DOVECOT SHARED FOLDERS AND LDAP

Document created on 2013-12-12
Nicolas Bondier

[pdf][doc][htm]

Contents

Introduction

This document present the configuration of a dovecot server with LDAP authentication and IMAP folders shared between members.

Prerequisites

We need one LDAP server and two Debian server with Dovecot installed and PAM configured for our LDAP server. The both dovecot servers store their configuration and mailboxes on the Ceph cluster and share the same data.

The LDAP installation is presented here http://switzernet.com/3/public/131007-ldap-gosa-unix/ and the dovecot installation above a Ceph cluster is presented here http://switzernet.com/3/public/130910-ceph-dovecot/.

Configuration of Dovecot authentication

As explained in the precedent document about Dovecot on Ceph cluster, the Dovecot configuration directory '`/etc/dovecot`' is a link to a directory on the shared Rados block device. Both Dovecot servers have the same configuration and will not start if the block device is not mounted.

```
root@ceph-client-1:~# ls -l /etc/dovecot
lrwxrwxrwx 1 root root 46 Oct  1 14:16 /etc/dovecot ->
/mnt/switzernet rbd/131001-dovecot-data/config
root@ceph-client-1:~#
```

Bellow, is our configuration file with comment. Each part will be explained latter.

```
# 2.1.7: /etc/dovecot/dovecot.conf
# OS: Linux 3.8.13-xxxx-grs-ipv6-64-vps x86 64

protocols = imap lmtp
log timestamp = "%Y-%m-%d %H:%M:%S "
mail privileged group = vmail

disable plaintext auth = yes
ssl = yes

ssl cert = < /mnt/switzernet rbd/131001-dovecot-
data/certs/switzernet and gd bundle.crt
ssl key  = < /mnt/switzernet rbd/131001-dovecot-
data/certs/switzernet.com.key

# Location for the mail storage on shared block device. Each user have a
distinguish folder.
mail location = maildir:/mnt/switzernet rbd/131001-dovecot-
data/Maildir/accounts/%u:LAYOUT=fs
mail access groups = vmail

# Configuration for storing on OCFS2.
mail fsync = always
mail nfs storage = yes
mail nfs index = yes
mmap disable = yes

# Auth user name format. This is the format used by pam.
auth username format = %n

namespace {
  type = private
  separator = /
  prefix =
  subscriptions = yes
  inbox = yes

}

namespace {
  type = public
```

```
  separator = /
  prefix = Root/
  location = maildir:/mnt/switzernet rbd/131001-dovecot-
data/Maildir/root/:LAYOUT=fs
  subscriptions = no
  ignore on failure = yes
  list = yes
}

service imap {
  # Tell imap to do post-login lookup using a socket called "imap-
postlogin"
  executable = imap imap-postlogin
}

service imap-postlogin {
  # All post-login scripts are executed via script-login binary
  executable = script-login /mnt/switzernet rbd/131001-dovecot-
data/scripts/link shared mailboxes.sh

  # The script process runs as the user given here (v2.0.14+):
  user = root
  # This UNIX socket listener must use the same name as given to imap
executable
  unix listener imap-postlogin {
  }
}

# Postfix authentication
service auth {
  unix listener /var/spool/postfix/private/auth {
    mode = 0660
    # Assuming the default Postfix user and group
    user = postfix
    group = postfix
  }
}

# Local delivery agent for Postfix
service lmtp {
 unix listener /var/spool/postfix/private/dovecot-lmtp {
    group = postfix
    mode = 0600
    user = postfix
  }
}

# Sqlite database with email adress and username
passdb {
  driver = sql
  args = /etc/dovecot/dovecot-sql.conf.ext
}

# LDAP authentication through pam, as it is configured in our system.
passdb {
  driver = pam
  args = session=yes failure_show_msg=yes dovecot
}
```

```
# Sqlite database with email adress and username
userdb {
  driver = sql
  args = /etc/dovecot/dovecot-sql.conf.ext
}

# LDAP authentication through pam, as it is configured in our system.
userdb {
  driver = passwd
}

auth mechanisms = plain login

protocol imap {
  mail plugins = acl imap acl autocreate
}

plugin {
  # When the mailbox dooesn't exist, this plugin create the basic folders.
  autocreate  = Drafts
  autocreate2 = Sent
  autocreate3 = Junk
  autocreate4 = Trash
  autosubscribe = Drafts
  autosubscribe2 = Sent
  autosubscribe3 = Junk
  autosubscribe4 = Trash
  # With global ACLs:
  acl = vfile

  # some logging
  mail log events = delete undelete expunge copy mailbox delete
mailbox rename flag change save mailbox create

  # Allown authenticated keyword
  acl anyone = allow
}
```

Mailboxes and configuration storage

On our installation, all dovecot files are stored under the same folder on the Rados block device.

The paths are not the same as in the [Dovecot on Ceph cluster] document.

The Rados Block device is mounted on '`/mnt/switzernet_rbd/`'.

The Dovecot folder '`/mnt/switzernet_rbd/131001-dovecot-data`' has been manually created.

We created under this folder:

- A '`config`' directory in that is the copy of the '`/etc/dovecot`' configuration directory. On each of the server, '`/etc/dovecot`' have been replaced with a symbolic link to '`/mnt/switzernet_rbd/131001-dovecot-data/config`'.

- A '`certs`' folder for the SSL certificates.

- A '`Maildir`' folder for the storage.

- A '`scripts`' folder for post-login scripts.

The '`Maildir`' folder has:

- An '`accounts`' directory, for the mailboxes of LDAP user accounts.

- A '`root`' directory, which is the root folder of all shared mailboxes.

PAM Authentication

The '`/etc/pam.d/dovecot`' file has the following lines:

```
root@ceph-client-2:~# cat /etc/pam.d/dovecot
#%PAM-1.0

@include common-auth
@include common-account
@include common-session

root@ceph-client-2:~#
```

Verify or change the content of the included files:

- Account rule in '`/etc/pam.d/common-account`':

```
account   required                    pam_permit.so
```

- Authentication rule in '`/etc/pam.d/common-auth`':

```
auth      required                    pam_permit.so
```

In the '`/etc/dovecot/dovecot.conf`', set the parameters as given. Under '`passdb`', '`session=yes`' is needed for some plug-ins, and '`dovecot`' argument is the file under '`/etc/pam.d/`' where to find the PAM configuration.

```
passdb {
  # LDAP authentication through pam, as  configured in our system.
  driver = pam
  args = session=yes failure_show_msg=yes dovecot
}
```

```
userdb {
  # User database use unix GID and UID
  driver = passwd
}
```

With PAM, the only supported authentication mechanism is plaintext. We must set the 'auth_mechanisms' variable this way:

```
auth_mechanisms = plain login
```

Setting-up SSL/TLS encryption

With the current authentication process, it is required to set-up the SSL/TLS encryption to not send plaintext passwords over the network.

