

Formation Sympa : les bases

Cette journée de formation vous permettra d'aborder les bases de Sympa : manipulations de base, éléments essentiels d'administration (interface, scénarios d'autorisation, sources de données).

Avant toute chose, assurez-vous que vous accédez à votre messagerie depuis votre poste de travail.

Si ce n'est pas le cas, créez-vous, pour le temps du TP, une adresse gmail. Gmail supportant les plussed aliases, vous pourrez la décliner à l'infini, pour les besoins du TP, sous la forme « partie_locale+autre_chaine@gmail.com ».

Personnalisation du support de TP

Dans toute cette documentation, le nom du poste pris en exemple est **nouveau.domaine.fr**, un petit script permet d'entrer votre nom de machine et de remplacer toutes les occurrences de **nouveau.domaine.fr** dans cette page, vous pourrez à tout moment corriger ou refaire l'opération en tapant simplement votre nom de poste dans le champs ci dessous.

Connectez-vous en SSH sur le serveur.

Utilisateur : stagiaire

Clé privée :

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAw+/gEnhRxET4TB529yVzAvUfRYfCg3t6XF46ZTD/AE8cE1zx
gJK7eSMERMsAtcEnwehdhtmdxJky70uuD1V22+HKEspxcA0X32RI7egkH6Pvdckk
4xte7A3P9ViQMgLD7KTHmWdnNjvPUBsLbLR0JEH49N/OiWSeZ7n0tUgm/YN6goiW
I4AGqJb932Lrv28Fh3EeFEL/9wB7vUHBze3D+k1+9nLFpN02bYE070zY/Nlw/VJC
08d8/awLS5EK5UEKJmXpU0YjsLChjvDkSK/orBjAjha3VGo3ez0SeJ5N86N0+gsA
ExREXD/N/iTWxq2229aQGxb0aGvjP50hsCs2ywIDAQABAoIBAB7C3SnPK+UnBMJm
kgTRI1JWi3d0DhK4Ywh3XrGeVJqG0QCV0fEWmEo3XjeGk3D1hzlhMrXGGoFQCXe6
tJQBtexlcWTqhe6xEbnns69uH6W8Bg9KshbZqDhLHr4FDnZbjt3lLNT/r+uKzkxk
QpIquC3nEZ/YN0PIwTnFrw566mwoXlpSgSXfmWZt3QeVeVVdATQkn6xB4Ij6RUXz
xCwTxgkLXGRAlrNBop6fllgR0u4gYmfx+8w7n209E1nZC4G7vx264Q6XIKPwwiBx
UKZw7c5jE5eKMy5kW1rKpzR613jrE8xeZUYPPc/76AgZ6+oZkvLJiGJyTHnHmx8L
aQmfe4ECgYEA+4PpUF5YhnGSejPa9XZisIX0BKd07llcWSuj9yyLVTi6/1a3MYzA
gBD2GiMSTnxVA0SWfiBTskMVPnFAJ+FTjFlKPvmaCHNH6a1w1rUGKZTemiIbL9IR
13CatqSowyyQ0FtYSjCD4sw/E5yFQDkx6f0Qw35gHXgLibzu3IaSJIsCgYEAx25E
OLK/qnZiHwSRRQM/rBXe1hNhx5oE6mgwSZ0tLBRqeqhRUVQG9c0siS+zs7IsCcc0
WHEohq0IbeJXuF/7GuNAsxmQJmjKGegh5rgGS7oHR0K4NVzJV0b4ZQF3GS1enJKA
aP3V4HCJE0GA1bmPerEg3yYjknU2s5AqZ1XsPsECgYEAqbG5Y+ETxzmVQ0XjUE0c
mE74cX9UcNyKpxsbmHP0Wf5ZpFckaIj3hDBtavsIqe2XCHAx3U0AA/0MI0ITsBSF
4yaHQM/zbgohlDbQT/x4+0sZ0VMTlrmCGIg/ykTUHELgPc0zkPKkuQtm71tmSAu0
0rlMaynDvX42Aqt3WVBuH7sCgYEAty8bzxCxaTx45jxVy5RuWf1E0FLPx4S72yyU
niDdwk2Ge0A+wXtzYThzHhgI8phIRzsJY+udFAfAZF6xwJ05LTts5JYoiH90di95
ZFnIvqpDnwy5s5pk/pwb8XtlEGVSMH0JK+dtG1mDL4LNeo0VvVcSIKcBqbes5UcZ
DA4qkIECgYEAoIqckMo9f0Vd35tPjZRoYUAF0oCpm2B94qHpSvtD3px0LFpF1vVH
-----
```

```
VYC+Uxup4mxRfidQA5w9jpCnbiCFDtXKkeeus6dQsSgaH5o4gevv7AKxw0yjdjFnp
XHkQDSKcLjjBldba1GoS5w6boHMjK4BZEzvUDDGo9xdVdKGoA6PBXDo=
-----END RSA PRIVATE KEY-----
```

Commençons par vérifier le nom du poste sur lequel vous travaillez. **Le nom de votre serveur est form-sympaXX.renater.fr, où XX est le numéro assigné en début de formation (01, 02, etc.) :**

Nom de votre machine :

Votre editeur :

Accès SSH

Le TP utilise des couples clé privée / clé publique pour l'accès SSH au serveur (ça évite de conserver l'authentification par mot de passe sur des serveurs publics). Pour accéder à votre machine de formation, suivez les étapes suivantes :

```
emacs ~/.ssh/id_rsa.stage
```

Collez dans le fichier le contenu de la clé privée que nous vous envoyons par mail.

```
emacs ~/.ssh/id_rsa.stage.pub
```

Collez dans le fichier le contenu suivant :

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDD7+ASeFHETvhMHnb3JXMC9R9Fh8KDe3pcXjpLMP8ATxwT
XPGAkrt5IwSsywC1wSfB6F2G2Z3EmTLs664PVXbb4coRKnFwA5ffZEjt6CQfo+92RyTjG17sDc/1
WJAYAt3spMeZZ2c2089QGwtuVE4kQfj0386JZJ5nufS1SCb9g3qCiJYjgAaoLv3fYuu/bwWHcR4U
SX/3AHu9QcHN7cP6TX72eUU+ftZtgSjs7Nj82XD9UkI7x3z9rAtLkQrlQQombGlQ5i0wsKG08ORI
r+hEGMC0FrdUajd7PRJ4nk3zo3T6CwATFERcP83+JNbGrbbb1pAbFs5oa+M/k6GwKzbL
stagiaire
```

Enregistrez et fermez.

Enfin :

```
emacs ~/.ssh/config
```

Collez dans le fichier le contenu suivant :

```
Host formation
Hostname <nom.de.votre.machine.de.formation>
IdentityFile ~/.ssh/id_rsa.stage
User stagiaire
```

Désormais, pour lancer une session ssh sur le serveur, il vous suffit de taper :

```
ssh formation
```

Architecture et organisation d'une install Sympa

Dans cette formation, nous avons fait l'impasse sur l'installation de Sympa pour vous laisser plus de temps à manipuler le logiciel. Vous pourriez légitimement trouver que ça manque. Nous sommes d'accord. En conséquence de quoi nous vous avons laissé l'ancienne [doc d'installation de Sympa dans les bonus](#). Nous vous invitons à la consulter si vous désirez voir comment installer Sympa depuis les sources (la méthode employée pour le TP).

