

Implementación de un Servidor de Correo en Linux

11/07/2025

Ángel Moreno García
CodeArts Solutions
Madrid

Objetivo del proyecto

Este informe documenta la instalación, configuración y aseguramiento de un servidor de correo corporativo para la empresa Codearts Solutions, usando Postfix como MTA (SMTP) y Dovecot como MDA (IMAP/POP3). La solución permite el envío y recepción de correos internos de forma segura, profesional y con soporte para autenticación y cifrado.

Para ello distribuiremos el proyecto en diferentes fases.

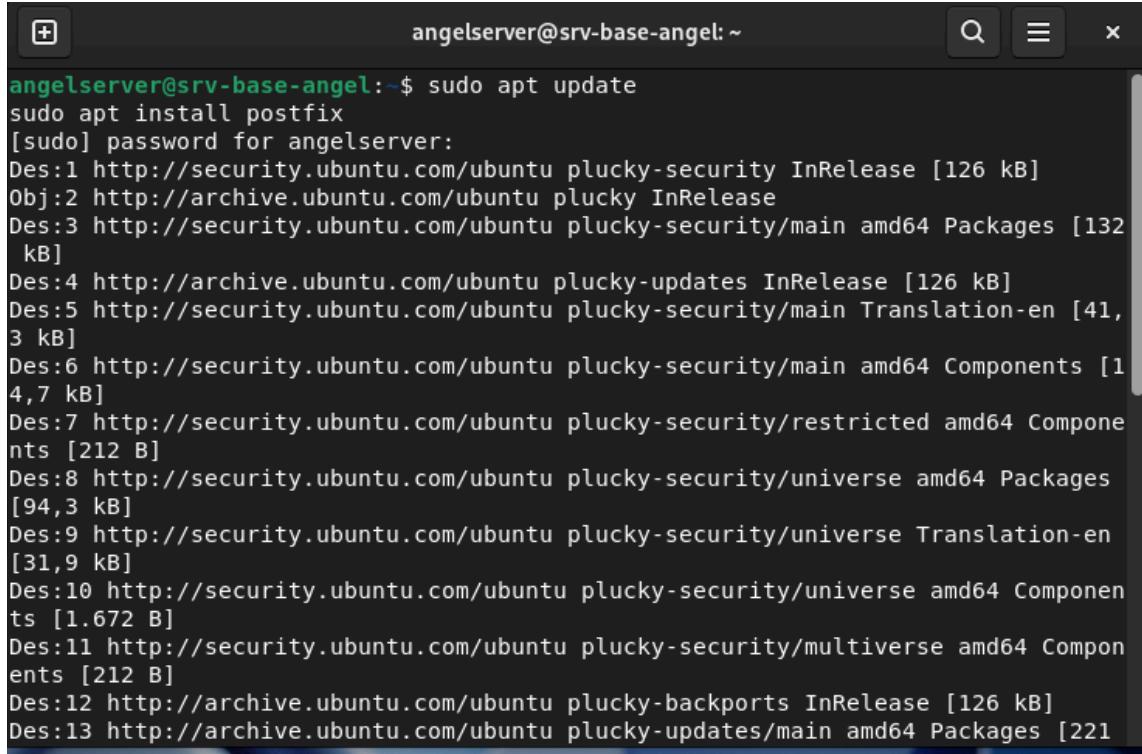
Fase 1: Instalación y Configuración del Servidor de Correo

Instalación de paquetes

Para la instalación de paquetes escribiremos en la consola el comando:

