

# Gestión del acceso a los buzones de correo electrónico

El servidor MTA (*Mail Transfer Agent*) almacena los correos electrónicos en buzones de correo electrónico locales.

Para acceder al contenido de su buzón de correo electrónico, el usuario destinatario ejecuta un programa cliente de mensajería (MUA, *Mail User Agent*). Este programa interroga a un servidor de gestión de correo recibido (MDA, *Mail Delivery Agent*), según el protocolo POP3 (*Post Office Protocol*) o IMAP4 (*Internet Message Access Protocol*).

El servidor MDA, después de haber comprobado la identidad del usuario, responde a las solicitudes del cliente de mensajería para gestionar los mensajes recibidos (lectura, almacenamiento, supresión...).

Los programas MDA que se tienen que conocer en el marco de la certificación LPIC-2 son courier-pop/courier-imap y sobre todo Dovecot.

## 1. El protocolo POP3

El protocolo POP3 usa TCP y el puerto bien conocido 110. El servidor transfiere los mensajes del buzón de correo del usuario al cliente de mensajería. Por defecto, los mensajes leídos se suprimen del buzón de correo electrónico, pero es posible guardar una copia en el servidor.



Este protocolo es simple, pero dispone de funcionalidades reducidas. Tiende a ser reemplazado por el protocolo IMAP4, más complejo pero más potente.

## 2. El protocolo IMAP4

El protocolo IMAP4, más reciente y que dispone de más funcionalidades, usa TCP y el puerto bien conocido 143.

El cliente descarga los encabezados de los mensajes desde el servidor, y permite a continuación que el cliente ejecute acciones en los mensajes: leer, desplazar, suprimir, etc.

Por defecto, los mensajes se conservan en el servidor, pero es posible configurar los clientes IMAP para que sincronicen los mensajes descargados y poder realizar una consulta sin conexión.

El protocolo IMAP permite organizar eficazmente los buzones de correo electrónico creando directorios y subdirectorios. También permite transferir un mensaje completo o una parte de él (encabezado, archivos adjuntos...).

### 3. Servidores courier imap y courier pop

La suite Courier Mail Server incluye los servidores courier-pop y courier-imap. Esta propone el conjunto de los servicios de gestión de correo electrónico, pero sus componentes se pueden instalar de manera separada, en función de las necesidades.

#### a. Formato de almacenamiento de los mensajes

Los servicios courier-pop y courier-imap solo funcionan con mensajes almacenados en los directorios de formato `maildir`. Por lo tanto, hay que configurar el servidor MTA para que utilice ese formato.

#### b. Configuración de los servicios

Los parámetros por defecto de los servicios son, generalmente, suficientes para un funcionamiento estándar. Su archivo de configuración se encuentra en el directorio `/etc/courier`, `pop3d` para el servicio POP e `imapd` para el servicio IMAP.

#### c. Validación de la autenticación

Durante la conexión al servidor, un cliente de correo electrónico presenta su cuenta de

usuario y su contraseña. La biblioteca de autenticación `courier` valida estos datos para autorizar el acceso al servidor.

El comando `authtest` permite comprobar que una cuenta de usuario puede abrir efectivamente una conexión hacia el servidor local (POP o IMAP).

### Sintaxis

```
authtest usuario contraseña
```

### Ejemplo

```
authtest aca micontraseña
```

```
Authentication succeeded.
```

```
Authenticated: aca (system username: aca)
```

```
Home Directory: /home/aca
```

```
Maildir: (none)
```

```
Quota: (none)
```

```
Encrypted Password: $1$zSIbmjnM$244213Mpk.
```

```
Cleartext Password: micontraseña
```

```
Options: (none)
```

## 4. El servidor Dovecot

Dovecot es un servidor de gestión de correo recibido, open source, concebido para asegurar un máximo de rendimiento y un alto nivel de seguridad.

Aunque su implementación de base sea relativamente simple, como presenta un gran número de funcionalidades, sus posibilidades de configuración son múltiples y pueden ser complejas.

Soporta los formatos de buzones de correo electrónico `mbox` y `maildir`.

## a. Configuración

El servidor Dovecot usa el archivo de configuración `dovecot.conf`, que se encuentra normalmente en el directorio `/etc/dovecot`, completado opcionalmente con archivos que se encuentran en el directorio `/etc/dovecot/conf.d`.