First we had to create a private key, then provide it to the SSL certificate provider (GoDaddy) to get our '.crt' file. We also have to get '.crt' bundle of the certification authority.

Place all the files in the 'certs' folder and create a new '.crt' files with all certificates. Order must be 'your_cert.crt', 'your_provider_cert.crt'.

```
root@ceph-client-1:/mnt/switzernet_rbd/131001-dovecot-data/certs# cat
switzernet.com.crt gd_bundle.crt > switzernet_and_gd_bundle.crt
```

Change the rights of the files. The private key should have 0400 rights and certificates should have 0444 rights. Dovecot will read them under the root privileges, so the owner must be root.

```
root@ceph-client-1:/mnt/switzernet_rbd/131001-dovecot-data/certs# ls -l
total 8
-r--r--r-- 1 root root 3197 Oct 10 14:12 gd bundle.crt
-r--r--r-- 1 root root 5092 Oct 10 15:43 switzernet and gd bundle.crt
-r--r--r-- 1 root root 1895 Oct 10 14:12 switzernet.com.crt
-r-------- 1 root root 1704 Oct 10 14:58 switzernet.com.key
```

Finally, add the following lines to your '/etc/dovecot/dovecot.conf':

```
disable plaintext auth = yes
ssl = yes

ssl cert = < /mnt/switzernet rbd/131001-dovecot-
data/certs/switzernet and gd bundle.crt
ssl key  = < /mnt/switzernet rbd/131001-dovecot-
data/certs/switzernet.com.key
```

And restart Dovecot on the both servers:

```
root@ceph-client-1:/mnt/switzernet rbd/131001-dovecot-data/certs#
/etc/init.d/dovecot restart
Restarting IMAP/POP3 mail server: dovecot.
root@ceph-client-1:/mnt/switzernet_rbd/131001-dovecot-data/certs#
```

Private mailboxes permissions

The 'Maildir' directory has the "execute" (for cd) and "read" (for listing) permissions for everybody. It also have the sticky bit for not permitting to users to delete their own folder.

```
root@ceph-client-1:/mnt/switzernet rbd/131001-dovecot-data# getfacl Maildir
# file: Maildir
# owner: root
# group: root
# flags: --t
user::r-x
group::r-x
other::r-x
```

The 'accounts' directory must have 777 rights for permitting any user to add new mailboxes. It also have the sticky-bit 't' for not letting users to remove the mailbox.

```
root@ceph-client-1:/mnt/switzernet rbd/131001-dovecot-data# getfacl
Maildir/accounts
# file: Maildir/accounts
# owner: root
# group: root
# flags: --t
user::rwx
group::rwx
other::rwx
```

Creation of regular folders

Inside the plugin directive of the '/etc/dovecot/dovecot.conf', place the following lines for auto-creating the basic folders at first login.

```
plugin {
  # When the mailbox doeesn't exist, this plugin create the basic folders.
  autocreate  = Drafts
  autocreate2 = Sent
  autocreate3 = Junk
  autocreate4 = Trash
  autosubscribe = Drafts
  autosubscribe2 = Sent
  autosubscribe3 = Junk
  autosubscribe4 = Trash
}
```

Testing login

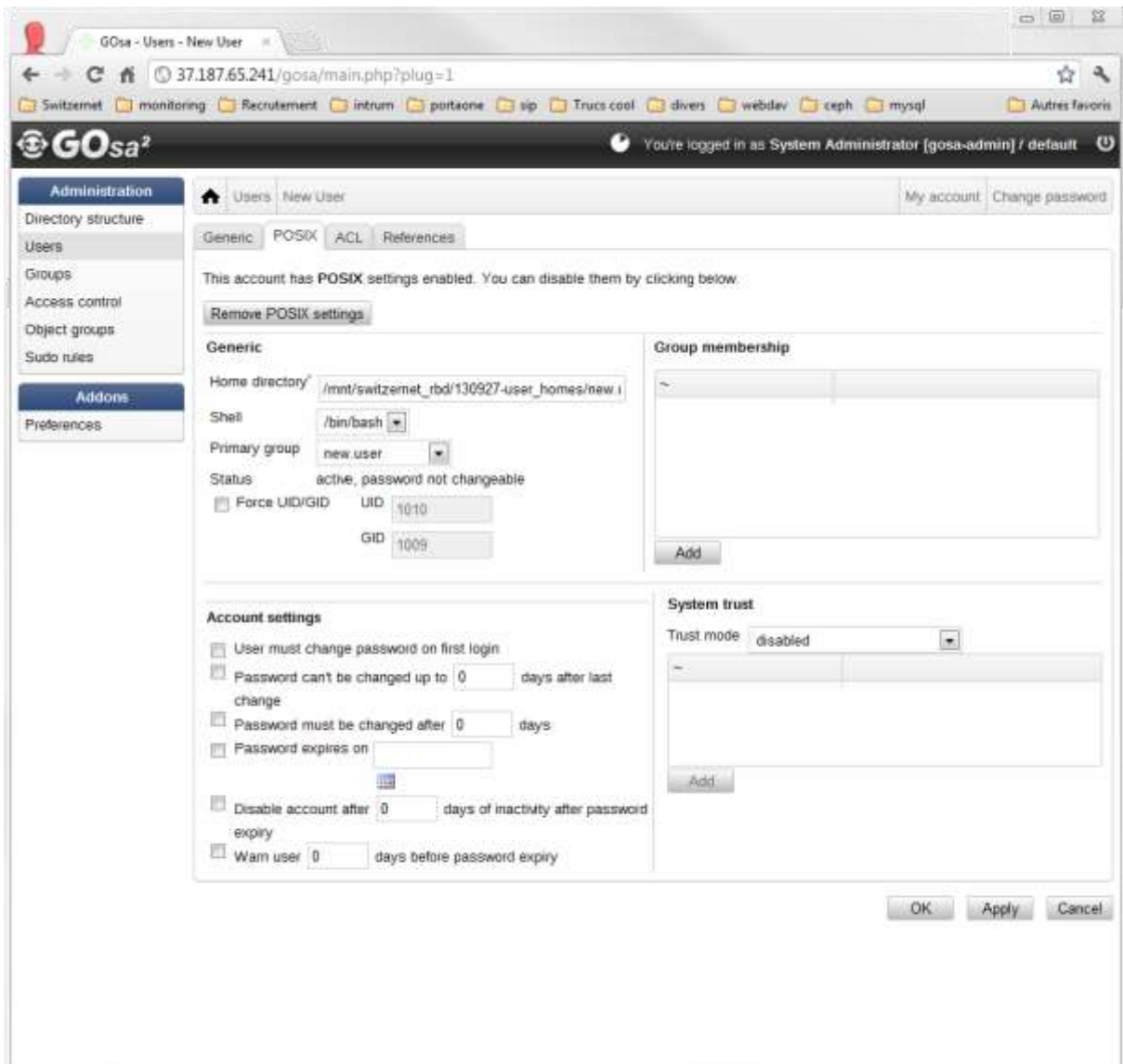We can now test to login with a new user. First we have to create it in the Gosa web interface.