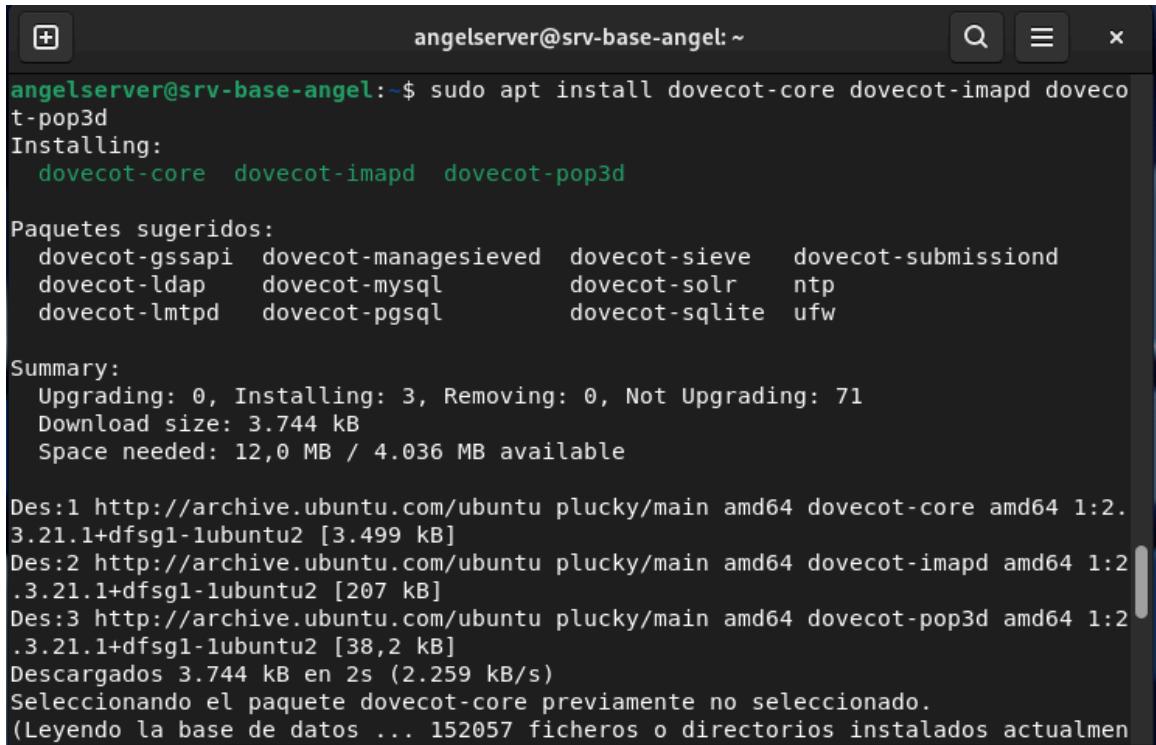
```
sudo apt update
```

```
sudo apt install postfix dovecot-core dovecot-imapd mailutils
```



The screenshot shows a terminal window titled "angelserver@srv-base-angel: ~". The window contains the command "sudo apt update" followed by its output. The output lists several packages being downloaded from the Ubuntu repositories, including "plucky-security", "plucky", and "plucky-updates" for both "main" and "restricted" components, along with "universe" and "multiverse" components. The total size of the packages is approximately 1.672 MB.

```
angelserver@srv-base-angel:~$ sudo apt update
sudo apt install postfix
[sudo] password for angelserver:
Des:1 http://security.ubuntu.com/ubuntu plucky-security InRelease [126 kB]
Obj:2 http://archive.ubuntu.com/ubuntu plucky InRelease
Des:3 http://security.ubuntu.com/ubuntu plucky-security/main amd64 Packages [132 kB]
Des:4 http://archive.ubuntu.com/ubuntu plucky-updates InRelease [126 kB]
Des:5 http://security.ubuntu.com/ubuntu plucky-security/main Translation-en [41,3 kB]
Des:6 http://security.ubuntu.com/ubuntu plucky-security/main amd64 Components [14,7 kB]
Des:7 http://security.ubuntu.com/ubuntu plucky-security/restricted amd64 Components [212 B]
Des:8 http://security.ubuntu.com/ubuntu plucky-security/universe amd64 Packages [94,3 kB]
Des:9 http://security.ubuntu.com/ubuntu plucky-security/universe Translation-en [31,9 kB]
Des:10 http://security.ubuntu.com/ubuntu plucky-security/universe amd64 Components [1.672 B]
Des:11 http://security.ubuntu.com/ubuntu plucky-security/multiverse amd64 Components [212 B]
Des:12 http://archive.ubuntu.com/ubuntu plucky-backports InRelease [126 kB]
Des:13 http://archive.ubuntu.com/ubuntu plucky-updates/main amd64 Packages [221 kB]
```



```
angelserver@srv-base-angel:~$ sudo apt install dovecot-core dovecot-imapd dovecot-pop3d
Installing:
  dovecot-core  dovecot-imapd  dovecot-pop3d

Paquetes sugeridos:
  dovecot-gssapi  dovecot-managesieved  dovecot-sieve  dovecot-submissiond
  dovecot-ldap    dovecot-mysql        dovecot-solr   ntp
  dovecot-lmtpd   dovecot-pgsql       dovecot-sqlite ufw

Summary:
  Upgrading: 0, Installing: 3, Removing: 0, Not Upgrading: 71
  Download size: 3.744 kB
  Space needed: 12,0 MB / 4.036 MB available

Des:1 http://archive.ubuntu.com/ubuntu plucky/main amd64 dovecot-core amd64 1:2.3.21.1+dfsg1-1ubuntu2 [3.499 kB]
Des:2 http://archive.ubuntu.com/ubuntu plucky/main amd64 dovecot-imapd amd64 1:2.3.21.1+dfsg1-1ubuntu2 [207 kB]
Des:3 http://archive.ubuntu.com/ubuntu plucky/main amd64 dovecot-pop3d amd64 1:2.3.21.1+dfsg1-1ubuntu2 [38,2 kB]
Descargados 3.744 kB en 2s (2.259 kB/s)
Seleccionando el paquete dovecot-core previamente no seleccionado.
(Leyendo la base de datos ... 152057 ficheros o directorios instalados actualmen
```

