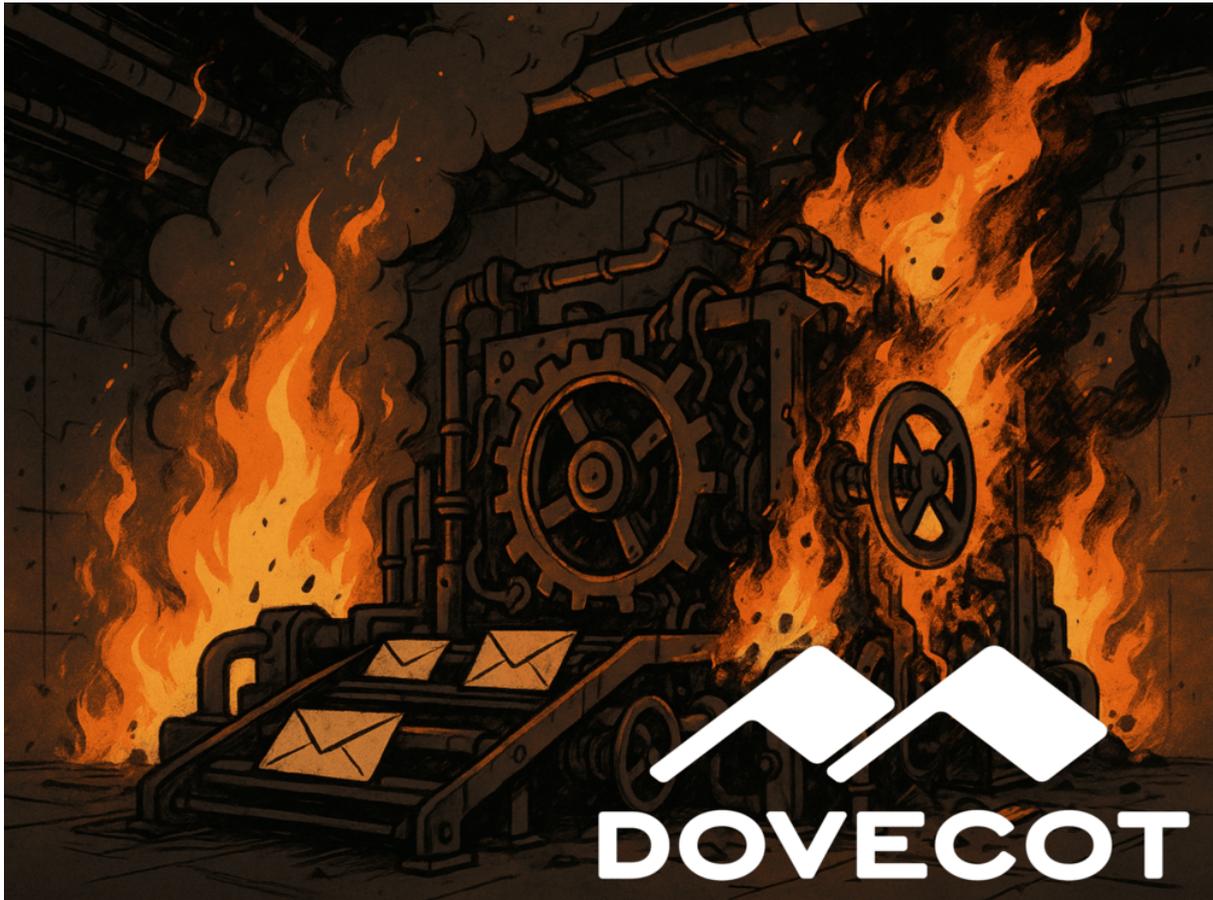


# Cambios importantes

*Actualización de Dovecot 2.3 a 2.4 en Debian Stable*

Willem L. Middelkoop

June 4, 2025



La semana pasada me encontré con un problema inesperado durante un procedimiento rutinario de mantenimiento en la infraestructura de correo electrónico de mi empresa. Procesando miles de correos electrónicos cada día, usamos Dovecot para dar a nuestros clientes acceso a sus mensajes. Todo eso se detuvo repentinamente cuando actualicé a la versión 2.4, que presenta cambios importantes... ¡vaya!