And then configure our IMAP client. Here we use thunderbird:
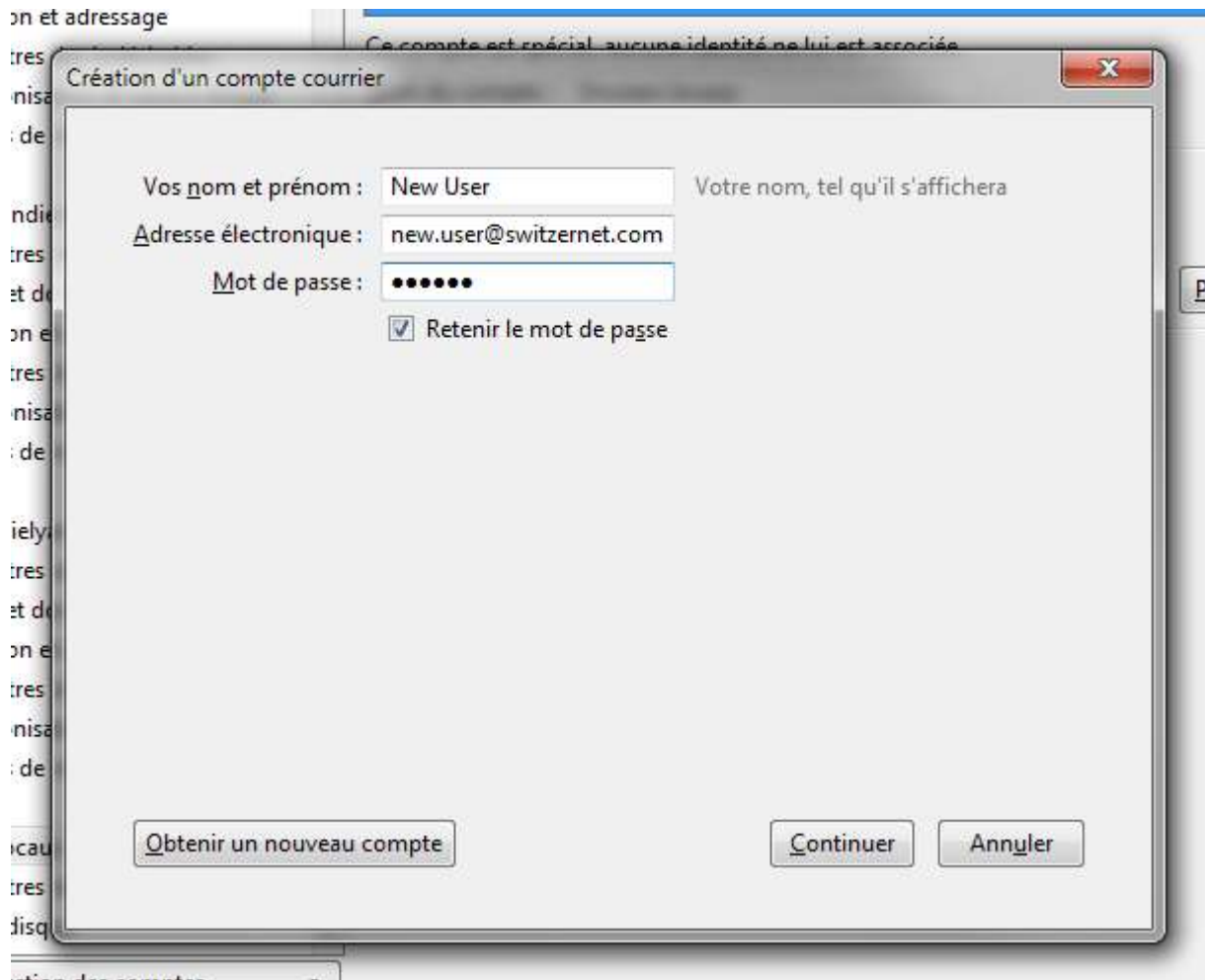
Manual settings:
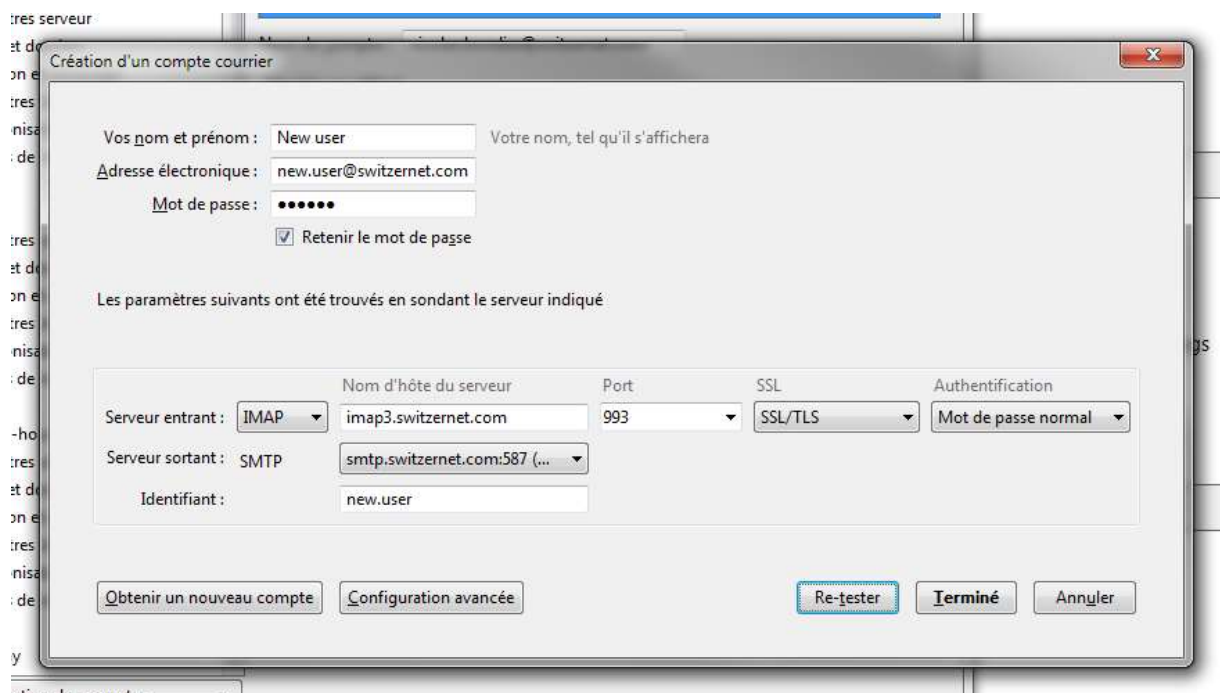
The connection is done. The lock on the letter icon means that the connection is encrypted:

Shared mailboxes configuration

Creation of the dovecot user for shared mailboxes

For each of the dovecot servers, add a new user and group 'vmail'. It will be the primary user used for the reception and storage of emails in the shared mailboxes.

```
root@ceph-client-1:~# groupadd vmail --gid 999
root@ceph-client-1:~# useradd vmail --uid 999 --gid 999
```

Our LDAP users will also be authorized to access shared mailboxes. We will use two types of permissions:

Virtual mailboxes from a SQLite

Creation of a root directory for shared mailboxes

```
root@ceph-client-1:/mnt/switzernet_rbd/131001-dovecot-data/Maildir# ls
accounts
```

Under 'Maildir', create the 'root' directory.

```
mkdir root
```

It must be readable and writable by the 'vmail' user.

```
chown vmail:vmail root
```

The group options must be respected for the subfolders/subdirectories.

```
chmod g+rwxs root
```

Here come the ACLs. First set the largest rights for all users to read, write, and execute to the directory. The '-d' option describes the ALCs to apply to the sub-folders of this directory.

```
setfacl -m mask::rwx root
setfacl -dm mask::rwx root
```

The 'vmail' client have all access to this directory and the permission must be inherit to the new subdirectories.

```
setfacl -m d:g::rwx root
setfacl -dm d:g:vmail:rwx root
```

Do not let users to delete mailboxes in this directory.

```
chmod +t root
```

Verify your ACL:

```
root@ceph-client-1:/mnt/switzernet rbd/131001-dovecot-data/Maildir# getfacl
root
# file: root
# owner: vmail
# group: vmail
# flags: -st
user::rwx
group::rwx
mask::rwx
other::---
default:user::rwx
default:group::rwx
default:mask::rwx
default:other::---
```

Configuration of the shared mailboxes

First, we need to add in our 'dovecot.conf' file two namespaces. One for the private mailboxes and the second one for the shared mailbox:

```
namespace {
  type = private
  separator = /
  prefix =
  subscriptions = yes
  inbox = yes

}

namespace {
  type = public
  separator = /
  prefix = root/
  location = maildir:/mnt/switzernet rbd/131001-dovecot-
data/Maildir/root/:LAYOUT=fs
  subscriptions = no
  ignore on failure = yes
  list = yes
}
```
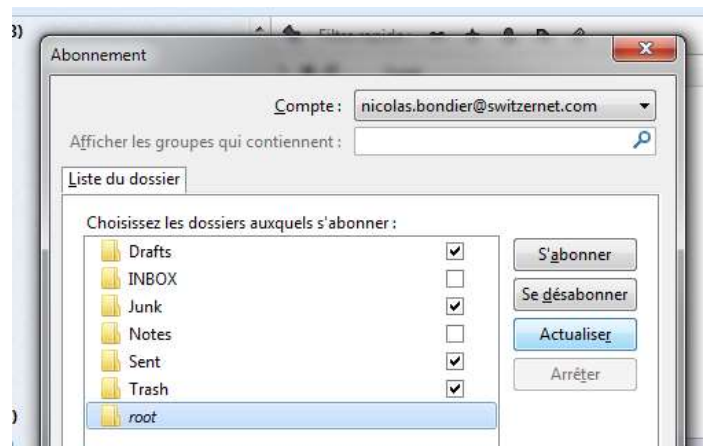
Now restart dovecot on both servers and restart your mail client. At launch, the mail client still have the same mailboxes displayed.

In the subscription management windows, we can see the new namespace 'root'. But it can't be selected because not recognized as a mailbox.

For this, we must create the minimum content of a mailbox.

In the path of the shared namespace, create a 'dovecot-acl' file with the content:

```
root@ceph-client-1:/mnt/switzernet rbd/131001-dovecot-data/Maildir/root#
cat dovecot-acl
anyone keilrwts
```

And create the default directories of the Maildir storage:

```
root@ceph-client-1:/mnt/switzernet rbd/131001-dovecot-data/Maildir/root#
mkdir tmp
root@ceph-client-1:/mnt/switzernet rbd/131001-dovecot-data/Maildir/root#
mkdir cur
root@ceph-client-1:/mnt/switzernet rbd/131001-dovecot-data/Maildir/root#
mkdir new
root@ceph-client-1:/mnt/switzernet rbd/131001-dovecot-data/Maildir/root# ls
-l
total 0
drwxrws---+ 2 root            vmail 3896 Dec 11 11:38 cur
```

```
-rw-rw----+ 1 root            vmail   16 Dec 11 11:37 dovecot-acl
-rw-rw----+ 1 emin.gabrielyan vmail    0 Dec 11 11:26 dovecot-acl-list
drwxrws---+ 2 root            vmail 3896 Dec 11 11:38 new
drwxrws---+ 2 root            vmail 3896 Dec 11 11:38 tmp
```