Présentation powerpoint (15 min) :

- système de fichier
 - configuration
 - spools
 - autres
- démons
- les listes
- Les rôles
- Les niveaux de configuration

Concepts de base

Personnalisation de l'installation

Avant toute chose, enregistrez-vous comme listmaster dans la configuration de Sympa

```
emacs /etc/sympa/sympa.conf
```

Trouvez la ligne suivante :

```
listmaster david@webmail.com,etienne@webmail.com,olivier@webmail.com
```

Remplacez l'adresse email par les adresses de listmaster que vous comptez employer (séparées par des virgules). Enregistrez et fermez.

Redémarrez les processus :

```
service sympa restart && service apache2 restart
```

Note : Après le redémarrage d'Apache, vous ne devez plus avoir de processus wwsympa.fcgi qui tourne. Si ce n'est pas le cas, tapez la commande suivante :

```
pkill wwsympa.fcgi
```

1. Se connecter comme listmaster

Accédez à l'interface web de Sympa : <https://nouveau.domaine.fr/sympa>

Dans l'interface web, cliquez sur l'icone de bonhomme en haut à droite puis sur « Première connexion ? ». Saisir une des adresses email définie comme listmaster, puis valider. Vérifier la boîte mail, Sympa a envoyé un message. Cliquez sur le lien de validation. Vous êtes redirigé vers Sympa. Saisir alors votre mot de passe.

2. Créer une liste.

Cliquez sur « Création de liste », choisissez le modèle de liste « Liste de discussion publique » puis remplir le formulaire et valider.

Une fois la liste créée, vous apparaissez comme gestionnaire dans le menu de la liste.

Un certain nombre de fichiers ont été créés dans le répertoire de la liste :

```
ls -l /usr/local/sympa/list_data/<nom-de-la-liste>
```

Vous devez avoir la sortie suivante :

```
config
config.0
config.bin
info
```

3. Abonnez des adresses à la liste

Pour vous abonner vous-même à la liste, il vous suffit de cliquer sur le lien “Subscribe” dans le menu de gauche de la liste.

Cliquez sur l'icone d'œil à gauche de “Subscribers” dans le menu de liste, puis ajoutez des adresses dans le formulaire.

Vous pouvez ajouter plusieurs adresses en cliquant sur « Abonnement par lot » et en mettant une adresse par ligne.

Attention : Si vous collez plusieurs milliers d'adresses, vous risquez de tomber sur une limite de modfcgi: MaxRequestLen. Par défaut, une requête sur mod_fcgid est limité à 131 ko. Si vous la dépassez, une erreur 500 est renvoyée en réponse. Vous pouvez augmenter cette valeur en ajoutant ce paramètre dans la section `IfModule mod_fcgid` de ce paramètre dans `/etc/apache2/mods-enabled/fcgid.conf`

4. Envoyer un message à la liste

Une fois le message diffusé, vous pouvez vérifier qu'il a été archivé en cliquant sur « Archives ». Les logs affichent des infos.

5. Vue abonné

Créer un compte pour une des adresses d'abonnés qui n'est pas une adresse de listmaster. Se connecter avec cette adresse et constater la différence d'options offertes dans l'interface web ...

Endosser une identité

En tant que listmaster, vous avez la possibilité, dans l'interface web, de visualiser ce qu'un autre utilisateur voit, tout simplement en endossant son identité.

Pour cela, suivez le chemin suivant : *Administrateur des listes* → *les utilisateurs*

En bas de la page vous tombez sur la section "Endosser l'identité d'un utilisateur". Tapez n'importe quelle adresse email (même une qui n'existe pas) et cliquez sur "Changer de contexte utilisateur".

Vous avez désormais l'identité correspondant à cette adresse.

Vous pouvez à tout moment retrouver votre identité en cliquant sur cette adresse, en haut à droite de Sympa, puis sur le bouton "Restaurer l'identité".

6. Ajouter un modérateur

Aller dans « Admin → Configurer la liste → Définition de la liste ». Ajouter une adresse dans la section « modérateurs ».

7. Options de configuration de la liste

Le propriétaire de la liste peut éditer de nombreux paramètres de configuration. Les privilèges affectés à chaque rôle sont gouvernés par un fichier de configuration, `edit_list.conf`. Pour chaque paramètre éditables via l'interface web, ce fichier définit, pour chaque rôle, un niveau d'accès :

- `hidden` : le paramètre est invisible
- `read` : la paramètre et sa valeur sont visibles, mais pas éditables
- `write` : le paramètre est éditable.

Avant toute personnalisation d'un fichier, il faut commencer par copier la version distribuée par Sympa (qui se trouve dans le répertoire « default ») vers la destination idoine :

- `etc` pour disposer d'une portée globale ;
- `etc/<nom_robot>/` (si vous utilisez des robots) pour disposer d'une portée sur le robot ;
- `list_data/[nom_liste]` pour disposer d'une portée de liste.

Personnaliser le fichier `edit_list.conf` :

```
cp -p /usr/local/sympa/default/edit_list.conf /usr/local/sympa/etc
emacs /usr/local/sympa/etc/edit_list.conf
```

Rendre le paramètre `include_sql_query` éditable pour un propriétaire privilégié.

Enregistrer le fichier et fermer.

Authentifiez-vous avec l'adresse de propriétaire privilégié prédéfinie à l'étape précédente.

Allez voir dans la configuration de la liste, section « Sources de données ». Une nouvelle section est disponible pour configurer une source de données de type SQL.

8. Personnalisation des listes

De nombreuses personnalisations sont possibles au sein d'une liste. Nous allons en présenter quelques-unes.

Footer et header

Vous pouvez ajouter des entêtes et des pieds de page aux messages.

Exemple avec un pied de page :

```
emacs /usr/local/sympa/list_data/<nom-de-la-liste>/message.footer
```

Taper un texte quelconque, enregistrer, fermer.

Envoyer un message à la liste : le footer apparaît en bas du message. Notez que, par défaut, le footer est inséré en tant que pièce jointe.

Homepage

Un fichier dans lequel vous pouvez mettre un fragment HTML qui sera présenté sur la page d'accueil de la liste (attention : seules les balises HTML acceptées entre les balises `<body>` et `</body>` seront prises en compte).

```
emacs /usr/local/sympa/list_data/<nom-de-la-liste>/homepage
```

9. Créer un modèle de liste personnalisé

“Avant toute personnalisation dans Sympa, il faut copier le modèle là où on veut appliquer la personnalisation”

Le Tao de Sympa - 15:3

Ce qu'on peut reformuler par : Toute personnalisation effectuée directement dans “default” sera écrasée à la prochaine mise à jour. Il faut donc commencer par copier ce qu'on veut personnaliser

depuis `default` vers `etc` pour une portée à l'échelle du serveur, ou vers `list_data` pour une portée à l'échelle d'une liste.

Ou encore : "NE MODIFIEZ QUE DES COPIES, JAMAIS L'ORIGINAL !"

