

1. Explique porque para prestar un servicio de correo se necesitan dos servidores.

R//:

Para prestar un servicio de correo, se necesitan dos servidores porque el correo electrónico es un sistema de comunicación bidireccional que implica tanto el envío (correo saliente) como la recepción (correo entrante) de mensajes. Los dos servidores clave involucrados en la prestación de servicios de correo electrónico son:

1. **Servidor de Correo Saliente (SMTP - Simple Mail Transfer Protocol):** Este servidor se encarga de enviar correos electrónicos desde el remitente hacia el destinatario. Su función principal es aceptar, enrutar y entregar mensajes de correo electrónico saliente. Cuando envías un correo electrónico desde tu cliente de correo, el servidor SMTP es responsable de entregar ese correo al servidor de correo del destinatario.
2. **Servidor de Correo Entrante (POP3 o IMAP - Protocolos de Oficina de Correo):** Este servidor se encarga de recibir y almacenar los correos electrónicos destinados a un usuario específico. Cuando un usuario se conecta a su servidor de correo entrante desde un cliente de correo (como Outlook o Thunderbird), puede recuperar los correos electrónicos almacenados en su buzón. Hay dos protocolos comunes para el acceso al correo entrante: POP3, que descarga correos electrónicos a la máquina del cliente, y IMAP, que permite acceder y administrar correos electrónicos directamente en el servidor.

La razón fundamental para tener dos servidores separados es separar las funciones de envío y recepción de correo electrónico para mejorar la eficiencia, la seguridad y la administración del sistema de correo. Algunas de las razones clave incluyen:

- **Mejor Rendimiento:** Separar el servidor SMTP y el servidor POP3/IMAP permite que cada servidor se especialice en su tarea específica, lo que puede mejorar el rendimiento. El servidor SMTP puede concentrarse en la entrega eficiente de mensajes salientes, mientras que el servidor POP3/IMAP puede manejar las solicitudes de recuperación de mensajes entrantes.
- **Seguridad:** Separar los servidores puede proporcionar una capa adicional de seguridad. El servidor SMTP, que está expuesto al mundo exterior, puede ser más estrictamente controlado y protegido, mientras que el servidor POP3/IMAP, que contiene correos electrónicos sensibles, puede ser configurado para un acceso más restringido.
- **Escalabilidad y Redundancia:** Separar los servidores facilita la escalabilidad y la redundancia. Puedes escalar y equilibrar la carga de manera independiente para cada tipo de servidor y configurar sistemas de respaldo y replicación según sea necesario.

En resumen, la separación de las funciones de envío y recepción de correo electrónico en dos servidores diferentes es una práctica estándar en la administración de sistemas de correo electrónico, lo que permite un funcionamiento más eficiente, seguro y confiable del servicio de correo electrónico.

2. Que relación hay entre el servicio de correo y el servidor de nombres?

R//:

El servicio de correo y el servidor de nombres (DNS, Domain Name System) están estrechamente relacionados en el funcionamiento de Internet y, en particular, en la entrega de correos electrónicos. Aquí se explica la relación entre estos dos componentes clave:

1. **Resolución de Nombres de Dominio (DNS):** El servidor de nombres (DNS) es un sistema distribuido y jerárquico que se utiliza para traducir nombres de dominio legibles por humanos (como "www.ejemplo.com") en direcciones IP numéricas (como "192.168.1.1") que las computadoras utilizan para identificarse entre sí en Internet. El DNS es fundamental para que los usuarios puedan acceder a servicios en línea utilizando nombres de dominio en lugar de recordar direcciones IP numéricas.
2. **Configuración de Registro MX (Mail Exchange):** En el contexto del servicio de correo, se utiliza un tipo especial de registro DNS llamado Registro MX (Mail Exchange). Los registros MX se utilizan para indicar qué servidores de correo son responsables de recibir correos electrónicos para un dominio específico. Cada dominio puede tener uno o varios registros MX, y cada registro MX apunta a un servidor de correo.
3. **Envío y Recepción de Correos Electrónicos:** Cuando un usuario envía un correo electrónico a una dirección de correo, el servidor de correo del remitente consulta el registro MX del dominio de destino en el DNS para averiguar a qué servidor de correo debe entregar el mensaje. Luego, el servidor de correo del remitente establece una conexión con el servidor de correo del destinatario utilizando la dirección IP obtenida del registro MX y entrega el correo electrónico.
4. **Resolución de Nombres para Identificación:** Los servidores de correo también pueden utilizar el DNS para resolver nombres de dominio en direcciones IP durante la comunicación. Por ejemplo, al verificar la autenticidad de un servidor de correo remoto, un servidor de correo puede consultar el DNS para obtener la dirección IP correspondiente al nombre de dominio del servidor.