**TL;DR: Actualización de Dovecot 2.3 → 2.4 en Debian**

- **Dovecot 2.4 introduce cambios importantes:** Los archivos de configuración de 2.3 son incompatibles. El software no se iniciará a menos que los reescriba.
- **Nueva sintaxis de configuración y sistema de variables:** Parámetros como ‘ssl\_cert’, ‘mail\_location’ y ‘%d’ han cambiado de formato y significado.

- **No hay ruta de migración automatizada:** Se requiere la revisión manual y la reescritura de la configuración, especialmente si se utilizan backends SQL, Sieve o autenticación personalizada.
- **Recomendación:** Lea la [guía de actualización oficial](#) detenidamente antes de actualizar y pruebe las configuraciones de forma aislada.

## Un paseo por el parque

Pendiente de algunas actualizaciones de seguridad y para resolver algunos problemas de compatibilidad, tomé la decisión de actualizar algunos paquetes en un servidor Debian GNU/Linux que es responsable de los servicios de correo electrónico de mi empresa. Esta distribución particular de GNU/Linux es conocida por su estabilidad y confiabilidad. No había razón para esperar ningún problema y comencé la actualización esperando que terminara en cuestión de minutos. Solo "un paseo por el parque", ¿qué podría salir mal...?

## Requiriendo 'dovecot\_config\_version'

La mayor parte del servidor volvió a estar en línea como se esperaba casi de inmediato, interrumpiendo los servicios tan poco que incluso la [alarma del monitor de tiempo de actividad](#) no detectó ninguna interrupción, excepto Dovecot. Este tipo particular de software es responsable del acceso IMAP y POP3 a los buzones de correo. Encaja como un engranaje finamente ajustado en una configuración más grande donde otros programas como Postfix dependen de su funcionamiento para aceptar el correo electrónico entrante. La falla de Dovecot al iniciar causó una serie de problemas.

Lo primero que se hace como administrador de un servidor es verificar los demonios del servidor ('systemctl status dovecot') seguido de la inspección de los registros, por ejemplo, ejecutando 'journalctl -u dovecot'. Eso reveló rápidamente que Dovecot se negó a iniciar porque faltaba un parámetro de configuración: 'dovecot\_config\_version'.

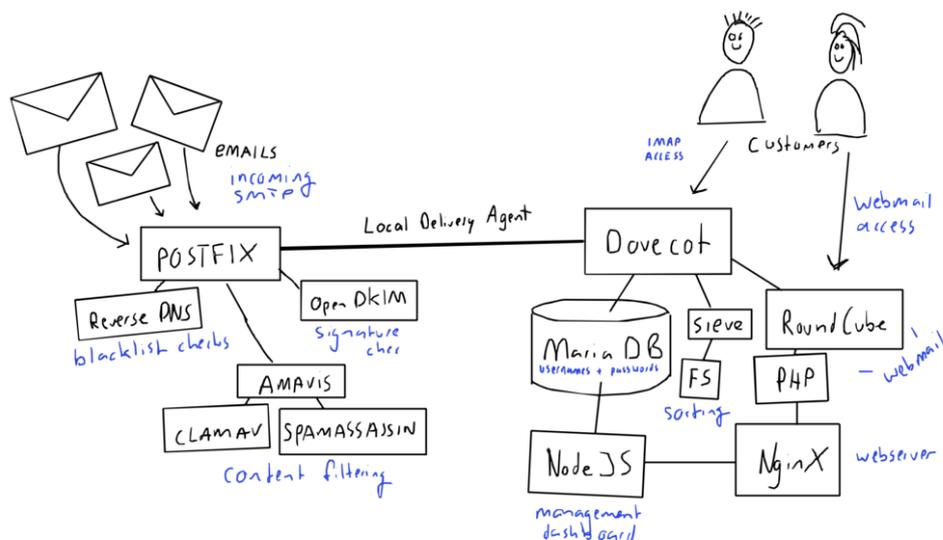
¿Simplemente agregue el parámetro a la configuración en '/etc/dovecot/dovecot.conf' podría pensar? Bueno... NO. Resulta que los creadores de Dovecot pusieron el error por una buena razón: la nueva versión 2.4 de Dovecot es incompatible con los archivos de configuración 2.3 anteriores; lo que significa que deberá reescribir muchos parámetros diferentes. No se trata solo de los nombres de las variables de configuración, sino que también introduce una sintaxis completamente nueva de expansión de variables de configuración. Un valor de variable antiguo como '%d' ahora se convierte en '%{user | domain}'. La lista de posibilidades [es larga](#) y no hay una forma automatizada de migrar su configuración. Imagina mi cara cuando me di cuenta de la magnitud de mi problema.