Voilà, et donc pour créer votre modèle personnalisé, voilà ce qu'il faut faire :

```
mkdir /usr/local/sympa/etc/create_list_templates/  
cp -pr /usr/local/sympa/default/create_list_templates/discussion_list  
/usr/local/sympa/etc/create_list_templates/  
chown -R sympa:sympa /usr/local/sympa/etc/create_list_templates/  
cd /usr/local/sympa/etc/create_list_templates/discussion_list
```

Contenu du répertoire de modèle :

```
ls -al
```

```
total 24  
drwxrwxrwx 2 sympa sympa 4096 mai 2 17:32 .  
drwxrwxrwx 10 sympa sympa 4096 jan 22 14:51 ..  
-rw-r--r-- 1 sympa sympa 202 mai 2 17:32 comment.tt2  
-rw-r--r-- 1 sympa sympa 835 mai 2 17:32 config.tt2
```

Les deux fichiers trouvés sont :

- `comment.tt2` : le message affiché dans l'interface de création de listes
- `config.tt2` : le modèle de configuration de la liste

Dans notre exemple, on va personnaliser le `config.tt2` :

```
emacs config.tt2
```

```
## Configuration de la liste sympa-l  
## Cree le Mercredi 31 Mars 99  
#send editorkey  
  
subject [% subject %]  
  
status [% status %]  
  
visibility noconceal  
  
subscribe open_notify  
unsubscribe open_notify  
  
[% FOREACH o = owner -%]  
owner  
    email [% o.email %]  
    profile privileged  
    [% IF o.gecos -%]
```

```
gecos [% o.gecos %]
[% END -%]

[% END -%]

[% IF owner_include %]
[% FOREACH o = owner_include -%]
owner_include
  source [% o.source %]
[% END -%]
[% END -%]

send privateoreditorkey

[% IF topics -%]
topics [% topics %]
[% END -%]

process_archive on

archive
  web_access public
  mail_access owner

digest 1,4 13:26

review owner

shared_doc
d_edit default
d_read public

pictures_feature on

creation
  date_epoch [% creation.date_epoch %]
  date        [% creation.date %]
[% IF creation_email -%]
  email        [% creation_email %]
[% END -%]
```

Remplacez `process_archive on` par `process_archive off`: toutes les listes créées à partir de ce modèle ne seront pas archivées.

La messagerie dans Sympa

1. Flux de diffusion des messages

Jouons avec les démons !

Préparons le terrain :

- Ajouter une adresse qui ne marche pas dans la liste des abonnés. - Passer la liste en modération.
Dans Admin → configurer la liste → Diffusion/réception, sélectionner l'option "Modérée, même pour les modérateurs" dans le menu déroulant "Qui peut diffuser des messages".
Ne pas oublier d'enregistrer la configuration après.

Arrêter tous les démons Sympa.

```
service sympy stop
```

Envoyer un message à la liste. Aller voir dans le spool msg : le message s'y trouve.

```
ll /usr/local/sympa/spool/msg
```

Démarrer le démon sympy_msg.pl.

```
/usr/local/sympa/bin/sympy_msg.pl
```

Regarder le spool msg.

```
ll /usr/local/sympa/spool/msg
```

Les logs affichent la prise en compte du message.

```
tail /var/log/syslog
```

Le message est désormais dans le spool moderation :

```
ll /usr/local/sympa/spool/moderation
```

Accepter la diffusion du message.

Regarder le spool outgoing : le message s'y trouve (sauf si ? À votre avis ?)

```
ll /usr/local/sympa/spool/outgoing
```

Regarder le spool bulk : il y a une entrée dans le sous-spool msg et une autre dans pct.

```
ll /usr/local/sympa/spool/bulk/pct
ll /usr/local/sympa/spool/bulk/msg
```

- **Dans msg** : un fichier avec le format [priorité-liste].[priorité message].[timestamp réception].[timestamp diffusion].[nombre aléatoire].[nom liste]@[domaine liste]_z,6329,8970. Ce fichier contient le texte source du message, - **Dans pct** : un répertoire

avec le format [priorité-liste].[priorité message].[timestamp réception].[timestamp diffusion].[nombre aléatoire].[nom liste]@[domaine liste]_z,6329,8970. Ce répertoire contient un certain nombre de fichiers, chacun contenant une liste d'adresses email d'abonnés. Chaque fichier correspond à une session SMTP à créer.

Démarrer bulk.pl : le message est distribué et disparaît du spool bulk.

```
/usr/local/sympa/bin/bulk.pl
```

Démarrer archived.pl : le message est archivé, et disparaît du spool outgoing.

```
/usr/local/sympa/bin/archived.pl
```

Regarder dans le spool bounce : le bounce pour l'adresse qui ne marche pas devrait s'y trouver.

```
ll /usr/local/sympa/spool/bounce
```

Démarrer bounced.pl.

```
/usr/local/sympa/bin/bounced.pl
```

Les logs affichent le traitement du bounce. Il disparaît du spool bounce. L'adresse est marquée en erreur dans la liste des abonnés et apparaît dans la liste des adresses en erreur.

Remarque : si vous subissez un crash d'un des processus Sympa, vous n'aurez sans doute pas d'info dans les logs. Il est cependant possible de récupérer les dernières erreurs dans le spool tmp. À chaque fois qu'un processus Sympa démarre, il crée un fichier [pid].stderr dans le spool tmp (où [pid] vaut le numéro de processus). L'erreur standard de chaque processus est écrite dans ce fichier.

Vérifier cela avec la commande

```
ls -ltr /usr/local/sympa/spool/tmp
```

2. Modes de réception

- mail : réception normale
- notice : réception d'un message vide, avec seulement le sujet du message original
- digest : réception d'une compilation régulière des messages (en général, une fois par jour, ceci est configurable via la paramètre de liste « digest ».
- summary : réception régulière (au même rythme que les digest) d'un message donnant quelques informations sur les messages diffusés dans la liste
- nomail : ne pas recevoir les messages
- txt : si le message original est au format multipart/alternative, ne recevoir que la partie text/plain. **Attention** : ça supprime tout le reste, même les pièces jointes...
- html : si le message original est au format multipart/alternative, ne recevoir que la partie text/html. **Attention** : ça supprime tout le reste, même les pièces jointes...
- urlize : certaines pièces jointes peuvent être trop grosses. Si un utilisateur a sélectionné cette option, les pièces jointes trop grosses sont conservées sur le serveur et remplacées, dans le message, par un lien permettant de les télécharger.

- `not_me` : ne pas recevoir ses propres messages.

Remarque : quelle que soit l'option de réception choisie, Sympa n'altèrera jamais un message signé S/MIME (pas d'ajout de footer par exemple).

Deux paramètres peuvent être utilisés pour contrôler la taille des messages diffusés :

- `urlize_min_size` : définissable seulement au niveau de `sympa.conf` : définit la taille minimale (en octets) des pièces jointes en deçà de laquelle elles ne seront pas stockées sur le serveur.
- `max_size` : définissable au niveau de `sympa.conf`, de `robot.conf` ou de chaque liste. Indique la taille maximale (en octets) autorisée pour diffuser un message. Ce filtre intervient avant qu'on se préoccupe des options de réception.