En resumen, el DNS es esencial para que los servidores de correo electrónico funcionen correctamente en Internet. Facilita la resolución de nombres de dominio de servidores de correo, lo que permite la entrega de correos electrónicos a sus destinatarios adecuados. La relación entre el servicio de correo y el DNS es fundamental para la comunicación efectiva y el enrutamiento de correos electrónicos en la red global.

3. Realice las modificaciones en el archivo de configuración de Sendmail para que se acepten correos de diversas maquinas y configure el dominio. Levante el Sendmail y el dovecot y verifique su funcionamiento. ¿Como lo hace?

R//:

Instalamos sendmail:

Yum install sendmail sendmail-cf

```
prueba2 — root@servidor:~ — ssh - vagrant ssh servidor — 80x24
Preparing      :                               1/1
Installing     : tinycdb-0.78-18.el9.x86_64      1/5
Installing     : procmail-3.22-56.el9.x86_64     2/5
Running scriptlet: cyrus-sasl-2.1.27-21.el9.x86_64 3/5
Installing     : cyrus-sasl-2.1.27-21.el9.x86_64 3/5
Running scriptlet: cyrus-sasl-2.1.27-21.el9.x86_64 3/5
Running scriptlet: sendmail-8.16.1-11.el9.x86_64 4/5
Installing     : sendmail-8.16.1-11.el9.x86_64 4/5
Running scriptlet: sendmail-8.16.1-11.el9.x86_64 4/5
Installing     : sendmail-cf-8.16.1-11.el9.noarch 5/5
Running scriptlet: sendmail-cf-8.16.1-11.el9.noarch 5/5
Verifying      : cyrus-sasl-2.1.27-21.el9.x86_64 1/5
Verifying      : procmail-3.22-56.el9.x86_64     2/5
Verifying      : sendmail-8.16.1-11.el9.x86_64 3/5
Verifying      : sendmail-cf-8.16.1-11.el9.noarch 4/5
Verifying      : tinycdb-0.78-18.el9.x86_64      5/5

Installed:
  cyrus-sasl-2.1.27-21.el9.x86_64      procmail-3.22-56.el9.x86_64
  sendmail-8.16.1-11.el9.x86_64      sendmail-cf-8.16.1-11.el9.noarch
  tinycdb-0.78-18.el9.x86_64

Complete!
[root@servidor ~]#
```

Instalamos telnet server: -----Para hacer las pruebas

Yum install telnet telnet-server

```
prueba2 — root@servidor:~ — ssh - vagrant ssh servidor — 80x24
Install 1 Package

Total download size: 39 k
Installed size: 58 k
Is this ok [y/N]: y
Downloading Packages:
telnet-server-0.17-85.el9.x86_64.rpm                20 kB/s | 39 kB    00:01
-----
Total                                              13 kB/s | 39 kB    00:02
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                               1/1
  Installing     : telnet-server-1:0.17-85.el9.x86_64 1/1
  Running scriptlet: telnet-server-1:0.17-85.el9.x86_64 1/1
  Verifying      : telnet-server-1:0.17-85.el9.x86_64 1/1

Installed:
  telnet-server-1:0.17-85.el9.x86_64

Complete!
[root@servidor ~]#
```

Vamos a los archivos de configuración de sendmail:

cd /etc/mail/

ls

ls sendmail.*

```
[[root@servidor ~]# cd /etc/mail/
[[root@servidor mail]# ls
access          helpfile        Makefile        trusted-users
access.db       local-host-names sendmail.cf     virtusertable
aliasesdb-stamp mailertable     sendmail.mc     virtusertable.db
domaintable    mailertable.db  submit.cf       submit.mc
domaintable.db make
[[root@servidor mail]# ls sendmail.*
sendmail.cf  sendmail.mc
[[root@servidor mail]#
```

