

TP 2 Messagerie - R303

GRONDIN Benjamin - BUT 2 Réseau et télécommunication

Table des matières

Introduction.....	2
Objectif de ce TP.....	2
Table d'adressage.....	2
1.1 - Configuration du serveur de messagerie.....	3
Configuration de bind sur le serveur de messagerie.....	3
Redémarrer bind9.....	3
Test de la configuration du fichier db.benjamin.rt.....	3
Configuration client DNS machine physique.....	4
Test résolution de domaine avec nslookup.....	4
Installation de postfix.....	4
Configuration de postfix.....	5
Installation procmail.....	5
Installation courier-base.....	5
Configuration courier-base.....	6
Installation de courier-pop et courier-imap.....	6
Création de user1 et user2.....	7
Création du dossier Maildir et attribution des droits.....	7
Edition du fichier main.cf.....	8
Envoie d'un mail via telnet.....	8
Vérification du message dans la boîte de réception de user2.....	9
Compte rendu.....	10

Introduction

Il existe aujourd'hui une multitude de moyens pour transmettre des messages via le réseau internet. Des applications telles que whatsapp, messenger ou encore telegram permettent de communiquer entre utilisateurs. Cependant, l'une des méthodes de communication les plus utilisées de nos jours reste la messagerie électronique. Ainsi, on estime qu'en 2021 plus de 319 milliards de mails ont été envoyés par jour. Pour que des utilisateurs puissent utiliser une messagerie électronique, il faut mettre en place un serveur dédié qui permettra de recevoir, lire et envoyer des messages. Pour ce faire, le serveur sera composé de plusieurs protocoles tels que SMTP, ou Simple Mail Transfer Protocol, qui sera chargé de l'envoi des messages. Mais encore, du protocole POP, Post Office Protocol, ou du protocole IMAP, Internet Message Access Protocol. Ces deux protocoles seront chargés de récupérer les messages dans la boîte mail. C'est ainsi que fonctionne une messagerie électronique et c'est aujourd'hui le but de ce TP, c'est-à-dire mettre en œuvre un serveur de messagerie fonctionnelle.

Objectif de ce TP

L'objectif de ce TP sera de mettre en place un serveur de messagerie et de réussir à le faire envoyer et recevoir des messages. Pour cela, il nous faudra récupérer la VM du TP précédent portant sur le DNS et y ajouter les applications manquantes et les configurer.