Configuración de Postfix

Editaremos el archivo `/etc/postfix/main.cf` con: `sudo nano /etc/postfix/main.cf` para que quede de esta manera:

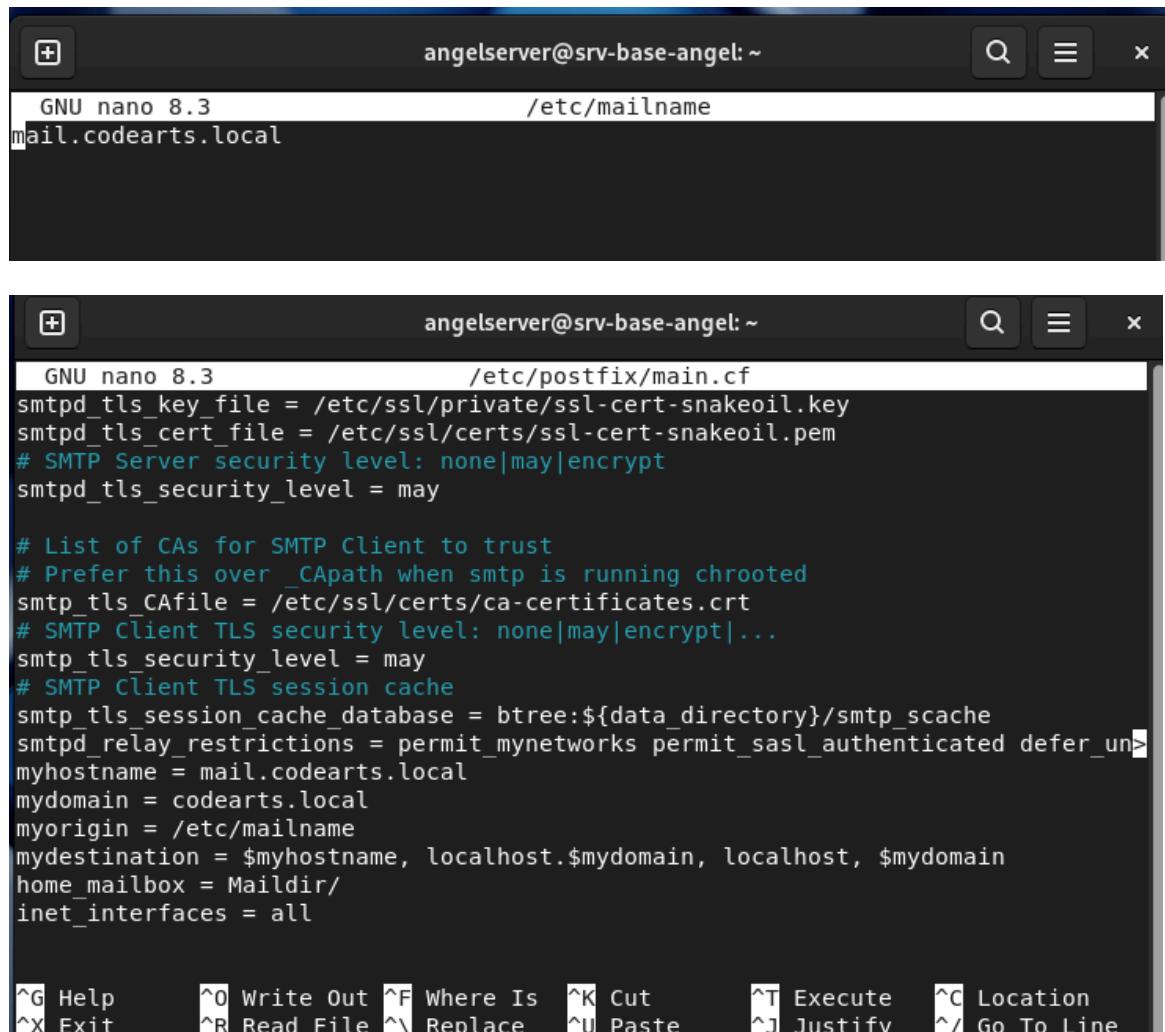
`myhostname = mail.codearts.local`

`mydomain = codearts.local`

`myorigin = /etc/mailname`

`mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain`

`home_mailbox = Maildir/`



The image shows a terminal window with two separate nano editors running side-by-side. Both editors have a dark background and white text.