Usamos el sendmail.mc y todo lo que modificamos ahí los pasamos al .cf con "m4"

HACEMOS UNA COPIA DE LOS ARCHIVOS POR SI LOS DESCONFIGURAMOS:

cp sendmail.cf old.sendmail.cf

cp sendmail.mc old.sendmail.mc

```
[[root@servidor mail]# ls sendmail.*
sendmail.cf  sendmail.mc
[[root@servidor mail]# cp sendmail.cf old.sendmail.cf
[[root@servidor mail]# cp sendmail.mc old.sendmail.mc
[[root@servidor mail]#
```

Vamos al archive sendmail.mc:

vim /etc/mail/sendmail.mc

En la linea:DAEMON_OPTIONS Borramos la parte (Addr=127.0.0.1,)

```
prueba2 — root@servidor:/etc/mail — ssh - vagrant ssh servidor — 80x24
dnl #
dnl # For using Cyrus-IMAPd as POP3/IMAP server through LMTP delivery uncomment
dnl # the following 2 definitions and activate below in the MAILER section the
dnl # cyrusv2 mailer.
dnl #
dnl define(`confLOCAL_MAILER', `cyrusv2')dnl
dnl define(`CYRUSV2_MAILER_ARGS', `FILE /var/lib/imap/socket/lmtp')dnl
dnl #
dnl # The following causes sendmail to only listen on the IPv4 loopback address
dnl # 127.0.0.1 and not on any other network devices. Remove the loopback
dnl # address restriction to accept email from the internet or intranet.
dnl #
DAEMON_OPTIONS(`Port=smtp, Name=MTA')dnl
dnl #
dnl # The following causes sendmail to additionally listen to port 587 for
dnl # mail from MUAs that authenticate. Roaming users who can't reach their
dnl # preferred sendmail daemon due to port 25 being blocked or redirected find
dnl # this useful.
dnl #
dnl DAEMON_OPTIONS(`Port=submission, Name=MSA, M=Ea')dnl
dnl #
dnl # The following causes sendmail to additionally listen to port 465, but
dnl # starting immediately in TLS mode upon connecting. Port 25 or 587 followed
-- INSERT --                               121.41          68%
```

Ahora configuramos el archivo **LocalHostName** y colocamos el del DNS que ya teníamos configurado:

vim /etc/mail/local-host-names

```
prueba2 — root@servidor:/etc/mail — ssh ◀ vagrant ssh servidor
# local-host-names - include all aliases for your machine here.
vasquez.com
server.vasquez.com
~
~
~
~
```

La ip de la maquina es 192.168.50.3 con mascara 255.255.255.0 de prefijo 24 y la parte de red seria 192.168.50 la cual debemos poner en el acces

vim /etc/mail/Access

```
prueba2 — root@servidor:/etc/mail — ssh ◀ vagrant ssh servidor — 80
# Check the /usr/share/doc/sendmail/README.cf file for a description
# of the format of this file. (search for access_db in that file)
# The /usr/share/doc/sendmail/README.cf is part of the sendmail-doc
# package.
#
# If you want to use AuthInfo with "M:PLAIN LOGIN", make sure to have the
# cyrus-sasl-plain package installed.
#
# By default we allow relaying from localhost...
Connect:localhost.localdomain      RELAY
Connect:localhost                  RELAY
Connect:127.0.0.1                  RELAY
Connect:192.168.50                 RELAY
~
~
~
~
```