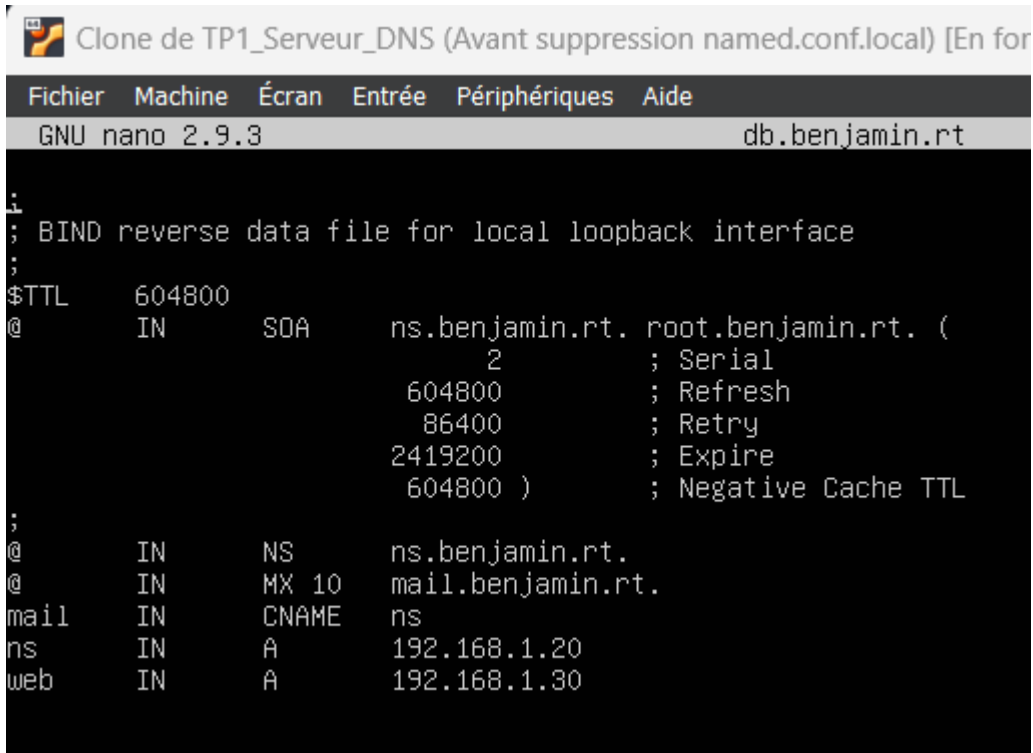
Table d'adressage

Machine	Adresse IP / CIDR	Interface
Serveur de messagerie	192.168.1.20/24	Enp0s3

Vous verrez dans ce TP que le nom de la machine du serveur de messagerie sera dns_server car je n'ai pas modifier le nom de cette dernière.

1.1 - Configuration du serveur de messagerie.

Configuration de bind sur le serveur de messagerie



```
Clone de TP1_Serveur_DNS (Avant suppression named.conf.local) [En for
Fichier  Machine  Écran  Entrée  Périphériques  Aide
GNU nano 2.9.3                                     db.benjamin.rt

;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      ns.benjamin.rt. root.benjamin.rt. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       ns.benjamin.rt.
@         IN      MX 10    mail.benjamin.rt.
mail      IN      CNAME    ns
ns        IN      A        192.168.1.20
web       IN      A        192.168.1.30
```

Sur le serveur DNS que nous avons configuré lors du TP 1, nous ajoutons dans le fichier de zone db.benjamin.rt deux nouvelles informations. Premièrement, la ligne @ IN MX 10 mail.benjamin.rt, qui va attribuer comme serveur racine pour le serveur de messagerie mail.benjamin.rt. Deuxièmement, on indique un Canonical NAME ou alias ns afin que les requêtes envoyées vers mail.benjamin.rt pointent vers ns.benjamin.rt.

Redémarrer bind9

```
benjamin@dns_server:/etc/bind/zones$ sudo systemctl restart bind9
```

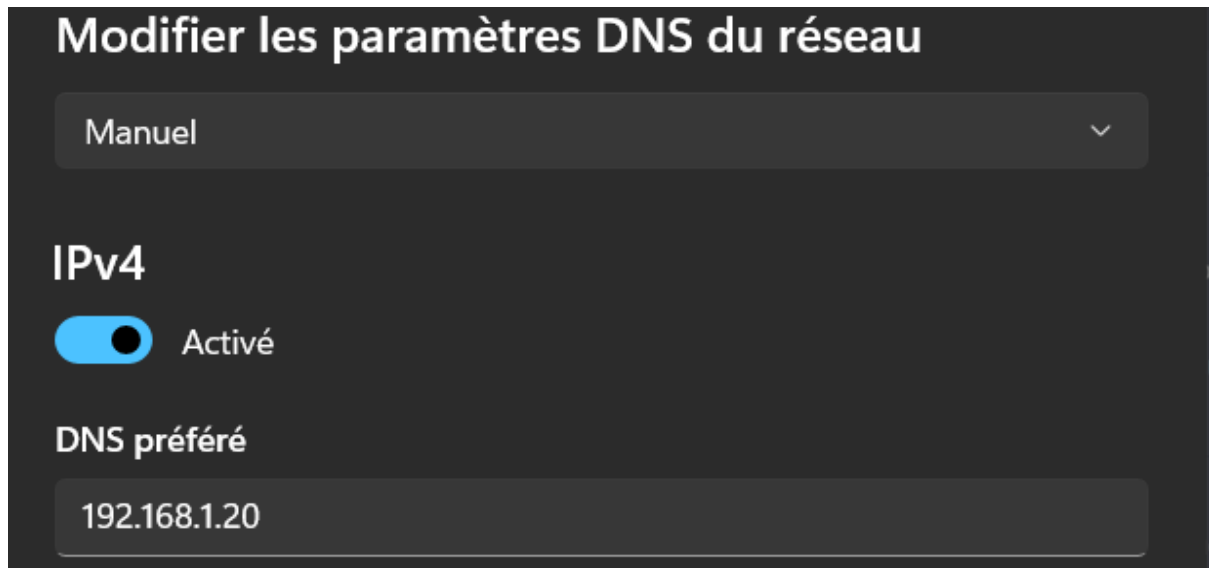
Après avoir édité le fichier db.benjamin.rt on redémarre le service bind9 afin que les nouvelles configurations soient prises en compte.

