~$ printf "Un message\nde test" | mail -s "sujet de test"
destinataire@domaine.fr
```

2. envoyer le contenu d'un fichier texte :

```
...@...:~$ cat readme.txt | mail johnny@liday.ch
```

3. envoyer un fichier binaire ou compressé en pièce jointe :

```
...@...:~$ uuencode application.zip | mail johnny@liday.ch
```

- le sujet du mail est une ligne en tête de message : **Subject:DeQuoiOnParle** → si on envoie un fichier readme.txt, il n'aura de sujet que s'il commence par Subject:...

Attention: il faut alors un saut de ligne : utilisez printf plutôt que echo. Exemple:



```
...@...:~$ printf "Subject:De Quoi On\nParle\nLeCorpsDuMessage" | mail\njohnny@liday.ch
```

Vérifier que les mails sont bien partis en regardant le contenu de la file d'attente :



message d'erreur : send-mail: impossible d'écrire dans le journal /var/log/msmtp.log : erreur d'ouverture de fichier: Permission non accordée

le fichier est à renommer et à créer avec les droits pertinents : modifier dans .msmtprc la ligne :

```
logfile /var/log/msmtp
```



il est nécessaire de créer le fichier de log , accessible par msmtp :

```
...@...:~ $ sudo groups msmtp\n...@...:/var/log $ sudo touch /var/log/msmtp\n...@...:/var/log $ sudo chown msmtp:msmtp /var/log/msmtp\n...@...:/var/log $ sudo chmod 660 /var/log/msmtp
```

Désinstallation

Voir aussi

- **(fr)** <https://doc.ubuntu-fr.org/msmtp>
- **(fr)** https://doc.ubuntu-fr.org/tutoriel/comment_envoyer_un_mail_par_smtp_en_ligne_de_commande
- **(fr)** <https://quick-tutoriel.com/installer-un-client-smtp-sous-linux/>
- **(en)** <http://msmtp.sourceforge.net/>
- **(en)** <https://marlam.de/msmtp/msmtp.pdf>
- **(fr)** <http://blogettrux.free.fr/index.php?2020/11/23/22/00/09-config-msmtp>

Basé sur « [Comment envoyer un courriel par smtp en ligne de commande](#) » par doc.ubuntu-fr.org.

From:

<https://nfrappe.fr/doc/> - **Documentation du Dr Nicolas Frappé**

Permanent link:

<https://nfrappe.fr/doc/doku.php?id=logiciel:systeme:msmtp:start>



Last update: **2022/11/08 19:28**