Indépendamment des options d'abonnement, Sympa ne diffusera pas un message dont la taille dépasse `max_size` (même si tout le monde est abonné en `urlize`).

Si vous le désirez, vous pouvez utiliser les fichiers ci-dessous pour faire des tests avec des pièces jointes de taille croissante :

-  : Image de 179 ko
-  : Image de 2,7 Mo
- 
- 
- 

3. Système de prévention des boucles

Sympa dispose de mécanismes de prévention des boucles de messagerie. L'un des points testés est l'identifiant du message. Pour vérifier cela :

- Arrêter Sympa ;
- Envoyer un message à une liste ;
- Copier le message des spools dans `/tmp` ;
- Relancer Sympa : le message est diffusé ;
- Copier le message depuis `/tmp` vers les spools : il est rejeté.

4. Conversion d'une page web en message

Sympa permet d'envoyer des pages web comme messages. L'un des intérêts de cette fonctionnalité est que Sympa s'assure que toutes les images sont bien embarquées dans le messages (multipart/related) et non à télécharger. Cela limite le risque que le messages soit taggué comme spam.

Dans le menu de liste cliquer sur « Poster », cliquer sur l'onglet « Envoyer une page web ».

Taper une URL dans le champ « Récupérez la page depuis une URL ». Taper un sujet pour le mail puis cliquez sur « envoyer au destinataire sélectionné ».

Vous pouvez consulter la source du message pour observer que Sympa a bien inclus toutes les images en tant que pièces jointes.

5. Paramétrage

Gérer la cadence de sortie des messages :

Vous pouvez définir le nombre de processus `bulk.pl` qui vont tourner. Plus il y en a, plus vite vous éliminerez un gros nombre de message, en particulier si les connexions sortantes sont lentes.

Pour tester cela, créons une liste avec un grand nombre d'utilisateurs. On ajoute un grand nombre d'adresse de la forme "pipo+[nombre]@nouveau.domaine.fr". Le serveur est configuré pour que tout mail envoyé à cette adresse parte dans `/dev/null`. de cette façon, nous pouvons tester Sympa sans avoir à réellement gérer le flux sortant de messagerie.

Créer un fichier source d'adresses email du même domaine :

```
mkdir /usr/local/sympa/includes/  
chown sympa:sympa /usr/local/sympa/includes/  
for i in $(seq 1 10000) ; do echo "pipo+$i@nouveau.domaine.fr" ; done >  
/usr/local/sympa/includes/enormeliste.incl
```

Dans une liste au choix, inclure le fichier produit comme source d'abonnés (les options d'abonnements à partir de sources externes seront présentées en détail à la fin du TP).

- Via l'interface web, allez dans **Admin → Configurer la liste → Définition des sources de données**.
- Dans le champ "Inclusion d'un fichier", tapez le chemin vers le fichier `/usr/local/sympa/includes/enormeliste.incl`.
- Cliquez sur "Mise à jour" en bas de la page. Il y a de fortes chances que vous rencontriez une erreur 500. C'est parce que l'inclusion de 10000 abonnés prend un temps qui dépasse le timeout d'Apache. Ce n'est pas gênant : les abonnés seront tout de même inclus.

Lancez un

```
tail -f /var/log/syslog
```

Envoyer un message à la liste.

Dans les logs, vous trouverez une entrée contenant « Done sending message » ainsi que le temps de diffusion.

Éditez le fichier de configuration de Sympa.

```
emacs /etc/sympa/sympa.conf
```

Ajoutez la ligne suivante :

```
bulk_max_count 10
```

Relancez Sympa :

```
service sympa restart
```

Refaites l'opération précédente et remarquez la différence de temps de traitement.

Le nombre de bulk n'est pas le seul paramètre de réglage de la messagerie ;

Dans `sympa.conf`, vous pouvez également jouer sur trois paramètres :

- `maxsmtp` : le nombre maximal de processus `sendmail` que Sympa lancera
- `nrcpt` : le nombre maximal d'adresses du même domaine utilisées dans une même session `sendmail`.

Ajoutez les lignes suivantes dans `sympa.conf` :

```
maxsmtp 100  
nrcpt 500
```

Rejouez l'opération d'envoi de mail et observez la différence.

Ces deux paramètres, bien que fort utiles, doivent être maniés avec prudence :

- un `maxsmtp` trop important peut écraser votre machine à cause du trop grand nombre de processus `sendmail` qui tournent
- un `nrcpt` trop élevé peut vous faire tagguer comme spammeurs.

Certains domaines (yahoo et gmail, par exemple) risquent de refuser des sessions qui arrosent trop d'utilisateurs d'un coup. Vous pouvez limiter le nombre de destinataires pour des domaines précis en ajoutant, dans le fichier `nrcpt_by_domain.conf`, des règles spécifiques aux domaines.

```
emacs /usr/local/sympa/etc/nrcpt_by_domain.conf
```

Ajouter la ligne suivante :

```
nouveau.domaine.fr 3
```

Et redémarrez Sympa :

```
service sympa restart
```

Relancez l'envoi de message et observez la différence.

Enfin, un dernier paramètre permet de limiter le nombre de domaines utilisés dans une même session `smtp` : [avg](#).

6. gestion des erreurs

6.1 Scores d'erreur

Sympa gère automatiquement les erreurs. Il affecte un score aux adresses en erreur. Le score permet à ces adresses d'être classés suivant trois seuils :

- avant le niveau 1 : rien de spécial à faire
- niveau 1 : une première action peut être engagée
- niveau 2 : en général, l'adresse est désabonnée.

Les niveaux 1 et 2 sont définis par les paramètres de liste `bouncer_level_1` et `bouncers_level_2`, accessible dans la config des listes (Admin → configurer la liste → Gestion des erreurs). Vous voyez, en accédant à ces paramètres, que pour chaque niveau, vous avez :

- un score de déclenchement de l'action
- un action possible : rien, avertir, désabonner
- une personne à prévenir de l'action effectuée.

Le calcul du score repose sur trois paramètres :

- Bounces count: Le nombre de bounces reçus pour cette adresse
- Type rate: Les bounces sont classés suivant la gravité de l'erreur. Pour une erreur mailbox is full (une erreur temporaire 4.2.2), le type rate sera de 0,5, alors que les erreurs permanentes (5.x.x) auront une valeur de 1
- Regularity rate: Score obtenu en comparant les nombre de messages reçus au trafic général de la liste, déduit du contenu du fichier `msg_count`.

Le calcul du score est fait selon la formule :

```
Score = bounce_count * type_rate * regularity_rate
```

Pour éviter de calculer un score précocement, ce calcul n'est entamé que si :

- Le nombre de bounce reçus est supérieur au paramètre `minimum_bouncing_count`.
- La période de temps courant entre le premier et le dernier bounce reçu est inférieure à la valeur du paramètre `minimum_bouncing_period`.

Les utilisateurs qui ne génèrent plus de bounce voient leur score baisser au bout d'un moment, défini par le paramètre `expire_bounce` (10 jours par défaut).