Restart your thunderbird and see the subscription list:
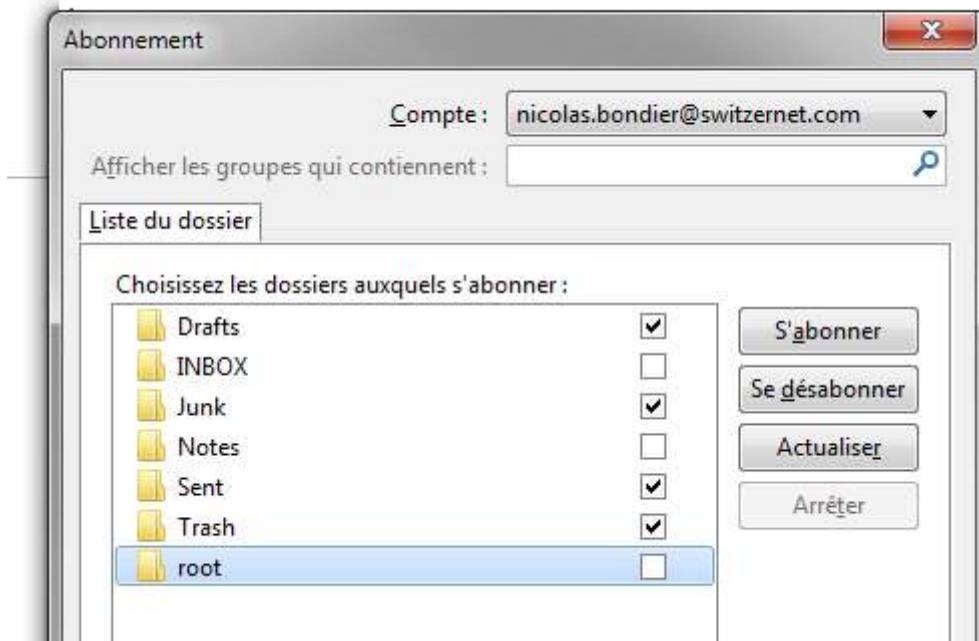
The folder can now be subscribed and you will see it displayed in your thunderbird folders (You may need to restart thunderbird again the first time):

The root folder cannot be deleted or renamed. You can now create new mailboxes inside of this folder.

Example, we create a new folder in the root directory. After subscribed in all accounts to this folder, we can see it in all account folder tree:

nicolas.bondier@switzernet.com
- Courrier entrant
- Brouillons
- Corbeille
- Junk
- root
  - first_shared_folder
- Sent

emin.gabrielyan@switzernet.com
- Courrier entrant
- Brouillons
- Envoyés
- Corbeille
- Junk
- root
  - first_shared_folder
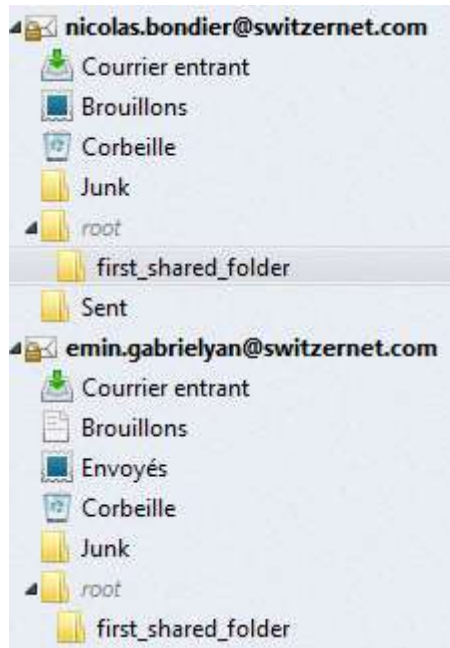
Incoming email configuration

Configuring Postfix with Dovecot LMTP

The SMTP server that will be used for sending and receiving emails is Postfix.

First install Postfix:

```
root@ceph-client-1:~# aptitude install postfix
```

```
root@ceph-client-2:~# aptitude install postfix
```

On our Rados block device, we will store all configuration files. Copy and replace the '/etc/postfix' folder by a symbolic link to the Rados block device (for both servers).

```
root@ceph-client-1:~# cp -r /etc/postfix /mnt/switzernet rbd/131015-postfix
root@ceph-client-1:~# rm -r /etc/postfix
root@ceph-client-1:~# ln -s /mnt/switzernet rbd/131015-postfix/
/etc/postfix
```

```
root@ceph-client-2:~# rm -r /etc/postfix
root@ceph-client-2:~# ln -s /mnt/switzernet rbd/131015-postfix/
/etc/postfix
```

We now have the same configuration files for all postfix servers.

We edited the 'main.cf' with the parameters bellow.

We use the dovecot LMTP socket for authenticating the users and storing the emails. This way, postfix doesn't need to access the LDAP users to authenticate them

This configuration is standard. Notice the SQLite virtual alias maps database that will be explained in the next chapter.

```
queue directory = /var/spool/postfix

myhostname = mail.switzernet.com

alias maps = hash:/etc/aliases

alias database = hash:/etc/aliases

myorigin = /etc/mailname

mydestination = testmail.switzernet.com, switzernet.com,
testmail.intarnet.com, intarnet.com, localhost

relayhost =
```

```
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128

mailbox transport = lmtp:unix:private/dovecot-lmtp

mailbox size limit = 0

recipient delimiter = +

inet interfaces = all

message size limit = 0

virtual alias maps = sqlite:/etc/postfix/sqlite-virtual.cf

transport_maps = hash:/etc/postfix/transport
```

Incoming email on the shared accounts and sub-folders.

We needed to be able to receive emails in some of the shared inboxes and its sub-folder.

For this purpose, we use a SQLite database for virtual alias maps only for the shared folders. The goal of this database is to tell to postfix, when delivering email with LMTP and to a user who is not a LDAP user but a shared directory, for which virtual user and which mailbox the email is, depending on the full email address. Postfix and dovecot use the same table of the database.

Create the database:

```
root@ceph-client-1:~# sqlite3 /mnt/switzernet_rbd/131205-mail-server-
www/131108-imap-admin/virtual.sqlite
```

The SQL for the table creation is bellowed.

```
CREATE TABLE 'virtual Mailboxes' (
  i mailbox INTEGER PRIMARY KEY AUTOINCREMENT,
  address   TEXT NOT NULL,
  path      TEXT NOT NULL
);
CREATE UNIQUE INDEX path_address ON virtual_Mailboxes (address, path);
```

Here is the configuration file for the SQL back-end '/etc/postfix/sqlite-virtual.cf' with the SQL request. (The '@switzernet.com' is only for matching the 'mydestination' variables of the 'main.cf' configuration file):

```
#
# sqlite config file for local(8) aliases(5) lookups
#

# Path to database
```

```
dbpath = /mnt/switzernet rbd/131108-www/131108-imap-acl/virtual.sqlite

# See sqlite table(5) for details.

query = SELECT (i_mailbox||'@switzernet.com') FROM virtual_Mailboxes WHERE
address='%s';
```

Postfix use only 'i_mailbox', which will be considered as a username and send to this user to dovecot during the LMTP delivering.