Ahora lo convertimos en una base de datos(Lo que agregamos al Access)

makemap hash /etc/mail/access.db < /etc/mail/access

Ahora usamos el procesador de macros para copiar lo que estaba en sendmail.mc para mandarlo a sendmail.cf:

m4 sendmail.mc > sendmail.cf

VERIFICACION DE ARCHIVOS

LocalHostName ----- vim /etc/mail/local-host-names

Archivo Access ----- vim /etc/mail/access

LEVANTAMOS SENDMAIL

service sendmail start

Verificamos:

service sendmail status

```
prueba2 — root@servidor:/etc/mail — ssh - vagrant ssh servidor — 80x24
Redirecting to /bin/systemctl start sendmail.service
[root@servidor mail]# service sendmail status
Redirecting to /bin/systemctl status sendmail.service
● sendmail.service - Sendmail Mail Transport Agent
   Loaded: loaded (/usr/lib/systemd/system/sendmail.service; disabled; preset>
   Active: active (running) since Mon 2023-09-04 02:39:02 UTC; 26s ago
   Process: 4395 ExecStartPre=/etc/mail/make (code=exited, status=0/SUCCESS)
   Process: 4396 ExecStartPre=/etc/mail/make aliases (code=exited, status=0/SU>
   Process: 4400 ExecStart=/usr/sbin/sendmail -bd $SENDMAIL_OPTS $SENDMAIL_OPT>
   Main PID: 4401 (sendmail)
     Tasks: 1 (limit: 11129)
    Memory: 3.5M
       CPU: 168ms
    CGroup: /system.slice/sendmail.service
           └─4401 "sendmail: accepting connections"

Sep 04 02:38:02 servidor systemd[1]: Starting Sendmail Mail Transport Agent...
Sep 04 02:38:02 servidor sendmail[4400]: My unqualified host name (servidor) un>
Sep 04 02:39:02 servidor sendmail[4400]: unable to qualify my own domain name (>
Sep 04 02:39:02 servidor sendmail[4401]: starting daemon (8.16.1): SMTP+queuein>
Sep 04 02:39:02 servidor systemd[1]: sendmail.service: Can't open PID file /run>
Sep 04 02:39:02 servidor systemd[1]: Started Sendmail Mail Transport Agent.
lines 1-19/19 (END)
[root@servidor mail]#
```

Conectamos telnet al Puerto 25:

telnet 127.0.0.1 25

Saludamos al servidor:

HELO localhost.localdomain

Para enviar un correo son MAIL FROM y RCPT TO para decirle de donde hasta donde

```
[root@servidor mail]# telnet 127.0.0.1 25
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 servidor ESMTP Sendmail 8.16.1/8.16.1; Mon, 4 Sep 2023 03:37:21 GMT
HELO localhost.localdomain
250 servidor Hello localhost [127.0.0.1], pleased to meet you
```

USUARIOS LOCALES

ls /home/

```
[root@servidor mail]# ls /home/
jmartin  juan  maria  pedro  vagrant
[root@servidor mail]#
```

Nos conectamos de nuevo:

telnet 127.0.0.1 25

Creamos y enviamos un correo desde jmartin hasta tinsho(Lo cree)

```
[root@servidor dovecot]# useradd tinsho
[root@servidor dovecot]# passwd tinsho
Changing password for user tinsho.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@servidor dovecot]# telnet 127.0.0.1 25
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^J'.
220 servidor ESMTP Sendmail 8.16.1/8.16.1; Mon, 4 Sep 2023 04:25:06 GMT
HELO localhost.localdomain
250 servidor Hello localhost [127.0.0.1], pleased to meet you
MAIL FROM:<jmartin@localhost.localdomain>
250 2.1.0 <jmartin@localhost.localdomain>... Sender ok
RCPT TO:<tinsho@localhost.localdomain>
250 2.1.5 <tinsho@localhost.localdomain>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Este es un mensaje de prueba para la practica de servicios sobre correo electronico
.
250 2.0.0 3844P6LQ005304 Message accepted for delivery
QUIT
221 2.0.0 servidor closing connection
Connection closed by foreign host.
```

DOVECOT

Para ver los correos recibidos

Instalación:

yum install dovecot

```
pruebas2 - root@servidor:/etc/mail - ssh - vagrant ssh servidor - 100x34

Downloading Packages:
(1/5): clucene-core-2.3.3.4-42.20130812.e8e3d20git.el9.x86_64.rpm 244 kB/s | 587 kB 00:02
(2/5): dovecot-2.3.16-10.el9.x86_64.rpm 1.3 MB/s | 4.7 MB 00:03
(3/5): libexttextcat-3.4.5-11.el9.x86_64.rpm 203 kB/s | 245 kB 00:01
(4/5): libstemmer-0-18.585svn.el9.x86_64.rpm 365 kB/s | 83 kB 00:00
(5/5): libicu-67.1-9.el9.x86_64.rpm 2.2 MB/s | 9.6 MB 00:04
-----
Total 2.1 MB/s | 15 MB 00:07
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                : 1/1
  Installing                : libstemmer-0-18.585svn.el9.x86_64 1/5
  Installing                : libexttextcat-3.4.5-11.el9.x86_64 2/5
  Installing                : clucene-core-2.3.3.4-42.20130812.e8e3d20git.el9.x86_64 3/5
  Installing                : libicu-67.1-9.el9.x86_64 4/5
  Running scriptlet: dovecot-1:2.3.16-10.el9.x86_64 5/5
  Installing                : dovecot-1:2.3.16-10.el9.x86_64 5/5
  Running scriptlet: dovecot-1:2.3.16-10.el9.x86_64 5/5
  Verifying                : libicu-67.1-9.el9.x86_64 1/5
  Verifying                : clucene-core-2.3.3.4-42.20130812.e8e3d20git.el9.x86_64 2/5
  Verifying                : dovecot-1:2.3.16-10.el9.x86_64 3/5
  Verifying                : libexttextcat-3.4.5-11.el9.x86_64 4/5
  Verifying                : libstemmer-0-18.585svn.el9.x86_64 5/5

Installed:
  clucene-core-2.3.3.4-42.20130812.e8e3d20git.el9.x86_64 dovecot-1:2.3.16-10.el9.x86_64
  libexttextcat-3.4.5-11.el9.x86_64 libicu-67.1-9.el9.x86_64
  libstemmer-0-18.585svn.el9.x86_64

Complete!
```


Debemos configurar algunos archivos
cd /etc/dovecot/
ls

```
prueba2 — root@servidor:/e
[root@servidor dovecot]# ls
conf.d  dovecot.conf
```

vim dovecot.conf

```
prueba2 — root@servidor:/etc/dovecot — ssh ◀ vagrant ssh servidor — 100
# Default values are shown for each setting, it's not required to uncomment
# those. These are exceptions to this though: No sections (e.g. namespace {})
# or plugin settings are added by default, they're listed only as examples.
# Paths are also just examples with the real defaults being based on configure
# options. The paths listed here are for configure --prefix=/usr
# --sysconfdir=/etc --localstatedir=/var

# Protocols we want to be serving.
#protocols = imap pop3 lmtp submission
```

A esa línea(24) le quitamos el # la descomentamos...En mi caso borre el "submission"

```
prueba2 — root@servidor:/etc/dovecot — ssh ◀ vagrant ssh servidor — 100
# Default values are shown for each setting, it's not required to uncomment
# those. These are exceptions to this though: No sections (e.g. namespace {})
# or plugin settings are added by default, they're listed only as examples.
# Paths are also just examples with the real defaults being based on configure
# options. The paths listed here are for configure --prefix=/usr
# --sysconfdir=/etc --localstatedir=/var

# Protocols we want to be serving.
protocols = imap pop3 lmtp
```

Ahora vamos a:

vim /etc/dovecot/conf.d/10-mail.conf
Descomentamos la línea 25


```
prueba2 — root@servidor:/etc/dovecot — ssh - vagrant ssh servidor — 100x34

##
## Mailbox locations and namespaces
##
# Location for users' mailboxes. The default is empty, which means that Dovecot
# tries to find the mailboxes automatically. This won't work if the user
# doesn't yet have any mail, so you should explicitly tell Dovecot the full
# location.
#
# If you're using mbox, giving a path to the INBOX file (eg. /var/mail/%u)
# isn't enough. You'll also need to tell Dovecot where the other mailboxes are
# kept. This is called the "root mail directory", and it must be the first
# path given in the mail_location setting.
#
# There are a few special variables you can use, eg.:
#
# %u - username
# %n - user part in user@domain, same as %u if there's no domain
# %d - domain part in user@domain, empty if there's no domain
# %h - home directory
#
# See doc/wiki/Variables.txt for full list. Some examples:
#
# mail_location = maildir:~/Maildir
# mail_location = mbox:~/mail:INBOX=/var/mail/%u
# mail_location = mbox:/var/mail/%d/%1n/%n:INDEX=/var/indexes/%d/%1n/%n
#
# <doc/wiki/MailLocation.txt>
#
#mail_location =

# If you need to set multiple mailbox locations or want to change default
# namespace settings, you can do it by defining namespace sections.
-- INSERT --
```