6.2 VERP

Variable enveloppe return path. Cette fonctionnalité permet d'encoder, dans le return-path du message, l'adresse d'expédition originale. Certains messages d'erreurs ne mentionnent en effet pas cette adresse (c'est le cas pour une redirection d'adresse email).

Toute adresse ayant généré une erreur est automatiquement envoyée en VERP.

Le VERP a un coût : comme chaque message est individualisé, il nécessite une session SMTP pour lui seul (impossible de grouper les envois par domaine, par exemple).

Vous pouvez régler le taux de VERP par défaut via le paramètre `verp_rate` du fichier de config de liste. Par défaut, il vaut le paramètre `verp_rate` de `sympa.conf`, dont la propre valeur par défaut est de 0 %.

Les interfaces de Sympa :

0.1 Ligne de commande

Sympa.pl peut être lancé en ligne de commande. Pour avoir toutes les options possibles :

```
/usr/local/sympa/bin/sympa.pl --help
```

Il est ainsi possible de créer une liste en ligne de commande. Cette fonctionnalité sera surtout utile pour instancier des familles de listes (voir formation « usages avancés »).

0.2 Mail

Vous pouvez effectuer des commandes par mail, adressées à l'adresse du robot Sympa (par défaut, sympa@nouveau.domaine.fr).

Toutes les commandes sont décrites dans l'aide en ligne de Sympa. Cet usage tend à diminuer, du fait de l'interface web, principalement.

0.3 Web

Vous connaissez désormais bien l'interface web de Sympa.

Vous pouvez personnaliser cette interface web de deux manière : via la feuille de style, ou via les templates.

Changez les couleurs de l'interface web en vous rendant dans le menu d'administration de Sympa. Onglet « Admin Sympa » en haut de l'interface web, puis « Skins, feuille de style et couleurs ».

Une fois les couleurs sélectionnées, cliquez sur « Installer mes couleurs de session dans une nouvelle CSS statique ».

Vérifier qu'une nouvelle CSS a bien été créée dans `/usr/local/sympa/static_content/css`. Un nouveau fichier « `style.css` » devrait y avoir été généré.

L'interface web est générée à chaque requête, à partir de templates fondés sur le langage TT2.

0.4 SOAP

Sympa offre un webservice par le biais d'une interface [SOAP](#).

Pour qu'un utilisateur puisse manipuler Sympa à travers un programme tiers, celui-ci doit envoyer des

credentials authentifiant l'utilisateur.

Le cas le plus courant (mais il n'est pas le seul) est d'établir une relation de confiance entre Sympa et l'application.

Autorisation

“Quand on veut faire un truc, dans la moitié des cas on trouve la solution avec un scénario.”

Le Tao de Sympa - 666:13

L'attribution de privilèges, dans Sympa, se fait par le biais de scénarios d'autorisation. Ainsi, le droit d'envoyer un message, de voir la liste des abonnés, d'accéder aux archives, etc. sont contrôlés par scénario.

La liste suivante de paramètres de configuration de Sympa ou d'une liste correspondent à de scénarios :

- `access_web_archive` : droit de consulter les archives web
- `add` : droit d'ajouter un abonné à la liste
- `automatic_list_creation` : droit de créer des listes automatiques
- `create_list` : droit de créer une liste via l'interface web
- `shared_doc.d_edit` : droit d'éditer les documents partagés
- `del` : droit de supprimer un abonné d'une liste
- `shared_doc.d_read` : droit de consulter les documents partagés
- `global_remind` : droit d'envoyer à tous les abonnés d'un serveur un rappel d'abonnement
- `info` : droit de consulter la description d'une liste
- `invite` : droit d'inviter une personne à s'abonner à une liste
- `remind` : droit d'envoyer un rappel d'abonnement aux abonnés d'une liste
- `review` : droit de consulter la liste des abonnés à une liste
- `send` : droit d'envoyer un message à une liste
- `subscribe` : droit de s'abonner à une liste
- `topics_visibility` : droit de voir un topic via l'interface web
- `tracking` : droit de consulter les données de suivi de la délivrance des messages
- `unsubscribe` : droit de se désabonner d'une liste
- `visibility` : droit de voir qu'une liste existe.

En pratique, un scénario est un fichier contenu par le répertoire « `scenari` » de l'échelle de travail considérée. En effet, tous les scénarios distribués se trouvent dans le répertoire `/usr/local/sympa/default/scenari/`.

Il est par la suite possible de personnaliser les scénarios à un niveau plus précis : site (dans le répertoire `/usr/local/sympa/etc/scenari`) robot (`/usr/local/sympa/etc/'nom_robot'/scenari/`) ou liste (`/usr/local/sympa/list_data/scenari/`).

Si, dans la config d'une liste, le paramètre « `send` » a la valeur « `private` », cela signifie que, quand quelqu'un voudra envoyer un message à la liste, le scénario évalué sera contenu par le fichier « `send.private` ». On cherchera, au moment d'envoyer un message, le scénario en question dans le répertoire de la liste, puis d'un éventuel robot, puis du site et enfin dans les scénarios distribués.

Un fichier de scénario est une succession de ligne ayant toutes la même structure :


```
<test> <méthode d'authentification> -> <action>
```

Un test effectue des opérations sur des variables. Il renvoie vrai ou faux. Par exemple, le test « `is_subscriber([email],[nom_liste])` » renvoie vrai si l'adresse [email] est abonnée à [nom_liste].

Une méthode d'authentification spécifie de quelle manière l'utilisateur est authentifié. Cela recouvre différentes réalités suivant le contexte :

Méthode	Contexte mail	Contexte web
smtp	On se fie au contenu du champs « From : » du mail	Inutilisable dans ce contexte
smime	Le mail est signé S/MIME	Le navigateur a un certificat X509 installé
md5	Un hash MD5 situé dans le sujet du message (utilisé pour les commandes mail)	L'utilisateur est authentifié sur l'interface web
dkim	Le mail est signé DKIM	Inutilisable dans ce contexte

L'**action** est ce que Sympa va faire *si le test est réalisé* (parce que la méthode d'authentification est utilisée) *et que le test est vrai* :

- Si elle vaut « `do_it` », l'action demandée est effectuée (le message est envoyé, la personne est abonnée, etc.),
- Si elle vaut « `reject` », l'action est impossible
- Les autres valeurs sont des modérations : l'action sera effectuée si une personne l'autorise (selon les cas, le listmaster, le propriétaire ou le modérateur).

Un scénario peut contenir un nombre illimité de règles. Elles sont évaluées les unes après les autres, de haut en bas. Dès qu'une règle est vérifiée, Sympa applique l'action spécifiée et arrête l'évaluation du scénario.

Exemple de scénario (`send.private_smime`) :

```
is_subscriber([listname],[sender]) smime -> do_it # accepte d'envoyer un
message si l'utilisateur est abonné à la liste et que son message est signé)
is_editor([listname],[sender]) smime -> do_it # accepte d'envoyer un message
si l'utilisateur est modérateur de la liste et que son message est signé)
is_owner([listname],[sender]) smime -> do_it # accepte d'envoyer un message
si l'utilisateur est propriétaire de la liste et que son message est signé)

true() smtp,dkim,md5,smime -> reject(reason='send_subscriber_smime') # Pour
toute autre méthode d'authentification, le message est rejeté et un message
de service est envoyé à l'utilisateur pour lui expliquer que ses messages
doivent être signés.
```

Bon à savoir : Sympa possède un mécanisme de débogage de scénarios (au moins jusqu'à la version 6.2.32).