You may need to run the following command to create the aliases database from '/etc/aliases'.

```
root@ceph-client-1:~# newaliases
```

In Dovecot configuration file, we add the LMTP service:

```
# Local delivery agent for Postfix
service lmtp {
 unix listener /var/spool/postfix/private/dovecot-lmtp {
   group = postfix
   mode = 0600
   user = postfix
  }
}
```

Now we need to tell to postfix how to authenticate users for sending and receiving emails. It will use SASL for authenticating Dovecot's users, who are themselves authenticated with PAM.

First, in Dovecot configuration file, we add the authentication service:

```
service auth {
  unix listener /var/spool/postfix/private/auth {
    mode = 0660
    # Assuming the default Postfix user and group
    user = postfix
    group = postfix
  }
}
```

Then in '/etc/postfix/master.cf', use a dedicated submission port for the authentication (details here: http://wiki2.dovecot.org/HowTo/PostfixAndDovecotSASL#line-41).

```
submission inet n - - - - smtpd
  -o smtpd tls security level=none
  -o smtpd sasl auth enable=yes
  -o smtpd sasl type=dovecot
  -o smtpd sasl path=private/auth
  -o smtpd sasl security options=noanonymous
  -o smtpd sasl local domain=$myhostname
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o smtpd_sender_login_maps=hash:/etc/postfix/virtual
```

```
    -o
smtpd_recipient_restrictions=reject_non_fqdn_recipient,reject_unknown_recip
ient_domain,permit_sasl_authenticated,reject
```

In dovecot, we must add the second authentication with the SQLite database instead of the LDAP database (through PAM).

Another authentication is set before the PAM authentication:

```
# Sqlite database with email adress and username
passdb {
  driver = sql
  args = /etc/dovecot/dovecot-sql.conf.ext
}

# Sqlite database with email adress and username
userdb {
  driver = sql
  args = /etc/dovecot/dovecot-sql.conf.ext
}
```

Dovecot will receive the email send by postfix and match the user part of the address (without the '@domain.com' part) in the same database, to get all the user variables including the mailbox path.

The user part send through LMTP is the 'i_mailbox' field that Postfix get from the 'virtual_alias_maps' database. It will be used by Dovecot to get all other information on a shared account from the SQLite database.

The '/etc/dovecot/dovecot-sql.conf.ext' returns:

- Mailbox path

- UID and GID dovecot must run under for reading and storing the emails.

- The password is random because we do not want to connect to these accounts directly and store them. LMTP doesn't use it.

```
driver = sqlite

connect = /mnt/switzernet rbd/131205-mail-server-www/131108-imap-
admin/virtual.sqlite

default pass scheme = PLAIN

user query = SELECT '' AS home, ('maildir:/mnt/switzernet rbd/131001-
dovecot-data/Maildir/root/'||path) AS mail, '999' AS uid, '999' AS gid FROM
'virtual Mailboxes' WHERE i mailbox = '%n';

password query = SELECT (lower(hex(randomblob(16)))) as password;

iterate_query = SELECT i_mailbox AS username FROM 'virtual_Mailboxes';
```

After the modifications, restart Dovecot and Postfix and both servers.

We insert our folder in the alias map with the address
'first_shared_folder@switzernet.com':

```
root@ceph-client-1:~# sqlite3 /mnt/switzernet rbd/131205-mail-server-
www/131108-imap-admin/virtual.sqlite
sqlite>
sqlite> INSERT INTO 'virtual Mailboxes' (address, path) VALUES
('first shared folder@testmail.switzernet.com','first shared folder');
sqlite>
```

Here is the example in the database:

```
sqlite> .tables
virtual Mailboxes
sqlite> SELECT * FROM virtual Mailboxes;
i mailbox             address                                 path
---------   -------------------------------------------   --------------------
1           first_shared_folder@testmail.switzernet.com   first_shared_folder
```

The steps when receiving an email are the following:

1. To resume, postfix receive an email to
   'first_shared_folder@testmail.switzernet.com'.

2. If address is in the SQLite database, the user is a shared mailbox and 'i_mailbox', will
   replace the username 'first_shared_folder'.

```
sqlite> SELECT (i mailbox||'@switzernet.com') FROM virtual Mailboxes WHERE
address='first shared folder@testmail.switzernet.com';
(i mailbox||'@switze
--------------------
1@switzernet.com
```

3. An authentication request and the mails are sent to Dovecot though LMTP. Dovecot looks in
   the SQLite database to authenticate with the user part of the mail address, which is "1" and
   get where to store the email:

```
sqlite> SELECT '' AS home, ('maildir:/mnt/switzernet rbd/131001-dovecot-
data/Maildir/root/'||path) AS mail, '999' AS uid, '999' AS gid FROM
'virtual Mailboxes' WHERE i mailbox = '1';
home   mail         [...]                        uid   gid
----   ------------[...]--------------------   ---   ----
       maildir:/mnt[...]/first shared folder   999   999
sqlite>
```