Para una configuración clásica y simple, solamente hay que definir el tipo de autenticación por contraseña sin cifrar.

### Sintaxis

```
disable_plaintext_auth = no
```



Se puede configurar la comunicación entre el cliente de mensajería y el servidor Dovecot para usar el protocolo POP o IMAP en SSL, para asegurar la confidencialidad total de las comunicaciones. Si además se desea una confidencialidad total de los intercambios, desde el envío del correo electrónico por el emisor hasta la recepción hecha por el destinatario, habría que implementar un protocolo de extremo a extremo como SMIME.

## b. Visualización de la configuración actual: `doveconf`

El comando `doveconf` ofrece los datos de la configuración del servidor.

La opción `-a` muestra los parámetros efectivos del servidor.

### Ejemplo

```
doveconf -a | head
```

```
# 2.2.36 (1f10bfa63): /etc/dovecot/dovecot.conf
```

```
# Pigeonhole version 0.4.24 (124e06aa)
```

```
# OS: Linux 3.10.0-1062.18.1.el7.x86_64 x86_64 CentOS Linux release 7.7.1908 (Core)
```

```
# Hostname: centos7
# NOTE: Send doveconf -n output instead when asking for help.
auth_anonymous_username = anonymous
auth_cache_negative_ttl = 1 hours
auth_cache_size = 0
auth_cache_ttl = 1 hours
auth_cache_verify_password_with_worker = no
```

La opción `-n` muestra únicamente los parámetros efectivos definidos en el archivo de configuración del servidor.

### Ejemplo

```
dovecot -n
# 2.2.36 (1f10bfa63): /etc/dovecot/dovecot.conf
# Pigeonhole version 0.4.24 (124e06aa)
# OS: Linux 3.10.0-1062.18.1.el7.x86_64 x86_64 CentOS Linux release 7.7.1908 (Core)
# Hostname: centos7
first_valid_uid = 1000
managesieve_notify_capability = mailto
managesieve_sieve_capability = fileinto reject envelope encoded-character vacation
subaddress comparator-i;ascii-numeric relational regex imap4flags copy include
variables body enotify environment mailbox date index ihave duplicate mime
foreverypart extracttext
mbox_write_locks = fcntl
namespace inbox {
  inbox = yes
  location =
  mailbox Drafts {
    special_use = \Drafts
  }
  mailbox Junk {
    special_use = \Junk
  }
  mailbox Sent {
    special_use = \Sent
  }
  mailbox "Sent Messages" {
    special_use = \Sent
```

```

}
mailbox Trash {
    special_use = \Trash
}
prefix =
}
passdb {
    driver = pam
}
plugin {
    sieve = file:~/sieve;active=~/.dovecot.sieve
}
ssl = required
ssl_cert = </etc/pki/dovecot/certs/dovecot.pem
ssl_key = # hidden, use -P to show it
userdb {
    driver = passwd
}

```

También se puede usar un nombre de directiva como argumento, para conocer su valor.

#### Ejemplo

```

doveconf disable_plaintext_auth
disable_plaintext_auth = no

```

### c. El comando **doveadm**

El comando `doveadm` permite interactuar con el servidor Dovecot.

#### Sintaxis

```
doveadm [-v] SubComando [Opciones] [Argumentos]
```

Donde:

|   |  |
|---|--|
| <code>-v</code>                           | Visualización detallada.   |
| <code>SubComando [Opciones] [Args]</code> | Subcomando de <code>doveadm</code> con sus opciones y argumentos opcionales. |

Subcomandos principales:

|                                    |   |
|------------------------------------|---|
| <code>reload</code>                | Recarga la configuración.   |
| <code>stop</code>                  | Para correctamente el servidor.   |
| <code>kick [Nombre   DirIP]</code> | Desconecta el cliente (usando un nombre de usuario o una dirección IP). |
| <code>who</code>                   | Hace una lista de los clientes conectados.                              |
| <code>auth login Nom</code>        | Comprueba la autenticación de un usuario.                               |

### Ejemplo

Prueba de la autenticación de un usuario:

```
doveadm auth login aca
Password: XXX
passdb: aca auth succeeded
extra fields:
user=aca

userdb extra fields:
aca
system_groups_user=aca
```

uid=1000  
gid=1000  
home=/home/aca