## Complejidad integrada: engranajes en una máquina

Lo que pasa con los servicios de correo electrónico es que es *difícil*. No porque un solo componente sea difícil, sino porque para ejecutar una pila de correo electrónico completa se necesitan muchas partes diferentes que funcionen juntas sin problemas. Estas son algunas de las piezas que componen el servicio de correo de mi empresa:

- **Postfix:** El agente principal de transferencia de correo SMTP (MTA), se comunica con otros servidores de correo para recibir y enviar correo electrónico. Necesita saber qué direcciones de correo electrónico son válidas para recibir correos y quién está autorizado para enviar nuevos mensajes.

- **Filtro Amavis:** Todos los correos procesados por el servidor pasan por el mecanismo de filtrado amavis que verifica los mensajes individuales utilizando SpamAssassin y ClamAV.
- **SpamAssasin:** Mediante un complejo conjunto de reglas y análisis bayesianos, los mensajes se analizan automáticamente para determinar la probabilidad de que sean spam.
- **ClamAV:** Los mensajes se analizan en busca de virus. Los archivos adjuntos se extraen y analizan, archivo por archivo.
- **Listas negras de DNS inversas:** Además de escanear los mensajes, el servidor de correo también realiza una verificación de antecedentes en los servidores externos con los que se comunica. Se analizan los encabezados de correo y se verifica la reputación de los saltos individuales.
- **OpenDKIM:** Algunos mensajes contienen una firma de 'clave de dominio', lo que demuestra que su remitente está autorizado. Se verifican los mensajes con encabezados, mientras que otros se firman utilizando un servicio de filtrado vinculado a Postfix.
- **Sieve:** Los mensajes entrantes se ordenan automáticamente utilizando reglas por usuario, por ejemplo, haciendo que ciertos mensajes se entreguen en carpetas específicas. Necesita saber qué reglas se aplican a qué direcciones.
- **MariaDB / MySQL:** Las direcciones de correo electrónico, los inicios de sesión válidos, los nombres de usuario, los hashes de contraseñas y los alias se almacenan en una base de datos centralizada. Se configuran consultas SQL específicas para verificar cosas como "¿es esta una contraseña válida?"
- **Nginx:** El servidor de correo electrónico también tiene un servidor web que permite la administración de cuentas y proporciona acceso al correo electrónico a través del webmail.
- **NodeJS:** La consola de administración utiliza un backend NodeJS para permitir que los administradores del servidor y los clientes configuren sus direcciones y contraseñas.
- **Roundcube:** El servidor utiliza un programa de webmail ampliamente utilizado para permitir que los clientes accedan a su correo electrónico a través del navegador.
- **PHP:** El webmail requiere que PHP se ejecute, incluidas algunas dependencias para que se comunique con Dovecot (usando IMAP) y MariaDB (usando SQL).
- **Dovecot:** Luego está Dovecot, el software real que permite a los usuarios finales acceder a su correo electrónico. Permite a los usuarios autenticados acceder a los correos electrónicos que se almacenan en los discos de la infraestructura utilizando protocolos como IMAP o POP3.



Una pila de servidor de correo completa comprende muchos componentes más pequeños ajustados entre sí

Estas piezas están todas finamente ajustadas entre sí, como engranajes en una máquina compleja, encajan con precisión. Su configuración está adaptada a su función específica, a menudo interactuando con varias otras piezas. ¡Puede imaginar que un cambio inesperado en uno de estos engranajes (como Dovecot) puede causar una gran interrupción!

No había una solución rápida disponible, ni una actualización de configuración automatizada. Dado que esta actualización es relativamente nueva, la mayoría de los modelos de lenguaje grande (IA) tampoco tenían una respuesta. ChatGPT simplemente generó sugerencias basadas en la documentación anterior de 2.3. Para resolver el problema, necesitaba aplicar *'habilidades de servidor Linux de la vieja escuela'*: crear manualmente una nueva configuración personalizada para Dovecot 2.4.