Test de la configuration du fichier db.benjamin.rt

```
benjamin@dns_server:/etc/bind/zones$ named-checkzone db.benjamin.rt db.1.168.192.in-addr.arpa
zone db.benjamin.rt/IN: loaded serial 2
OK
```

Les nouvelles configurations chargées, on teste à présent la configuration des fichiers de zones, avec la commande named-checkzone, afin de vérifier qu'il n'y a pas d'erreur.

Configuration client DNS machine physique



Modifier les paramètres DNS du réseau

Manuel

IPv4

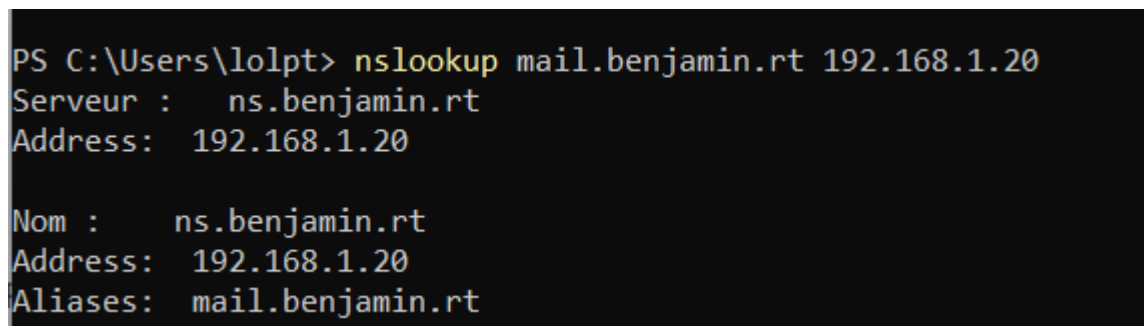
☒ Activé

DNS préféré

192.168.1.20

Afin de pouvoir passer aux étapes suivantes et pouvoir communiquer avec le serveur de messagerie avec l'utilitaire telnet, on doit d'abord ajouter l'adresse du DNS dans les paramètres de notre machine physique. Si nous sautons cette étape, il serait impossible de communiquer avec le serveur mail car la machine physique interrogerait le serveur DNS attribué par défaut par le DHCP et donc un serveur en dehors de notre réseau.

Test résolution de domaine avec nslookup



```
PS C:\Users\lolpt> nslookup mail.benjamin.rt 192.168.1.20
Serveur : ns.benjamin.rt
Address: 192.168.1.20

Nom : ns.benjamin.rt
Address: 192.168.1.20
Aliases: mail.benjamin.rt
```

Avec nslookup on vérifie que la résolution de domaine de mail.benjamin.rt s'effectue correctement. Ici, on peut voir que c'est bien le cas.