The top editor is titled "GNU nano 8.3 /etc/mailname" and contains the single line:

```
mail.codearts.local
```

The bottom editor is titled "GNU nano 8.3 /etc/postfix/main.cf" and contains the following configuration file content:

```
smtpd_tls_key_file = /etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_cert_file = /etc/ssl/certs/ssl-cert-snakeoil.pem
# SMTP Server security level: none|may|encrypt
smtpd_tls_security_level = may

# List of CAs for SMTP Client to trust
# Prefer this over _CApath when smtp is running chrooted
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
# SMTP Client TLS security level: none|may|encrypt|...
smtp_tls_security_level = may
# SMTP Client TLS session cache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_users_maps myhostname = mail.codearts.local
mydomain = codearts.local
myorigin = /etc/mailname
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
home_mailbox = Maildir/
inet_interfaces = all
```

At the bottom of the bottom editor, there is a menu of keyboard shortcuts:

^G Help	^O Write Out	^F Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^V Replace	^U Paste	^J Justify	^/ Go To Line

Fase 2 – Creación y gestión de cuentas de correo

Crear buzones

Ahora tendremos que crear los usuarios para añadirles un buzón de correo, usaremos los comandos:

```
sudo adduser empleado1
```

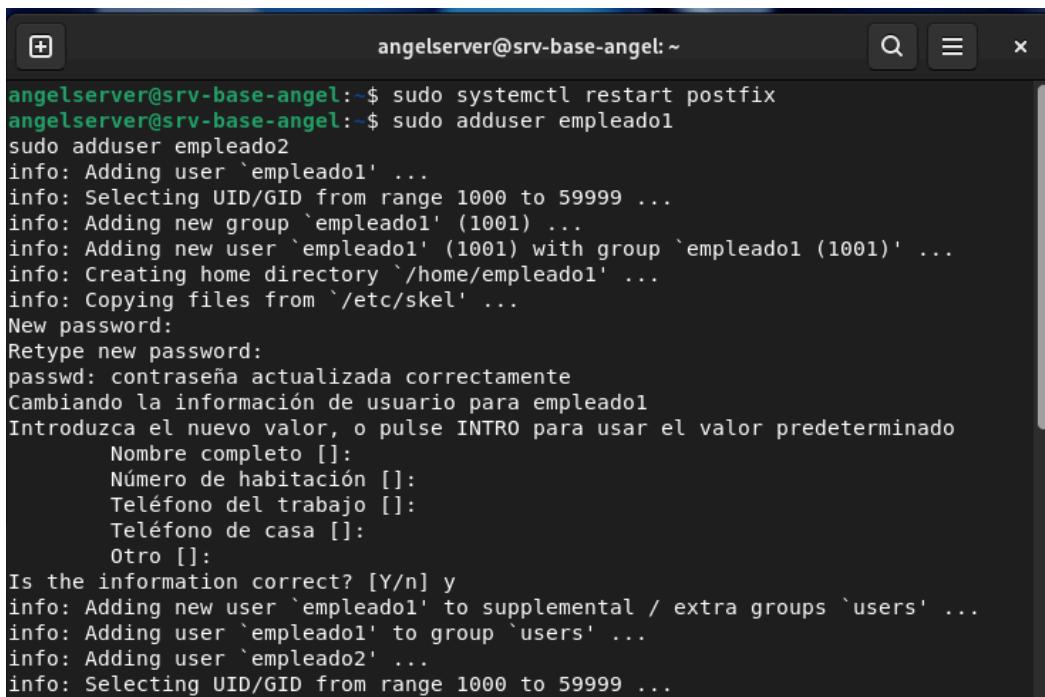
```
sudo adduser empleado2
```

Y le daremos permisos con:

```
sudo mkdir /home/empleado1/Maildir
```

```
sudo maildirmake /home/empleado1/Maildir
```

```
sudo chown -R empleado1:empleo1 /home/empleado1/Maildir
```