## Detener para evitar la pérdida de datos

Una vez que comprendí la naturaleza del problema, detuve inmediatamente la mayoría de los servicios de correo electrónico (por ejemplo, detuve por completo el motor). Esto evita la pérdida de datos ya que los correos electrónicos dejan de procesarse. Esto hace que otros servidores de correo pongan en cola los correos electrónicos, a veces emitiendo errores SMTP 4xx (como 421, 451, 454 o 455). El diseño del protocolo de correo electrónico mundial se adapta a las interrupciones temporales y otros servidores simplemente lo intentarán de nuevo más tarde.

## Reescribiendo la configuración por RTFM

El mensaje de error en los registros me señaló a los [documentos oficiales de Dovecot](#), y allí estaba: una ADVERTENCIA resaltada en amarillo que describe los cambios de configuración. Es un documento extenso: [como PDF cuenta 18 páginas](#). En un día normal, tomaría algún esfuerzo leerlo; ¡imagínese tener que hacer esto mientras la casa proverbial está en llamas!

La configuración de Dovecot consta de varios archivos que configuran partes del comportamiento del software. Por ejemplo, cómo se realiza la autenticación, cómo se integra con un filtro Sieve o dónde obtiene los nombres de usuario y los hashes de contraseña para iniciar sesión. Los cambios introducidos en Dovecot 2.4 afectan a casi todos los parámetros, ya que los creadores han introducido una nueva sintaxis para las variables del servidor.

## Sintaxis de notación variable

Anteriormente, la configuración utilizaba una notación variable como '%u' como marcador de posición para un nombre de usuario. La nueva sintaxis permite canalizar variables a un modificador, lo que permite la normalización y cosas como la selección de subcadenas. Por ejemplo: para obtener un dominio de correo electrónico de un nombre de usuario, ya no usa '%d' sino que usa la variable 'usuario' como base para canalizarla al modificador de dominio usando esta sintaxis: '%{user | domain}'.

Una vez que entendí este nuevo idioma, realmente aprecié la obra de arte que conlleva. Es una mejora con respecto a los nombres de variables abreviados más antiguos, ya que ya no necesita un diccionario para buscar cosas: en lugar de algo vago como '%r', ahora lee '%{remove\_ip}'. Es intrínsecamente más legible, algo que realmente valoro.

## Cifrado: 'ssl\_cert' se convierte en 'ssl\_server\_cert\_file'

Otros cambios incluyen la forma en que la configuración apunta a archivos externos utilizados para el cifrado. Anteriormente, tendría una configuración 'ssl\_cert' que podría contener el certificado SSL/TLS real como cadena. En la versión 2.4, tendrá la variable 'ssl\_server\_cert\_file' que toma una ruta que apunta al certificado en el sistema de archivos. **¡No se trata solo de cambiar el nombre, la naturaleza del valor también es nueva!** Para corregir la configuración, deberá recorrer cada uno de estos parámetros actualizados y determinar su impacto en su configuración particular. Lleva mucho tiempo y es propenso a errores.

## Dividiendo la configuración de 'mail\_location'

Una novedad en 2.4 es cómo se divide 'mail\_location' en varias variables más pequeñas. Anteriormente, una sola variable definiría dónde se guardaba el correo electrónico en el disco. Ahora deberá dividir esto en dos variables 'mail\_driver' y 'mail\_path', **Y** usar la nueva sintaxis de variable del servidor. Si se equivoca en alguno de estos, hará que todos los correos electrónicos desaparezcan (¡créame!). Dependiendo de su configuración de almacenamiento específica, querrá verificar dos veces la parte del nombre de usuario (por ejemplo, ¿incluye la dirección de correo electrónico completa o solo el prefijo sin el nombre de dominio?).

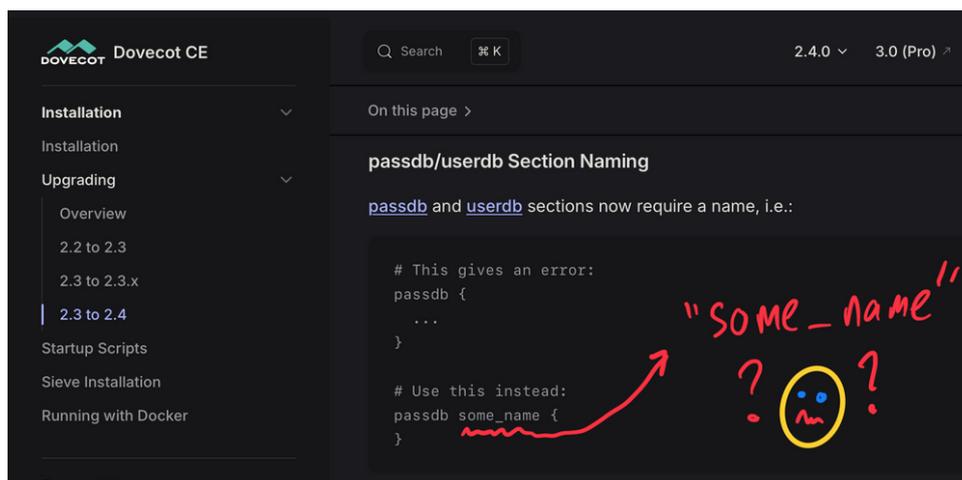
```
# Dovecot 2.3 config:
# mail_location = maildir:/var/vmail/%d/%u

# Corresponds to 2.4 syntax:
mail_driver = maildir
mail_path = /var/vmail/{user | domain }/{user | username}
```