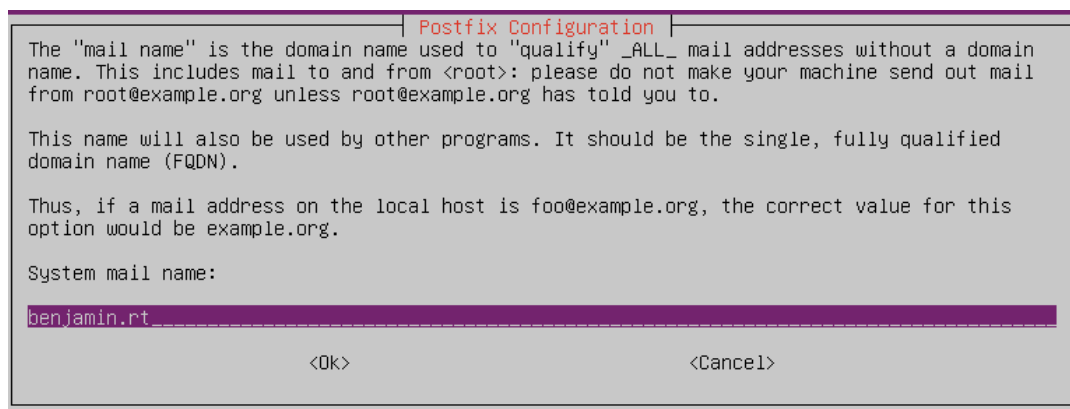
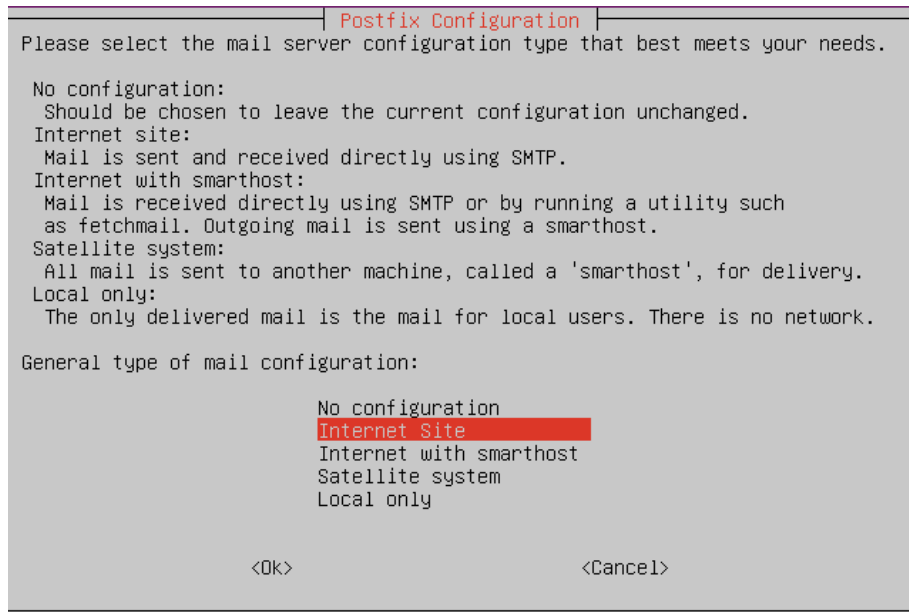
Installation de postfix



```
benjamin@dns_server:/etc/bind/zones$ sudo apt install postfix _
```

A présent, on installe avec la commande ci-dessus le serveur de messagerie électronique postfix .

Configuration de postfix



Arrivé à cette étape, on choisit comme sur la première capture d'écrans Internet site pour que les mail puissent être envoyés et reçus via le protocole SMTP. Ensuite, on choisit le nom de domaine qui sera utilisé, comme montré sur la seconde image. Le domaine utilisé sera benjamin.rt.

Installation procmail

```
benjamin@dns_server:/etc/bind/zones$ sudo apt install procmail
```

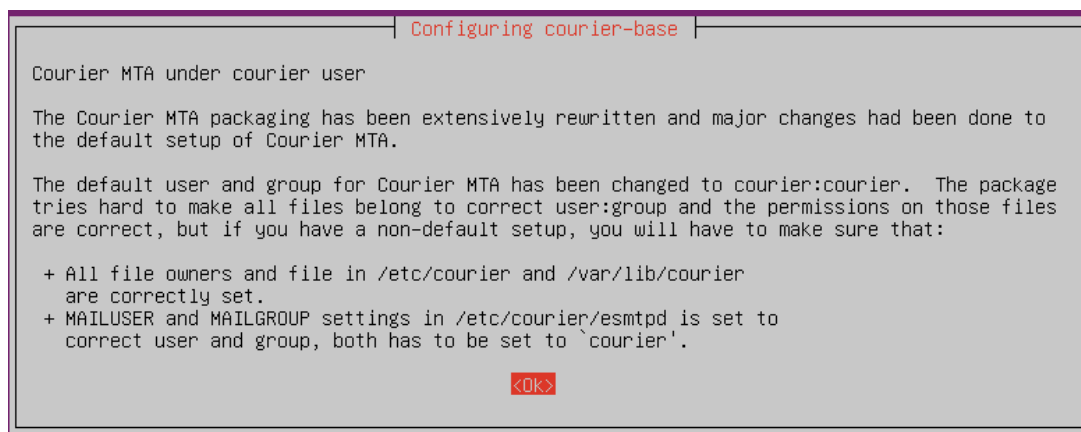
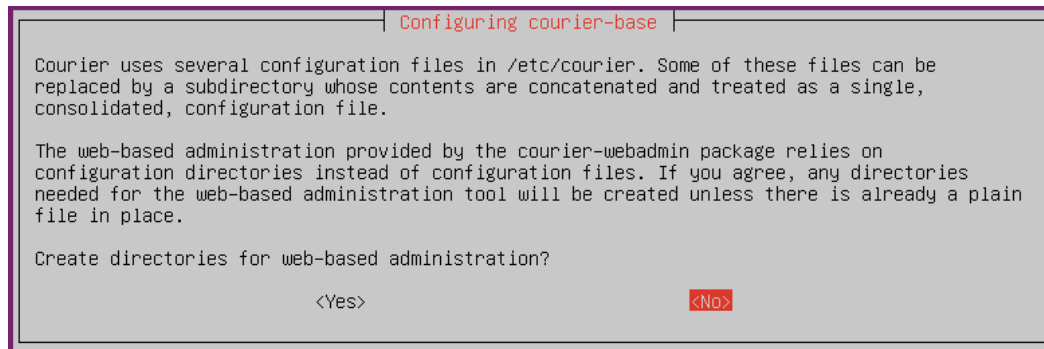
Après l'étape précédente, on installe procmail qui servira à envoyer et filtrer les emails dans les boîtes des destinataire.