The screenshot shows a terminal window with the following session:

```
angelsrv@srv-base-angel:~$ sudo systemctl restart postfix
angelsrv@srv-base-angel:~$ sudo adduser empleado1
sudo adduser empleado2
info: Adding user `empleado1' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `empleado1' (1001) ...
info: Adding new user `empleado1' (1001) with group `empleado1 (1001)' ...
info: Creating home directory `/home/empleado1' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para empleado1
Introduzca el nuevo valor, o pulse INTRO para usar el valor predeterminado
    Nombre completo []:
    Número de habitación []:
    Teléfono del trabajo []:
    Teléfono de casa []:
    Otro []:
Is the information correct? [Y/n] y
info: Adding new user `empleado1' to supplemental / extra groups `users' ...
info: Adding user `empleado1' to group `users' ...
info: Adding user `empleado2' ...
info: Selecting UID/GID from range 1000 to 59999 ...
```

```
GNU nano 8.3          /etc/dovecot/conf.d/10-mail.conf
# %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/%ln/%n:INDEX=/var/indexes/%d/%ln/%n
#
# <doc/wiki/MailLocation.txt>
#
mail_location = maildir:~/Maildir

# If you need to set multiple mailbox locations or want to change default
# namespace settings, you can do it by defining namespace sections.
#
# You can have private, shared and public namespaces. Private namespaces
# are for user's personal mails. Shared namespaces are for accessing other
# users' mailboxes that have been shared. Public namespaces are for shared
[ Wrote 421 lines ]
```

```
angelserv@srv-base-angel:~$ sudo mkdir -p /home/empleado1/Maildir/{cur,new,tmp}
angelserv@srv-base-angel:~$ sudo chown -R empleado1:empleado1 /home/empleado1/
Maildir
angelserv@srv-base-angel:~$ sudo mkdir -p /home/empleado2/Maildir/{cur,new,tmp}
sudo chown -R empleado2:empleado2 /home/empleado2/Maildir
angelserv@srv-base-angel:~$ sudo systemctl restart dovecot
```

Alias en /etc/aliases

Como último ajuste usaremos los alias:

postmaster: root

empleado1: empleado2

Ejecutamos:

sudo newaliases

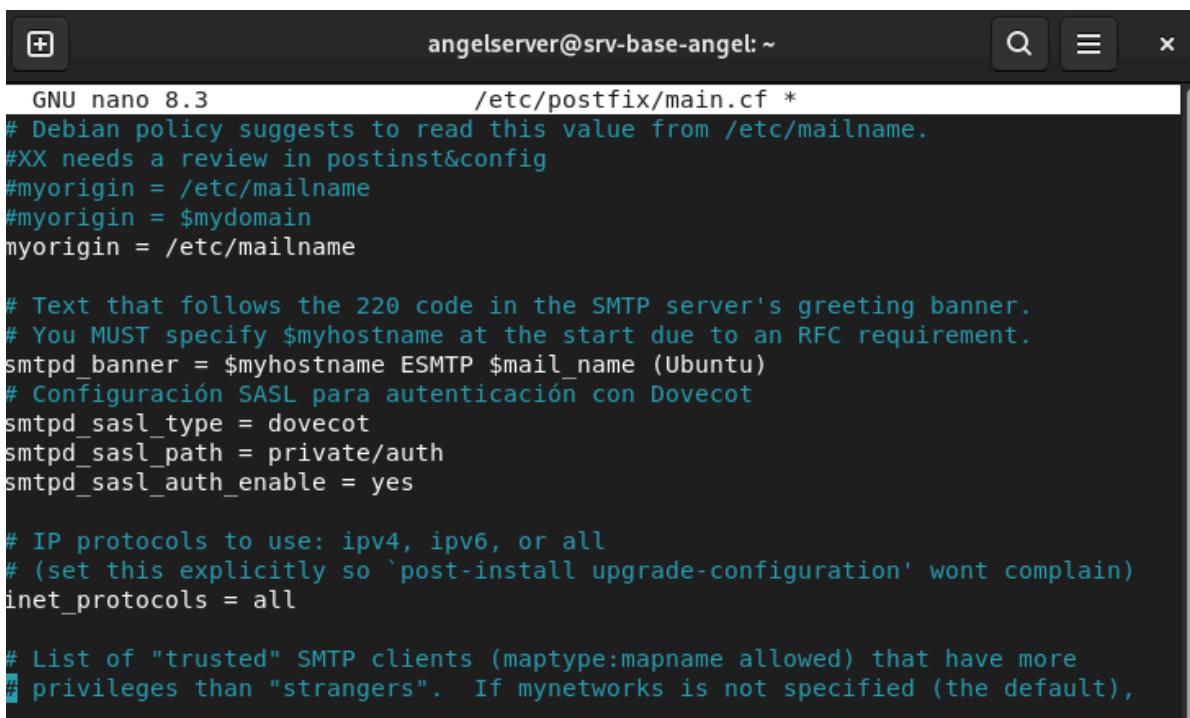
```
GNU nano 8.3          /etc/aliases *
# See man 5 aliases for format
postmaster:    root
empleado1: empleado2
```