*Migración de mail\_location a mail\_driver y mail\_path utilizando la nueva sintaxis de variables del servidor*

## Dovecot con backend SQL: passdb y userdb

Si su instalación de Dovecot funciona con una base de datos (o almacén) externa para autenticar nombres de usuario y contraseñas, deberá reescribir 'auth-sql.conf.ext' en '/etc/dovecot/conf.d/' que es llamado/incluido por el archivo 10-auth.conf. En Dovecot 2.4 hay una nueva forma de segmentar la configuración de passdb y userdb. En la configuración de Dovecot 2.3 utilicé un archivo externo para proporcionar las consultas de la base de datos y la cadena de conexión. Durante la autenticación, las múltiples consultas se utilizan para 1) autenticar a los usuarios y 2) cargar la configuración específica del usuario, como la ruta del almacén de correo electrónico.



*Las instrucciones de actualización de userdb y passdb son muy útiles a primera vista /s*

```
# /etc/dovecot/dovecot-sql.conf.ext
# This file is opened as root, so it should be owned by root and mode 0600.
#
# Database driver: mysql, postgresql, sqlite
driver = mysql

# Database connection string. This is driver-specific setting.
# mysql:
# Basic options emulate PostgreSQL option names:
# host, port, user, password, dbname
#
# Examples:
# connect = host-192.168.1.1 dbname=users
# connect = host-sql.example.com dbname=virtual user=virtual password=blarg
# connect = /etc/dovecot/authdb.sqlite
connect = host=localhost dbname=mail user=super password=secret

# Default password scheme.
#
# List of supported schemes is in
# http://wiki.dovecot.org/Authentication/PasswordSchemes
#
default_pass_scheme = MD5

# Query to retrieve the password.
#
# This query must return only one row with "user" and "password" columns.
# The query can also return other fields which have a special meaning, see
# http://wiki.dovecot.org>PasswordDatabase/ExtraFields
#
# The "user" column is needed to make sure the username gets used with exactly
# the same casing as it's in the database. Note that if you store username and
# domain in separate fields, you most likely want to return a combination of
# them as the "user" column, otherwise the domain gets stripped.
#
# Commonly used available substitutions (see
# http://wiki.dovecot.org/Variables for full list):
# %u = entire userid
# %n = user part of user@domain
# %d = domain part of user@domain
#
# Example:
password_query = SELECT password FROM users WHERE userid = '%n' AND domain = '%d'
password_query = SELECT pw AS password FROM users WHERE userid = '%u' AND active = 'Y'
password_query = SELECT username as user, password, '/var/mail/%d/%n' as userdb_home, '%d:/var/mail/%d/%n' as userdb_mail, 150 as userdb_uid, 8 as userdb_gid FROM mailbox WHERE username='%u' AND active='1'

# Query to retrieve the user information.
#
# The query must return only one row. Commonly returned columns are:
# uid - System UID
# gid - System GID
# home - Home directory
# mail - Mail location
#
# Either home or mail is required, uid and gid are required. If more than one
# row is returned or there are missing fields, the login will fail. For a list
# of all fields that can be returned, see
# http://wiki.dovecot.org/UserDatabase/ExtraFields
#
# Examples
# user_query = SELECT home, uid, gid FROM users WHERE userid = '%n' AND domain = '%d'
# user_query = SELECT dir AS home, user AS uid, group AS gid FROM users where userid = '%u'
# user_query = SELECT home, 501 AS uid, 501 AS gid FROM users WHERE userid = '%u'
#
# user_query = SELECT '/var/mail/%d/%n' as home, '%d:/var/mail/%d/%n' as mail, 150 as uid, 8 as gid, concat('dirsize:storage',quota) as quota from mailbox where username='%u' AND active='1'
iterate_query = SELECT username as user FROM mailbox;
```