Pour vous en servir :

```
emacs /etc/sympa/sympa.conf
```

Ajoutez les lignes suivantes :

```
log_condition email=<une.adresse.email>
```

```
log_module scenario
```

Puis redémarrez Sympa.

```
service sympa restart
```

À partir de ce moment, Sympa va journaliser le détail de chaque évaluation de scénario pour l'adresse email concernée.

Notez que log_conditon accepte également de filtrer sur les adresses IP. Exemple :

```
log_condition email=<une.adresse.email>,ip=127.0.0.1
```

ou un IP seule.

```
log_condition email=ip=127.0.0.1
```

log_module n'accepte qu'un valeur : scenario.

Il semble, hélas, que ce paramètre ait été rendu obsolète dans les toutes dernières versions de Sympa. Nous verrons si nous parvenons à convaincre la communauté de le réintroduire.

Petit exercice :

“On ne peut faire confiance à personne.”

C'est ce que vous répète votre RSSI depuis qu'il a vu “Game of Thrones”. Vous lui aviez bien dit que ça n'était pas un spectacle adapté pour lui mais il n'en a fait qu'à sa tête ; et maintenant il sursaute à chaque fois qu'il croise quelqu'un dans le couloir. D'ailleurs il a un peu maigri.

Bref, quand il vient vous voir pour vous dire qu'il est indispensable que les enseignants puissent écrire aux étudiants, mais seulement s'ils prouvent qui ils sont, vous n'avez pas le cœur de le contredire.

C'est une petite université : tous les étudiants sont dans une liste, et les enseignants dans une autre. Débrouillez-vous pour que les enseignants puissent écrire aux étudiants, mais seulement après authentification via l'interface web.

Vous pouvez vous aider de la documentation des scénarios (https://sympa-community.github.io/manual/man/sympa_scenario.5.html).

Inclusion de scénarios

Il est possible d'inclure des scénarios dans d'autres grâce à la ligne suivante :

```
include general
```

Alors Sympa va inclure toutes les règles du scénario nommé “include.general” à l'emplacement de la

ligne.

Il est possible d'inclure systématiquement dans tous les scénarios un autre scénario en le postfixant avec la chaîne ".header". Ainsi le fichier "include.send.header" sera systématiquement inclus à tous les scénarios "send" de son niveau (robot ou serveur, suivant l'emplacement dans lequel il se trouve).

Petit exercice (encore) :

Votre université héberge de dangereux agitateurs qui complotent pour qu'advienne une aube nouvelle où toutes les ampoules électriques seront clignotantes.

Bon.

Les RG débarquent à la DSI et, commission rogatoire à l'appui, exigent de pouvoir accéder à tous les documents partagés de votre serveur de listes.

Ils ont l'air nerveux et pressés. Débrouillez-vous pour que l'un d'entre eux ait les accès demandés avant qu'ils vous embarquent pour complicité.

Sources de données pour l'alimentation des listes

Un de atouts de Sympa est de permettre de définir automatiquement la liste des abonnés à partir d'une source externe.

Vous avez déjà expérimenté cela avec un fichier d'adresses email plus tôt dans le TP.

Sympa peut alimenter les listes à partir des sources suivantes :

- fichier texte local d'adresses email (une adresse email par ligne)
- fichier d'adresse email distant (ou toute URL renvoyant du plain text et contenant une adresse email par ligne)
- autre liste du serveur
- liste située sur un autre serveur (nécessite une configuration sur chacun des deux serveurs)
- requête dans une base de données SQL
- requête dans un annuaire LDAP

Création d'une table d'adresses fictives

Éditer le fichier de configuration de la liste que vous avez utilisée dans la partie « Messagerie / performances ».

Ajouter le paragraphe suivant :

```
include_sql_query
db_name dummy
db_type mysql
```

```
host localhost
user francis
passwd AnSCB3g23etr3
sql_query SELECT email FROM dummy_emails
```

Enregistrez et fermez.

Allez consulter la liste des abonnés via l'interface web : un message apparaît signalant que la liste a été mise à jour.

Observez les premiers abonnés : dans la colonne « Source », plusieurs sources apparaissent : le fichier utilisé dans la partie précédente et la source SQL (nommée d'après le serveur SQL contacté, donc “localhost”, ici).

La liste des membres est mise à jour depuis les sources de données en plusieurs circonstances :

- régulièrement, via un tâche du gestionnaire de tâches. La fréquence de mise à jour est gouvernée par le paramètre ttl de la liste
- à chaque fois que la liste des membres est consultée ou qu'un message est envoyé. Afin d'éviter de surcharger le serveur, un délai minimum entre ces synchronisation déclenchées par l'utilisateur est défini dans la configuration de la liste, gouverné par le paramètre « distribution_ttl ».

Pour déclencher une synchronisation, vous pouvez ainsi simplement supprimer la tâche de synchronisation en attente : elle sera immédiatement recréée, puis exécutée.

```
cd /usr/local/sympa/spool/task/
find . -name 'sync_include*' -exec rm -f '{}' \; ; tail -f /var/log/syslog
```

Vous voyez passer dans les logs le message de recréation de la tâche de synchronisation.

```
ls -ltr
```

Ce déclenchement est également possible via l'interface web, dans la partie « liste des membres ».

Inclusion d'une liste avec un filtre

Créez une nouvelle liste. Dans la configuration de ses sources de données, sous “Inclusion d'une liste (include_list)” ajoutez

1. nom court pour cette source (name) : Ce que vous voulez
2. Nom de liste à inclure (listname) : nom_de_la_liste_de_l_exercice_precedent
3. Définition du filtre (filter) :

```
email.match('^pipo\+[0-9]@')
```

Régénérez la liste des abonnés, elle comporte maintenant les abonnés “pipo” de 1 à 9.

Plus d'informations sur la syntaxe des filtres :

https://sympa-community.github.io/manual/man/list_config.5.html#include_sympa_list

Exclusions

L'exclusion permet de désabonner une personne incluse depuis une source externe. En pratique, Sympa ajoute l'adresse email de la personne concernée dans une table (`exclude_table`). Les adresses emails contenues par cette table sont soustraites de la liste des membres quand elle est demandée.

L'exclusion n'a pas de paramètre d'activation / désactivation : elle est toujours possible, dès lors que, via les scénarios d'autorisation, il est possible de supprimer des abonnés (scénarios « subscribe » et « del »).

Inclusion de propriétaires / modérateurs

De la même manière que l'on peut inclure des abonnés dans une liste, il est possible de tirer la liste des propriétaires et modérateurs d'une source de données externes.