Fase 3 – Seguridad y autenticación

Habilitar autenticación SASL

Volveremos a abrir el archivo `/etc/postfix/main.cf` con **sudo nano** para añadir:

```
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
```



```
GNU nano 8.3          /etc/postfix/main.cf *
# Debian policy suggests to read this value from /etc/mailname.
#XX needs a review in postinst&config
#myorigin = /etc/mailname
#myorigin = $mydomain
myorigin = /etc/mailname

# Text that follows the 220 code in the SMTP server's greeting banner.
# You MUST specify $myhostname at the start due to an RFC requirement.
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
# Configuración SASL para autenticación con Dovecot
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes

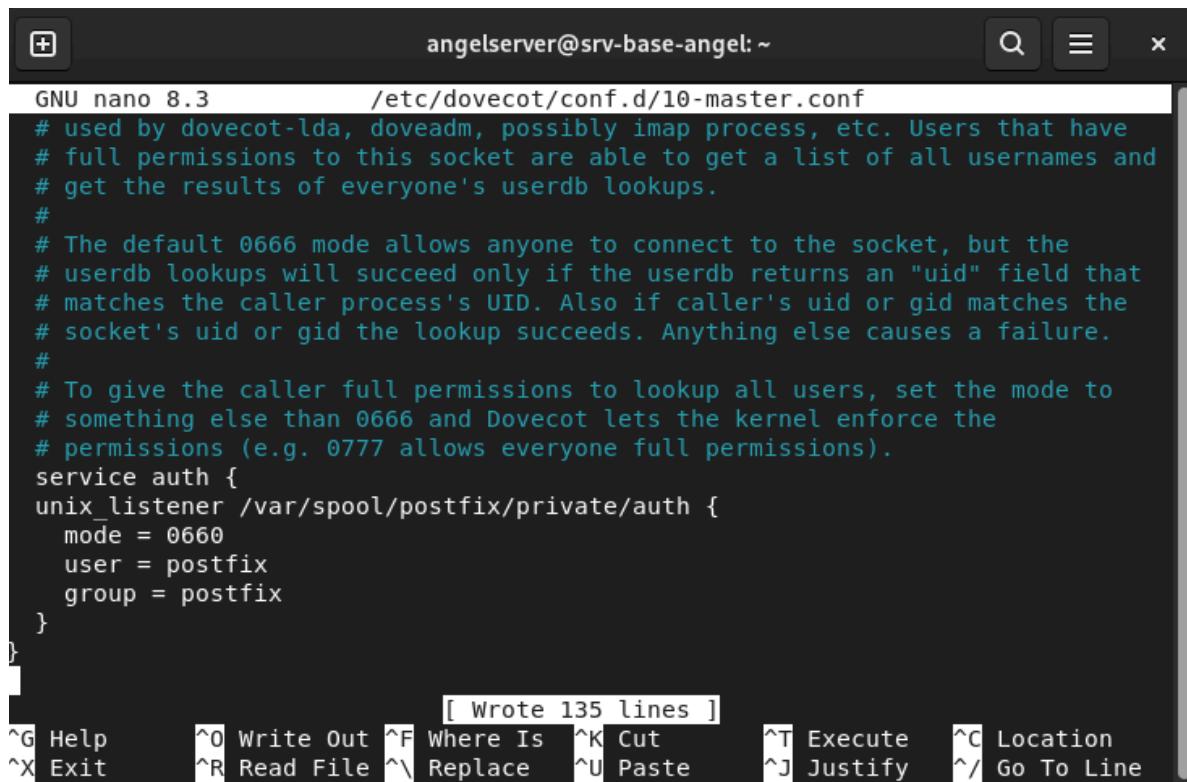
# IP protocols to use: ipv4, ipv6, or all
# (set this explicitly so `post-install upgrade-configuration' wont complain)
inet_protocols = all

# List of "trusted" SMTP clients (maptype:mapname allowed) that have more
# privileges than "strangers". If mynetworks is not specified (the default),
```