25,1

Top

En el mismo archive descoemntamos la linea 114 y agregamos mail:

```
prueba2 — root@servidor:/etc/dovecot — ssh - vagrant ssh servidor — 100x34

# Mailboxes are visible under "shared/user@domain/"
# %n, %d and %u are expanded to the destination user.
#prefix = shared/%u/

# Mail location for other users' mailboxes. Note that %variables and ~/
# expands to the logged in user's data. %n, %d, %u and %h expand to the
# destination user's data.
#location = maildir:%h/Maildir:INDEX=~/Maildir/shared/%u

# Use the default namespace for saving subscriptions.
#subscriptions = no

# List the shared/ namespace only if there are visible shared mailboxes.
#list = children
#}
# Should shared INBOX be visible as "shared/user" or "shared/user/INBOX"?
#mail_shared_explicit_inbox = no

# System user and group used to access mails. If you use multiple, userdb
# can override these by returning uid or gid fields. You can use either numbers
# or names. <doc/wiki/UserIds.txt>
#mail_uid =
#mail_gid =

# Group to enable temporarily for privileged operations. Currently this is
# used only with INBOX when either its initial creation or dotlocking fails.
# Typically this is set to "mail" to give access to /var/mail.
mail_privileged_group = mail

# Grant access to these supplementary groups for mail processes. Typically
# these are used to set up access to shared mailboxes. Note that it may be
# dangerous to set these if users can create symlinks (e.g. if "mail" group is
# set here, ln -s /var/mail ~/mail/var could allow a user to delete others'
-- INSERT --
```

114,29

22%

Igual con la linea 121, la descomentamos y agregamos mail

```
prueba2 — root@servidor:/etc/dovecot — ssh • vagrant ssh servidor — 100x34
#mail_gid =
# Group to enable temporarily for privileged operations. Currently this is
# used only with INBOX when either its initial creation or dotlocking fails.
# Typically this is set to "mail" to give access to /var/mail.
mail_privileged_group = mail

# Grant access to these supplementary groups for mail processes. Typically
# these are used to set up access to shared mailboxes. Note that it may be
# dangerous to set these if users can create symlinks (e.g. if "mail" group is
# set here, ln -s /var/mail ~/mail/var could allow a user to delete others'
# mailboxes, or ln -s /secret/shared/box ~/mail/mybox would allow reading it).
mail_access_groups = mail

# Allow full filesystem access to clients. There's no access checks other than
# what the operating system does for the active UID/GID. It works with both
# maildir and mbox, allowing you to prefix mailboxes names with eg. /path/
# or ~user/.
#mail_full_filesystem_access = no

# Dictionary for key=value mailbox attributes. This is used for example by
# URLAUTH and METADATA extensions.
#mail_attribute_dict =

# A comment or note that is associated with the server. This value is
# accessible for authenticated users through the IMAP METADATA server
# entry "/shared/comment".
#mail_server_comment = ""

# Indicates a method for contacting the server administrator. According to
# RFC 5464, this value MUST be a URI (e.g., a mailto: or tel: URL), but that
# is currently not enforced. Use for example mailto:admin@example.com. This
# value is accessible for authenticated users through the IMAP METADATA server
-- INSERT --
```

Ahora vamos al:

`vim /etc/dovecot/conf.d/10-auth.conf`

Buscamos la línea 10, la descomentamos y cambiamos el `yes` por un `no`

```
prueba2 — root@servidor:/etc/dovecot — ssh • vagrant ssh servidor
##
## Authentication processes
##

# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that if the remote IP
# matches the local IP (ie. you're connecting from the same computer), the
# connection is considered secure and plaintext authentication is allowed.
# See also ssl=required setting.
disable_plaintext_auth = no
```

Ahora vamos al:

`vim /etc/dovecot/conf.d/10-ssl.conf`

Cambiamos la línea 8 por `yes`

```
prueba2 — root@servidor:/etc/dovecot — ssh • vagrant ssh servidor
##
## SSL settings
##

# SSL/TLS support: yes, no, required. <doc/wiki/SSL.txt>
# disable plain pop3 and imap, allowed are only pop3+TLS, pop3s, imap+TLS and imaps
# plain imap and pop3 are still allowed for local connections
ssl = yes
```

INICIAMOS DOVECOT

```
service dovecot start
service dovecot status
```

```
[root@servidor dovecot]# service dovecot start
Redirecting to /bin/systemctl start dovecot.service
[root@servidor dovecot]# service dovecot status
Redirecting to /bin/systemctl status dovecot.service
● dovecot.service - Dovecot IMAP/POP3 email server
   Loaded: loaded (/usr/lib/systemd/system/dovecot.service; disabled; preset: disabled)
   Active: active (running) since Mon 2023-09-04 04:14:17 UTC; 10s ago
     Docs: man:dovecot(1)
           https://doc.dovecot.org/
  Process: 5267 ExecStartPre=/usr/libexec/dovecot/prestartscript (code=exited, status=0/SUCCESS)
 Main PID: 5274 (dovecot)
   Status: "v2.3.16 (7e2e900c1a) running"
    Tasks: 4 (limit: 11129)
  Memory: 5.0M
    CPU: 102ms
   CGroup: /system.slice/dovecot.service
           └─5274 /usr/sbin/dovecot -F
             └─5275 dovecot/anvil
               └─5276 dovecot/log
                 └─5277 dovecot/config

Sep 04 04:14:17 servidor systemd[1]: Starting Dovecot IMAP/POP3 email server...
Sep 04 04:14:17 servidor dovecot[5274]: master: Dovecot v2.3.16 (7e2e900c1a) starting up for imap,
Sep 04 04:14:17 servidor systemd[1]: Started Dovecot IMAP/POP3 email server.
```

PRUEBA DOVECOT

Nos conectamos con POP:

```
telnet 127.0.0.1 110
```

(Al final usamos RETR 1 para ver el mensaje)

```
[root@servidor dovecot]# telnet 127.0.0.1 110
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
+OK Dovecot ready.
USER tinsho
+OK
PASS Tntmasc4
+OK Logged in.
STAT
+OK 1 445
LIST
+OK 1 messages:
1 445
.
RETR 1
+OK 445 octets
Return-Path: <jmartin@localhost.localdomain>
Received: from localhost.localdomain (localhost [127.0.0.1])
        by servidor (8.16.1/8.16.1) with SMTP id 3844P6LQ005304
        for <tinsho@localhost.localdomain>; Mon, 4 Sep 2023 04:26:38 GMT
Date: Mon, 4 Sep 2023 04:25:06 GMT
From: jmartin@localhost.localdomain
Message-Id: <202309040426.3844P6LQ005304@servidor>

Este es un mensaje de prueba para la practica de servicios sobre correo electronico
.
QUIT
```

Ahora hacemos la prueba con IMAP4:

telnet 127.0.0.1 143

```
prueba2 — root@servidor:/etc/dovecot — ssh · vagrant ssh servidor — 109x34
[root@servidor dovecot]# telnet 127.0.0.1 143
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^'.
* OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+ STARTTLS AUTH=PLAIN] Dovecot ready
.
x LOGIN tinsho Tntmasc4
x OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE SORT SORT=DISPLAY THREAD=REFERENCES THREAD=
REFS THREAD=ORDEREDSUBJECT MULTIAPPEND URL-PARTIAL CATENATE UNSELECT CHILDREN NAMESPACE UIDPLUS LIST-EXTENDED
I18NLEVEL=1 CONDSTORE QRESYNC ESEARCH ESORT SEARCHRES WITHIN CONTEXT=SEARCH LIST-STATUS BINARY MOVE SNIPPET=
FUZZY PREVIEW=FUZZY PREVIEW STATUS=SIZE SAVEDATE LITERAL+ NOTIFY SPECIAL-USE] Logged in
x SELECT inbox
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft *)] Flags permitted.
* 1 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1693801766] UIDs valid
* OK [UIDNEXT 2] Predicted next UID
x OK [READ-WRITE] Select completed (0.002 + 0.000 + 0.002 secs).
x FETCH (flags body [header.fields (subject)])
x BAD Error in IMAP command FETCH: Invalid arguments (0.001 + 0.000 secs).
x FETCH 1 (flags body[header.fields (subjects)])
* 1 FETCH (FLAGS (\Seen) BODY[HEADER.FIELDS (SUBJECTS)] {2}
)
x OK Fetch completed (0.002 + 0.000 + 0.001 secs).
x FETCH 1 (body[text])
* 1 FETCH (BODY[TEXT] {85}
Este es un mensaje de prueba para la practica de servicios sobre correo electronico
)
x OK Fetch completed (0.002 + 0.000 + 0.001 secs).
QUIT
```