Testing sending and receiving emails

We can now test receiving and sending email. As we currently use our primary domain for emails, we have created the testmail subdomain to test sending the email to our new servers. Our DNS configuration is the following. This is the sub-domain used in our precedent examples:

```
;;;; Tests emails

testmail        IN      MX      5 mail-a
testmail        IN      MX      5 mail-b

mail-a          IN      A       5.135.149.193
mail-b          IN      A       5.135.149.195
```
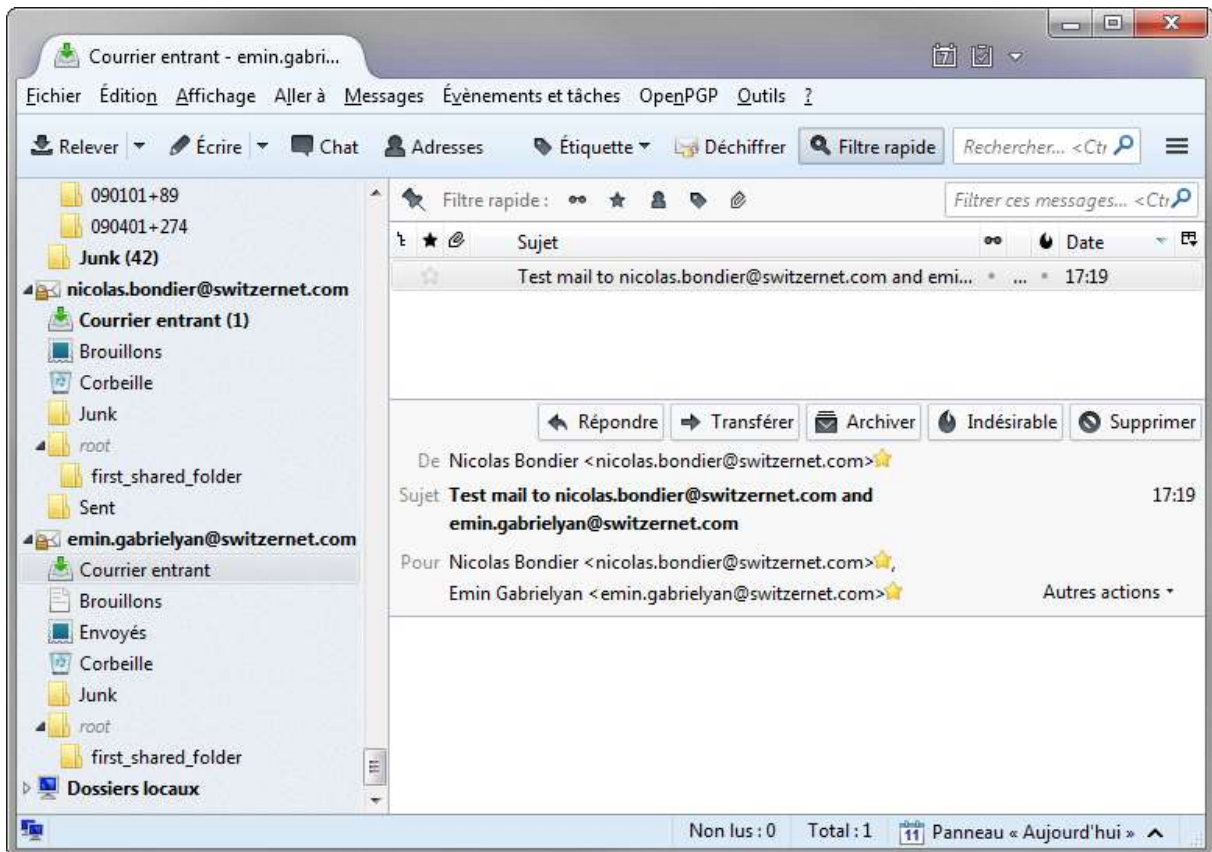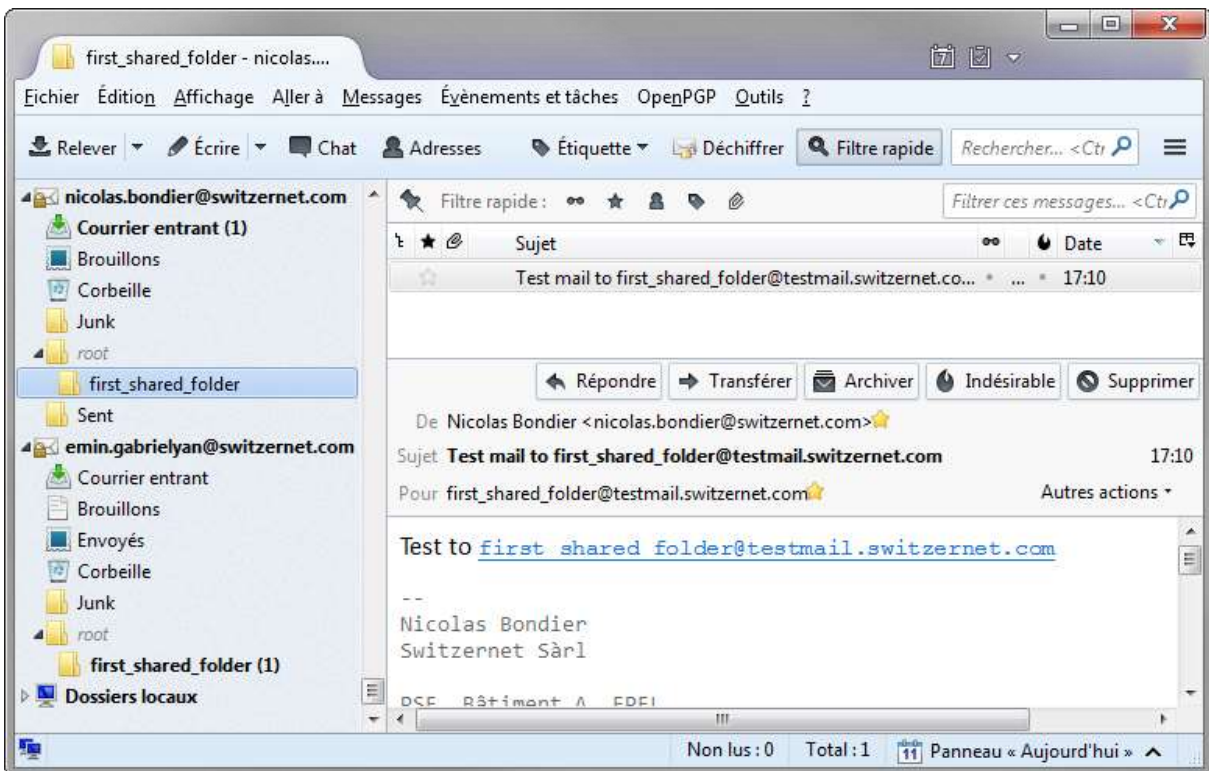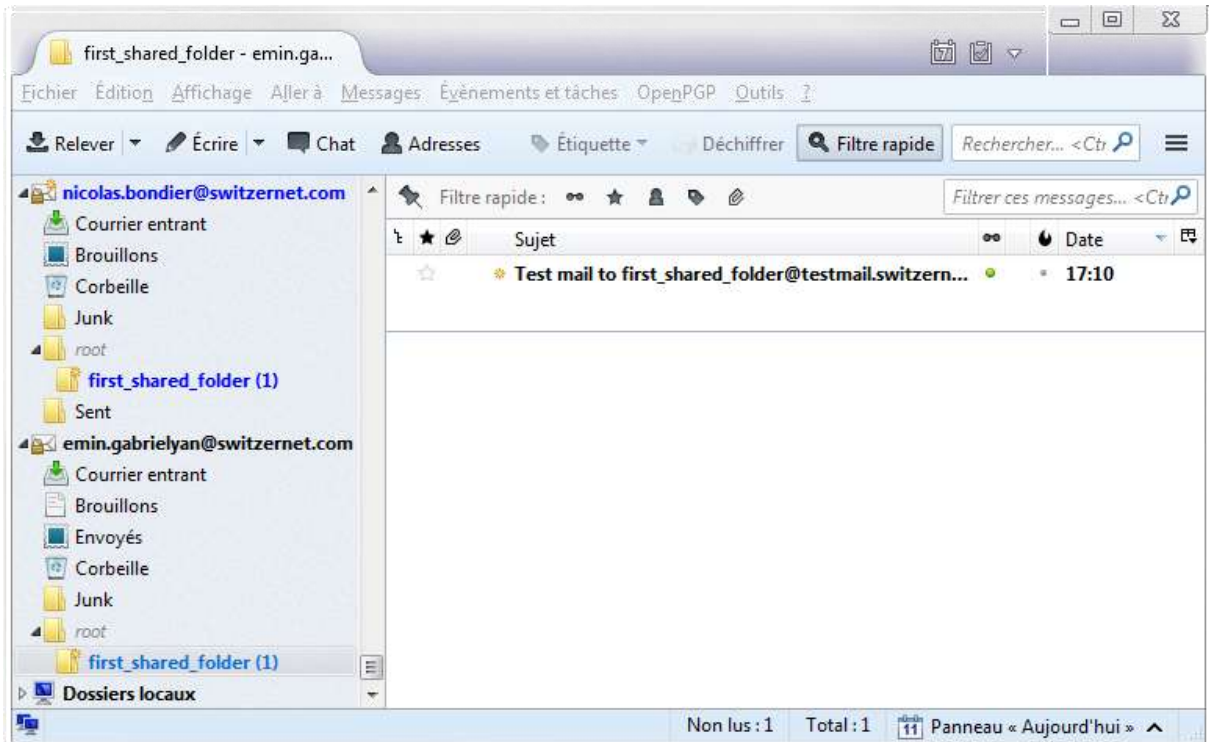
From and to personal inboxes
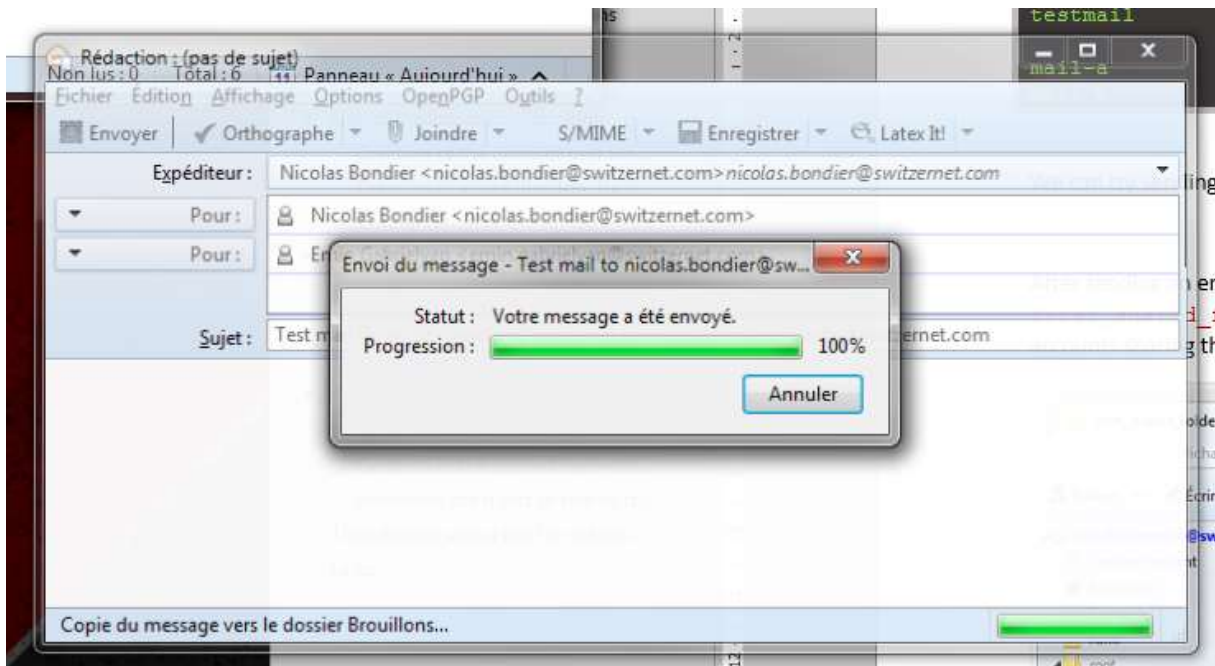
Sending email