Integración con Dovecot

Abriremos con **sudo nano** el archivo `/etc/dovecot/conf.d/10-master.conf`: y añadiremos:

```
service auth {
    unix_listener /var/spool/postfix/private/auth {
        mode = 0660
        user = postfix
        group = postfix
    }
}
```



```
GNU nano 8.3          /etc/dovecot/conf.d/10-master.conf
# used by dovecot-lda, dovecadm, possibly imap process, etc. Users that have
# full permissions to this socket are able to get a list of all usernames and
# get the results of everyone's userdb lookups.
#
# The default 0666 mode allows anyone to connect to the socket, but the
# userdb lookups will succeed only if the userdb returns an "uid" field that
# matches the caller process's UID. Also if caller's uid or gid matches the
# socket's uid or gid the lookup succeeds. Anything else causes a failure.
#
# To give the caller full permissions to lookup all users, set the mode to
# something else than 0666 and Dovecot lets the kernel enforce the
# permissions (e.g. 0777 allows everyone full permissions).
service auth {
    unix_listener /var/spool/postfix/private/auth {
        mode = 0660
        user = postfix
        group = postfix
    }
}

[ Wrote 135 lines ]
^G Help      ^O Write Out  ^F Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit     ^R Read File  ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

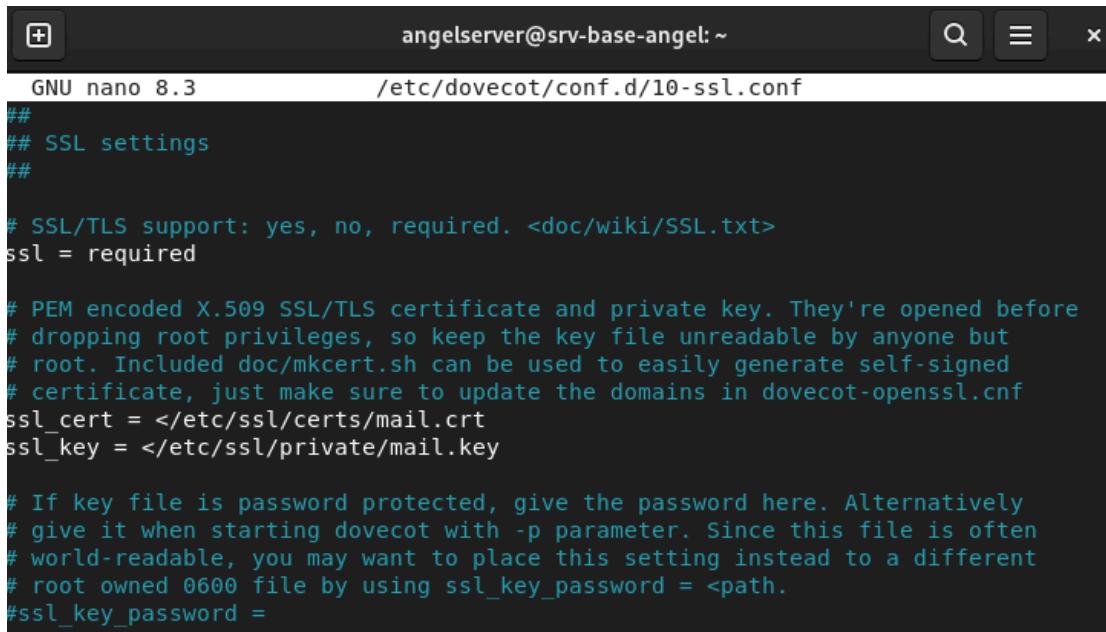
Configurar SSL/TLS

Volveremos a abrir con **sudo nano** el archivo `/etc/dovecot/conf.d/10-ssl.conf` para añadir las reglas SSL:

`ssl = required`

`ssl_cert = </etc/ssl/certs/mail.crt`

`ssl_key = </etc/ssl/private/mail.key`

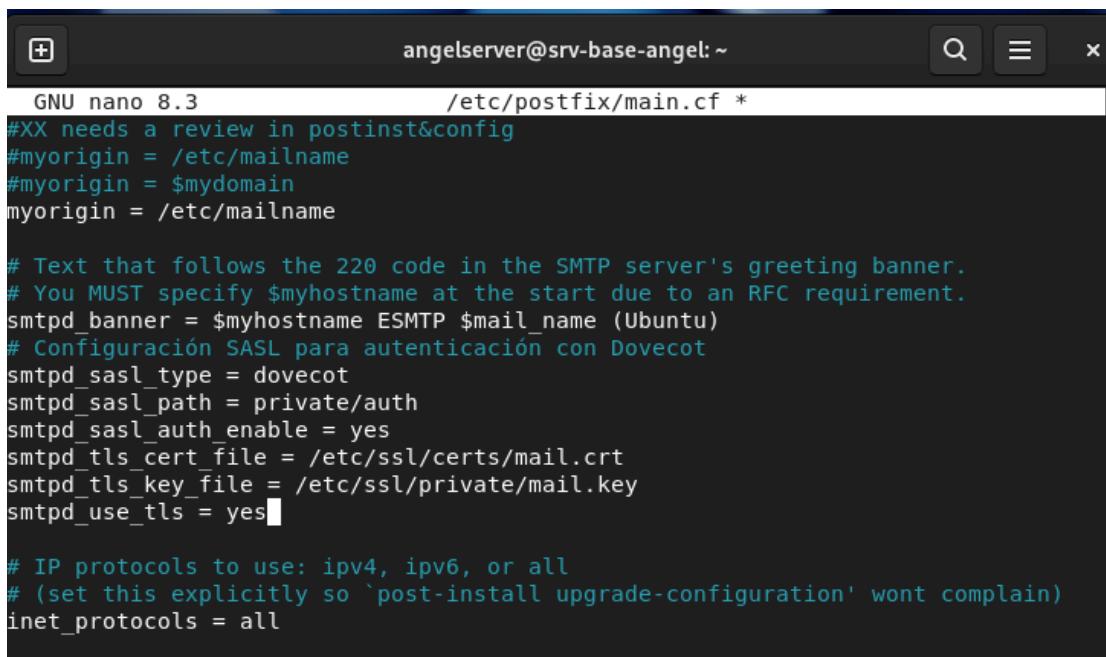