Le mécanisme est très similaire à celui employé pour les abonnés, à ceci près que le paragraphe de configuration n'est pas écrit directement dans le fichier de configuration de la liste, mais dans un fichier externe. Cette mesure supplémentaire ne provient pas d'une volonté sadique des développeurs, mais d'un souci de factorisation. En effet, l'inclusion de propriétaires ou de modérateurs est définie par le paramètre `owner_include` ou `editor_include`. Ces paramètres peuvent contenir un champ nommé « `source_parameters` » contenant une liste de valeurs séparées par des virgules, qui seront interprétées lors de l'analyse de la source de données.

Créer un fichier source de données :

```
mkdir /usr/local/sympa/etc/data_sources
chown sympa:sympa /usr/local/sympa/etc/data_sources
emacs /usr/local/sympa/etc/data_sources/my_owners.incl
```

Dans l'éditeur, ajouter le texte suivant :

```
include_sql_query
db_type mysql
host localhost
user francis
passwd AnSCB3g23etr3
db_name dummy
sql_query SELECT DISTINCT email FROM dummy_emails WHERE email LIKE
'pipo+[%param.0%]%%'
```

Note : Remarquez que, pour pouvoir utiliser le caractère '%' avec sa valeur spéciale dans la requête SQL, nous avons dû le doubler. En effet, Sympa utilise un `sprintf` pour préparer les requêtes.. Un '%' seul déclencherait donc une erreur.

Enregistrez et fermez.

Dans la configuration de la liste, ajoutez le paramètre suivant :

```
owner_include
source my_owners
source_parameters 9
reception mail
profile privileged
```

Enregistrez et fermez. Allez voir la liste dans l'interface web pour constater l'ajout des nouveaux propriétaires. Si vous changez la valeur du paramètre, la liste des propriétaires change.

Attention ! La version de Sympa employée présente un bug de cache des configs de listes (corrigée dans les version postérieures). Il est possible que vous ayez à redémarrer les processus Apache et Sympa pour que le fichier de définition des propriétaires soit pris en compte.

Petit exercice :

Finalement, votre RSSI pense que c'est mieux si les enseignants sont propriétaires de la liste des étudiants.

Ses mains tremblent, signe qu'il est à court de pilules de grenouilles séchées.

Faites-lui plaisir. Ce sera mieux pour tout le monde.

Rappel : les étudiants sont dans une liste, les enseignants dans une autre.

Hôtes virtuels

Nous allons transformer l'installation pour tout localiser dans un hôte virtuel.

A vrai dire, c'est une opération qui aurait dû être faite dès l'installation. En effet, nous recommandons que, même si vous ne gérez qu'un seul domaine au départ, vous le configuriez comme un hôte virtuel. De cette manière, si vous avez un jour à ajouter un hôte virtuel, vous ne risquez pas de vous retrouver avec des configuration de listes ou de robot sur plusieurs niveaux.

Cela dit, le transfert d'une installation existante sur un hôte virtuel est une opération intéressante, car lors de la reprise d'un existant, vous risquez d'être amenés à la réaliser.

Création de l'hôte virtuel

Configuration Sympa

```
mkdir /usr/local/sympa/etc/nouveau.domaine.fr
chown sympa:sympa /usr/local/sympa/etc/nouveau.domaine.fr
cd /usr/local/sympa/etc/nouveau.domaine.fr
emacs robot.conf
```

Copiez le contenu suivant dans l'éditeur (**en remplaçant <une.adresse.email> par une vraie adresse**) :

```
## Main robot hostname
# was domain domain.tld

domain nouveau.domaine.fr

listmaster      <une.adresse.email>

## Who is able to create lists
## This parameter is a scenario, check sympy documentation about scenarios
if you want to define one
create_list      public_listmaster

title <Un titre pour le service>

http_host nouveau.domaine.fr

wvsympa_url https://nouveau.domaine.fr/sympa
```

Supprimez les références à votre domaine dans sympy.conf (sauf domain et wvsympa_url) :

```
emacs /etc/sympy/sympy.conf
```

Redémarrez Sympa et Apache

```
service sympy restart && service apache2 restart
```

Allez sur l'interface web. Que constatez-vous ?

Déplacement des listes

Toutes les listes doivent être déplacées dans un répertoire list_data du robot.

```
mkdir /usr/local/sympy/list_data/nouveau.domaine.fr
chown sympy:sympy /usr/local/sympy/list_data/nouveau.domaine.fr
mv /usr/local/sympy/list_data/*
   /usr/local/sympy/list_data/nouveau.domaine.fr
```

Aucune autre configuration n'est nécessaire, le mail et le web étant déjà configurés.

Création d'un second hôte virtuel

Configuration web

Comme nous avons désormais deux domaines sur le serveur, il est nécessaire de configurer Apache pour qu'il positionne correctement le nom du serveur lors de la transmission de la requête à Sympa.

Créez le fichier de configuration web du nouveau domaine:

```
emacs /etc/apache2/sites-available/a-nouveau-domaine.fr:80.conf
```

Copiez-y ce contenu. Vous constatez que la configuration sur le port 80 n'a pour seule fonction que de rediriger vers le port 443. :

```
<VirtualHost *:80>
    ServerAdmin listmaster@nouveau.domaine.fr
    ServerName a-nouveau.domaine.fr:80
    ServerSignature off

    ErrorLog ${APACHE_LOG_DIR}/error_a-nouveau.domaine.fr.log
    CustomLog ${APACHE_LOG_DIR}/access_a-nouveau.domaine.fr.log combined

    # VHost specific config
    DocumentRoot "/var/www/html/"

    ## Redirection systématique en HTTPS
    RedirectMatch ^(.*)$ https://a-nouveau.domaine.fr$1

</VirtualHost>
```

Créez le fichier de configuration en HTTPS du nouveau domaine:

```
emacs /etc/apache2/sites-available/a-nouveau-domaine.fr:443.conf
```

Copiez-y ce contenu. Si vous regardez le fichier `/etc/apache2/sites-available/a-nouveau-domaine.fr:80.conf`, vous constaterez que le contenu est identique à prt pour le nom de domaine. Il y a donc là matière à factorisation. On pourrait regrouper tout ce qui n'est pas spécifique au domaine dans un fichier à part et l'inclure.

```
<VirtualHost *:443>
    DocumentRoot /var/www/html
    ServerAdmin support@renater.fr

    ServerName a-nouveau-domaine.fr
    ServerSignature off

    ErrorLog ${APACHE_LOG_DIR}/error_a-nouveau-domaine.fr.log
    CustomLog ${APACHE_LOG_DIR}/access_a-nouveau-domaine.fr.log combined

    # SSL/TLS Configuration
    SSLEngine on
    SSLCertificateKeyFile /etc/ssl/private/nouveau-domaine.fr.key
    SSLCertificateFile /etc/ssl/certs/nouveau-domaine.fr.crt
    SSLCipherSuite ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES256-SHA256:ECDHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA:DHE-RSA-AES256-SHA:ECDHE-RSA-AES128-SHA:DHE-RSA-AES128-SHA:AES256-GCM-SHA384:AES128-GCM-SHA256:AES256-SHA256:AES128-
```



```

SHA256: AES256-SHA: AES128-SHA
  SSLHonorCipherOrder on
  SSLCompression off
  SSLProtocol all -SSLv2 -SSLv3
  Header always set Strict-Transport-Security max-age=15768000
  SSLUseStapling on
  Alias /check-smtp-service-results /usr/local/check_smtp/results/a-
nouveau-domaine.fr
</VirtualHost>