Email reception

To shared folders

After sending an email from an external web mail to
'`first_shared_folder@testmail.switzernet.com`', we received it to our mailbox. All
accounts sharing the sub-folder see the email in the public namespace:

ACLs and shared folders management

For managing the shared folder incoming email addresses and the user and group ACLs, we added a web interface.

The link the documentation will be added later.

Thunderbird useful tips for shared mailboxes

Here is an extension for changing the default identities of outgoing emails depending of the current selected folder/sub-folder: https://addons.mozilla.org/fr/thunderbird/addon/folder-account/?src=cb-dl-updated

Files

Dovecot files and folder tree:

```
root@ceph-client-1:/mnt/switzernet rbd# tree 131001-dovecot-data -L 2

131001-dovecot-data
|-- certs
|   |-- gd bundle.crt
|   |-- switzernet and gd bundle.crt
|   |-- switzernet.com.crt
|   `-- switzernet.com.key
|-- config
|   |-- dovecot.conf
|   `-- dovecot-sql.conf.ext
|-- logs
|   `-- mails.log
|-- Maildir
|   |-- accounts
|   |   |-- emin.gabrielyan
|   |   |-- new.user
|   |   `-- nicolas.bondier
|   `-- root
|       |-- cur
|       |-- dovecot-acl
|       |-- dovecot-acl-list
|       |-- dovecot-uidvalidity
|       |-- dovecot-uidvalidity.52a84f4d
|       |-- first shared folder
|       |-- new
|       `-- tmp
`-- scripts
    `-- post-login.sh
```

Files in orange are included in the downloadable archive [131001-dovecot-data.tar.gz].

Postfix files:

```
root@ceph-client-1:/mnt/switzernet rbd# tree -L 2 131015-postfix
131015-postfix
|-- main.cf
|-- master.cf
|-- sqlite-virtual.cf
`-- virtual.sqlite
```

Files in orange are included in the downloadable archive [131015-postfix.tar.gz].

Links

This document: http://switzernet.com/3/public/131007-ldap-gosa-unix/

Debian LDAP PAM: https://wiki.debian.org/fr/LDAP/PAM

Gosa: https://oss.gonicus.de/labs/gosa

OpenLDAP: http://www.openldap.org/

This document is related to the project including:

Ceph cluster: http://switzernet.com/3/public/130925-ceph-cluster/

Dovecot + Ceph: http://switzernet.com/3/public/130910-ceph-dovecot/

*   *   *