Installation courier-base

```
benjamin@dns_server:/etc/bind/zones$ sudo apt install courier-base _
```

Pour pouvoir réceptionner et transférer les messages reçus sur le serveur vers les boîtes mails des utilisateurs il faut à présent installer courier-base avec la commande montrée ci-dessus. Pour effectuer cette tâche, cette utilitaire utilise le protocole IMAP et pop.

Configuration courier-base



Il faut après avoir lancé l'installation accepter les configurations qui sont proposées en cliquant sur OK comme montré sur les trois images précédentes.

Installation de courier-pop et courier-imap

```
benjamin@dns_server:/etc/bind/zones$ sudo apt install courier-pop
```

```
benjamin@dns_server:/etc/bind/zones$ sudo apt install courier-imap_
```

Maintenant que courier-base est installé, il faut ajouter les protocoles pop et imap avec les commandes ci-dessus.

Création de user1 et user2

```
root@dns_server:/etc/postfix# adduser user1
Adding user `user1' ...
Adding new group `user1' (1005) ...
Adding new user `user1' (1001) with group `user1' ...
Creating home directory `/home/user1' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for user1
Enter the new value, or press ENTER for the default
    Full Name []: user1
    Room Number []: 1
    Work Phone []: 0
    Home Phone []: 0
    Other []: 0
```

```
root@dns_server:/etc/postfix# adduser user2
Adding user `user2' ...
Adding new group `user2' (1006) ...
Adding new user `user2' (1002) with group `user2' ...
Creating home directory `/home/user2' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for user2
Enter the new value, or press ENTER for the default
    Full Name []: user2
    Room Number []: 0
    Work Phone []: 0
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

Pour pouvoir tester notre serveur mail il faut à présent créer deux utilisateurs comme montrée ci-dessus. Ils nous serviront pour essayer la transmission et la réception de message.

Création du dossier Maildir et attribution des droits

```
root@dns_server:/home/user1# maildirmake Maildir
```

```
root@dns_server:/home/user1# chown -R user1:user1 Maildir
root@dns_server:/home/user1# ls -l
total 4
drwx----- 5 user1 user1 4096 Oct  6 11:08 Maildir
```

```
root@dns_server:/home/user2# maildirmake Maildir
root@dns_server:/home/user2# chown -R user2:user2 Maildir
root@dns_server:/home/user2# ls -l
total 4
drwx----- 5 user2 user2 4096 Oct  6 11:11 Maildir
```

Maintenant, on crée le dossier Maildir/ dans le répertoire de chacun des utilisateurs. Le répertoire new situé dans Maildir/ sera utilisé pour vérifier que les mails envoyés ont bien été reçus.

Edition du fichier main.cf

```
GNU nano 2.9.3 main.cf
# information on enabling SSL in the smtp client.