5: Crear un Alias para un Usuario y Verificar su Funcionamiento

Abre el archivo /etc/aliases para editar las alias de correo

En este caso, queremos que los correos para "tinsho" se envíen también a "martin"

```
prueba2 — root@servidor:~ — ssh · vagrant ssh servidor — 80x24
newsadmin:      news
usenet:         news
ftpadm:         ftp
ftpadm:         ftp
ftp-admin:      ftp
ftp-admin:      ftp
www:            webmaster
webmaster:      root
noc:            root
security:       root
hostmaster:     root
info:           postmaster
marketing:      postmaster
sales:          postmaster
support:        postmaster
tinsho: martin@server.vasquez.com
equipo: martin@server.vasquez.com, tinsho@server.vasquez.com

# trap decode to catch security attacks
decode:         root

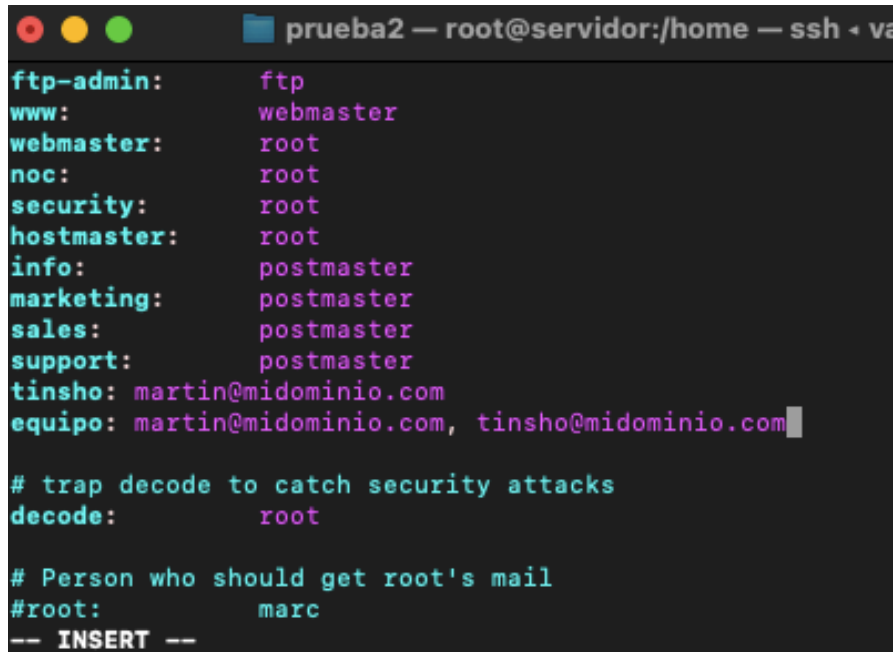
# Person who should get root's mail
```

Aplicamos los cambios con:

newaliases

6: Crear un Alias para un Grupo de Usuarios y Verificar su Funcionamiento

En este caso, queremos que los correos destinados a "equipo" se envíen a "martin" y "tinsho":



```
prueba2 — root@servidor:/home — ssh • va
ftp-admin:      ftp
www:            webmaster
webmaster:      root
noc:            root
security:       root
hostmaster:     root
info:           postmaster
marketing:      postmaster
sales:          postmaster
support:        postmaster
tinsho: martin@midominio.com
equipo: martin@midominio.com, tinsho@midominio.com

# trap decode to catch security attacks
decode:         root

# Person who should get root's mail
#root:         marc
-- INSERT --
```