```

Configuration de SYMPA

Créez les répertoire de configuration et des listes de l'hôte virtuel

```

mkdir /usr/local/sympa/etc/a-nouveau.domaine.fr
chown sympa:sympa /usr/local/sympa/etc/a-nouveau.domaine.fr
mkdir /usr/local/sympa/list_data/a-nouveau.domaine.fr
chown sympa:sympa /usr/local/sympa/list_data/a-nouveau.domaine.fr

```

Remplissez le fichier de configuration

```
emacs /usr/local/sympa/etc/a-nouveau.domaine.fr/robot.conf
```

```

domain a-nouveau.domaine.fr

listmaster      <une.adresse.email>

## Who is able to create lists
## This parameter is a scenario, check sympa documentation about scenarios
if you want to define one
create_list     public_listmaster

title <Un titre pour le service>

http_host a-nouveau.domaine.fr

wwsympa_url https://a-nouveau.domaine.fr/sympa

```

Configuration du MTA

Ce n'est pas la seule manière de configurer Postfix, mais celle-ci ne marche pas trop mal.

```
emacs /etc/postfix/domains.sympa
```

Ajouter le nouveau domaine:

```
a-nouveau.domaine.fr  #nouveau domaine
```

Compilez la map:

```
postmap /etc/postfix/domains.sympa
```

Ajoutez ensuite les alias indispensables à Sympa :

```
emacs /etc/postfix/virtual.sympa
```

Ajoutez les lignes suivantes :

```
sympa-request@a-nouveau.domaine.fr    postmaster@localhost  
sympa-owner@a-nouveau.domaine.fr      postmaster@localhost
```

Et compilez-les :

```
postmap /etc/postfix/virtual.sympa
```

Pareil pour ceux qu'on doit passer à Sympa :

```
emacs /etc/postfix/transport
```

Ajoutez ces lignes :

```
sympa@a-nouveau.domaine.fr            sympa:sympa@a-nouveau.domaine.fr  
listmaster@a-nouveau.domaine.fr        sympa:listmaster@a-nouveau.domaine.fr  
bounce@a-nouveau.domaine.fr            sympabounce:sympa@a-nouveau.domaine.fr  
abuse-feedback-report@a-nouveau.domaine.fr  sympabounce:sympa@a-nouveau.domaine.fr
```

Et pareil, on compile !

```
postmap /etc/postfix/transport
```

Vous pouvez tester la nouvelle config avec cette commande :

```
sendmail -bv sympa@a-nouveau.domaine.fr; tail -f /var/log/mail.log
```

Copie de liste d'un robot à l'autre

Il existe une commande for pratique pour cela :

```
/usr/local/sympa/bin/sympa.pl --rename_list=<nom-liste>@nouveau.domaine.fr -  
-new_listname=<nom-liste> --new_listrobot=a-nouveau.domaine.fr
```

Changement de nom de domaine

Voilà une bonne raison d'employer des hôtes virtuels dans Sympa, même si vos ne gérez qu'un domaine.

Supposons que a-nouveau.domaine.fr soit le nouveau domaine de votre établissement. Vous avez créé l'hôte virtuel correspondant à ce domaine. Maintenant, comme votre ancien domaine était également un hôte virtuel, il ne vous reste qu'à déplacer les données de l'un à l'autre pour que le nouveau domaine héberge les mêmes listes que l'ancien.

Renommage du robot

Les informations de cette section ne sont là qu'à titre d'exemple. Dans le cadre du TP, vous avez déjà créé le second hôte virtuel, il n'y a donc rien à créer.

Cela dit, vous pourriez parfaitement partir d'un unique hôte virtuel correspondant à l'ancien domaine et le renommer. Dans ce cas, vous effectueriez les opérations ci-dessous (en plus des opérations de configuration du web et du MTA détaillés dans la section précédente).

```
service sympa stop
service apache2 stop
cd /usr/local/sympa/static_content/css/
mv nouveau.domaine.fr a-nouveau.domaine.fr
cd /usr/local/sympa/etc
mv nouveau.domaine.fr a-nouveau.domaine.fr
cd a-nouveau.domaine.fr
emacs robot.conf
```

Dans l'éditeur, remplacez toutes les occurrences de nouveau.domaine.fr et remplacez-les par a-nouveau.domaine.fr. Enregistrez et fermez.

```
service sympa start
service apache2 start
```

Modification des alias de listes

Si vous utilisez une configuration exploitant les alias vous devez mettre à jour ces alias.

Dans la cadre de la config du TP, cette modification est inutile. Passez au paragraphe suivant.

Dans le fichier /etc/mail/sympa_aliases, effectuez les opérations suivantes :

- Remplacez toutes les occurrences de "nouveau.domaine.fr" par "a-nouveau.domaine.fr"
- Préfixez tous les alias par "a-nouveau.domaine.fr-"

Reconstruisez les alias:

newaliases

Déplacement des listes

Pour cela, il faut déplacer les répertoires des listes et renommer les archives

Déplacement des listes

```
mv /usr/local/sympa/list_data/nouveau.domaine.fr  
/usr/local/sympa/list_data/a-nouveau.domaine.fr
```

Déplacement des archives

```
cd /usr/local/sympa/arc  
for i in $(ls -l | grep nouveau.domaine.fr | sed 's/@nouveau.domaine.fr//');  
do mv $i@nouveau.domaine.fr $i@a-nouveau.domaine.fr; done
```

Si vous désirez que les anciennes adresses continuent à fonctionner, vous pouvez alimenter un fichier de redirections, comme vu dans la formation sur [les bases de Sympa](#).

Enfin, remplacez le nom de l'ancien hôte virtuel par le nouveau dans la table subscribers. Connectez-vous d'abord à la base de données.

```
mysql -u sympa_user -pJwIHlhr_GJK57hD
```

Effectuez les requêtes suivantes :

```
use sympa  
UPDATE subscriber_table SET robot_subscriber = 'a-nouveau.domaine.fr' WHERE  
robot_subscriber LIKE 'nouveau.domaine.fr';
```

Ceci met à jour les données statiques seulement, c'est-à-dire les abonnements. Il faudrait, si vous faites un renommage en catastrophe, renommer les spools (notamment en base de données). L'idéal est d'attendre que les spools soient vides avant de renommer un hôte virtuel.

Exercices

1. Créer une liste dans laquelle les messages sont modérés pour tout le monde, sauf pour les abonnés d'une autre liste et les abonnés authentifiés par l'interface web.
2. Configurer une liste pour que tous les messages envoyés commencent par une texte d'introduction contenant le nom et le prénom de l'abonné.
3. Déplacer tous les spools de Sympa dans un autre répertoire racine
4. Configurer une liste pour que les propriétaires soient autorisés à créer des sources de données.

From:

<https://www.sympa.org/> - **Sympa mailing list server**

Permanent link:

https://www.sympa.org/doc/formation/sympa_bases



Last update: **2018/11/06 09:42**