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = dns_server
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = $myhostname
mydestination = $myhostname, benjamin.rt, dns_server, localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
#mailbox_command = procmail -a "$EXTENSION"
mydomain = benjamin.rt
default_transport = smtp
home_mailbox = Maildir/
```

On édite le fichier main.cf pour que le serveur mail puisse fonctionner correctement. Ainsi, on y entre les informations tel que la boîte de réception des messages, ici Maildir. Ensuite, le protocole utilisé pour la transmission, et pour finir le nom du domaine depuis lequel on souhaite transmettre les messages.

Envoie d'un mail via telnet

```
PS C:\Users\lolpt> telnet mail.benjamin.rt 25
```

```
Telnet mail.benjamin.rt
220 dns_server ESMTP Postfix (Ubuntu)
HELO benjamin.rt
250 dns_server
MAIL FROM: user1@benjamin.rt
250 2.1.0 Ok
RCPT TO:<user2@benjamin.rt>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Message de test.
.
250 2.0.0 Ok: queued as 30CE614021F
```

Maintenant, depuis la machine physique on ouvre un canal de communication grâce à telnet sur le domaine mail.benjamin.rt via le port 25 qui est utilisé par le protocole SMTP. Puis, on exécute les commandes comme montré sur la deuxième image pour envoyer un message depuis l'utilisateur user1 vers user2. Le message est : Message de test. Après avoir taper le message il faut cliquer sur entrer, puis ajouter un point pour confirmer le message.

Vérification du message dans la boîte de réception de user2

```
user2@dns_server:~/Maildir/new$ ls -l
total 4
-rw----- 1 user2 user2 302 Oct  7 14:11 1728310284.V802Ia002dM602550.dns_server
user2@dns_server:~/Maildir/new$
```

A partir du moment où le message a été envoyé on peut le consulter dans les nouveaux messages reçus par le destinataire dans le dossier "user2/Maildir/new".

```
GNU nano 2.9.3 1728315693.V802Ia002fM13808.dns_server

^Return-Path: <user1@benjamin.rt>
X-Original-To: user2@benjamin.rt
Delivered-To: user2@benjamin.rt
Received: from benjamin.rt (unknown [192.168.1.6])
        by dns_server (Postfix) with SMTP id 30CE614021F
        for <user2@benjamin.rt>; Mon,  7 Oct 2024 15:40:57 +0000 (UTC)

Mez^He^Hssage de test ^H.
```

En faisant nano sur le fichier qui se trouve dans le répertoire new on peut lire le message qui a été transmis par user1 via telnet.

```
PS C:\Users\lolpt> telnet mail.benjamin.rt 110_
```

Sur la machine physique on se connecte via telnet au serveur mail.benjamin.rt et sur le port 110 utilisé par le protocole POP qui permet de voir les messages reçus.

```
Telnet mail.benjamin.rt

+OK Hello there.
USER user2
+OK Password required.
PASS user2
+OK logged in.
LIST
+OK POP3 clients that break here, they violate STD53.
1 310
2 297
```

Sur la capture d'écrans ci-dessus on peut voir qu'après s'être connecté avec user2 et afficher la liste des messages avec LIST, que deux messages ont été reçus. Le message 1 contient le texte envoyé lors d'un premier test et le message 2 le texte envoyé lors de l'étape

Envoie d'un mail via telnet.

```
RETR 2
+OK 297 octets follow.
Return-Path: <user1@benjamin.rt>
X-Original-To: user2@benjamin.rt
Delivered-To: user2@benjamin.rt
Received: from benjamin.rt (unknown [192.168.1.6])
        by dns_server (Postfix) with SMTP id 30CE614021F
        for <user2@benjamin.rt>; Mon,  7 Oct 2024 15:40:57 +0000 (UTC)

Message de test.
.
```

On peut voir qu'il y a bien dans le mail 2 le message envoyé tout à l'heure.

Compte rendu

L'objectif final de ce TP est de mettre en place un serveur mail et le configurer afin de pouvoir échanger des messages entre plusieurs utilisateurs. Ainsi, après avoir suivi les différentes étapes nous sommes parvenus à l'objectif final. Cependant, ayant utilisé une distribution ubuntu 18.04 pour effectuer ce TP je n'ai pas fait la sixième partie car le utilitaire squirrelmail n'est pas disponible pour cette version de ubuntu.