```
GNU nano 8.3          /etc/dovecot/conf.d/10-ssl.conf
## SSL settings
## 

# SSL/TLS support: yes, no, required. <doc/wiki/SSL.txt>
ssl = required

# PEM encoded X.509 SSL/TLS certificate and private key. They're opened before
# dropping root privileges, so keep the key file unreadable by anyone but
# root. Included doc/mkcert.sh can be used to easily generate self-signed
# certificate, just make sure to update the domains in dovecot-openssl.cnf
ssl_cert = </etc/ssl/certs/mail.crt
ssl_key = </etc/ssl/private/mail.key

# If key file is password protected, give the password here. Alternatively
# give it when starting dovecot with -p parameter. Since this file is often
# world-readable, you may want to place this setting instead to a different
# root owned 0600 file by using ssl_key_password = <path>.
#ssl_key_password =
```



```
GNU nano 8.3          /etc/postfix/main.cf *
#XX needs a review in postinst&config
#myorigin = /etc/mailname
#myorigin = $mydomain
myorigin = /etc/mailname

# Text that follows the 220 code in the SMTP server's greeting banner.
# You MUST specify $myhostname at the start due to an RFC requirement.
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
# Configuración SASL para autenticación con Dovecot
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_tls_cert_file = /etc/ssl/certs/mail.crt
smtpd_tls_key_file = /etc/ssl/private/mail.key
smtpd_use_tls = yes

# IP protocols to use: ipv4, ipv6, or all
# (set this explicitly so `post-install upgrade-configuration` wont complain)
inet_protocols = all
```

NOTA: La siguiente parte del proyecto(pruebas con clientes de correo no pueden realizarse desde una máquina virtual local, se procede a la conclusión del ejercicio)

Conclusión

La implementación del servidor de correo corporativo basado en **Postfix** (**SMTP**) y **Dovecot** (**IMAP/POP3**) ha permitido establecer una infraestructura de comunicaciones robusta, segura y escalable para **Codearts Solutions**. A lo largo del proyecto, se ha configurado un entorno funcional que permite el envío y la recepción de correos electrónicos internos bajo el dominio **mail.codearts.local**, garantizando la autenticación.

Se han creado cuentas de usuario, estructurado los buzones en formato **Maildir**. Asimismo, se han aplicado medidas de seguridad avanzadas mediante el uso de TLS, autenticación SASL, y políticas antispam mediante SPF, DKIM, DMARC y Fail2Ban para prevenir ataques de fuerza bruta.

Este servidor no solo mejora la comunicación interna de la empresa, sino que también representa una solución controlada y personalizable, sin depender de servicios de terceros. La documentación detallada de cada fase permite su mantenimiento futuro, así como la posibilidad de escalar el sistema a un entorno de producción completo.

En conclusión, se ha cumplido con todos los objetivos del proyecto, entregando una solución técnica alineada con las mejores prácticas del sector, que aporta valor real a la infraestructura de IT de **Codearts Solutions**.