### Configuración del backend SQL de Dovecot 2.3 (auth-sql.conf.ext y dovecot-sql.conf.ext)

En la nueva versión 2.4, debe reestructurar los parámetros para que estén incluidos en secciones que incluyan un indicador de tipo de backend justo detrás de las palabras clave 'userdb' y 'passdb' en la configuración. Entonces, si usa una base de datos MySQL como backend, necesitará que 'userdb' se convierta en 'userdb sql'. Puede incluir los parámetros de la cadena de conexión directamente en el archivo 'auth-sql.conf.ext' para que todos los parámetros relevantes estén en un archivo de configuración. Para reducir la complejidad, reescribí la sección userdb para usar el controlador de base de datos 'estático': generando variables de configuración como la ruta de correo del usuario en función del nombre de usuario del usuario. Consulte los [documentos dedicados](#).

```

1 # Authentication for SQL users. Included from auth.conf.
2 #
3 # DOVECOT 2.4
4 #
5 # <https://doc.dovecot.org/latest/core/config/auth/databases/sql.html>
6
7 # Database driver: mysql, postgres, sqlite
8 sql_driver = mysql
9
10 # Database connection string. This is driver-specific setting.
11 mysql localhost {
12     user = super
13     password = secret
14     dbname=mail
15 }
16
17 passdb sql {
18     default_password_scheme = MD5
19     query = SELECT username as user, password FROM mailbox WHERE username='%{user}' AND active='1'
20 }
21
22 # If you don't have any user-specific settings, you can avoid the user_query
23 # by using userdb static instead of userdb sql, for example:
24 # <https://doc.dovecot.org/latest/core/config/auth/databases/static.html>
25 userdb static {
26     fields {
27         uid = 150
28         gid = 8
29         home = /var/vmail/{user | domain}/{user | username}
30     }
31 }

```

*Configuración del backend SQL de Dovecot 2.4 (auth-sql.conf.ext) - ahora mucho más compacta que su predecesora. Ahora solo hay una consulta SQL ya que la sección userdb ahora usa el backend estático*

## Complemento Sieve

Si su configuración utiliza Sieve para ordenar los correos electrónicos entrantes, deberá ajustar su configuración con respecto a la ubicación del script, sus valores predeterminados y su comportamiento. La nueva versión de Dovecot 2.4 introduce un concepto llamado "Almacenamiento de scripts" que permite que se ejecuten scripts cargados dinámicamente para diferentes usuarios en varios eventos. Nuestra configuración implica un script de tamiz predeterminado simple que ordena el correo electrónico marcado como spam automáticamente a una carpeta llamada Spam. La nueva sintaxis permite una configuración más compacta:

```

##
## Settings for the Sieve interpreter
##
## DOVECOT 2.3
##

# Do not forget to enable the Sieve plugin in 15-lda.conf and 20-lmtp.conf
# by adding it to the respective mail_plugins= settings.

plugin {
  # The path to the user's main active script. If ManageSieve is used, this the
  # location of the symbolic link controlled by ManageSieve.
  sieve = ~/.dovecot.sieve

  # The default Sieve script when the user has none. This is a path to a global
  # sieve script file, which gets executed ONLY if user's private Sieve script
  # doesn't exist. Be sure to pre-compile this script manually using the sievec
  # command line tool.
  # --> See sieve_before fore executing scripts before the user's personal
  # script.
  sieve_default = /var/lib/dovecot/sieve/default.sieve

  # Directory for :personal include scripts for the include extension. This
  # is also where the ManageSieve service stores the user's scripts.
  sieve_dir = ~/sieve
}

```

*Configuración de Sieve en Dovecot 2.3 (90-sieve.conf) con las variables ahora obsoletas sieve\_default y sieve\_dir*

```

1 # /etc/dovecot/conf.d/90-sieve.conf
2 #
3 # DOVECOT 2.4
4 #
5 # Sieve default settings loading global script
6
7 sieve_script default {
8     type = default
9     name = default
10    driver = file
11    path = /var/lib/dovecot/sieve/default.sieve
12 }

```

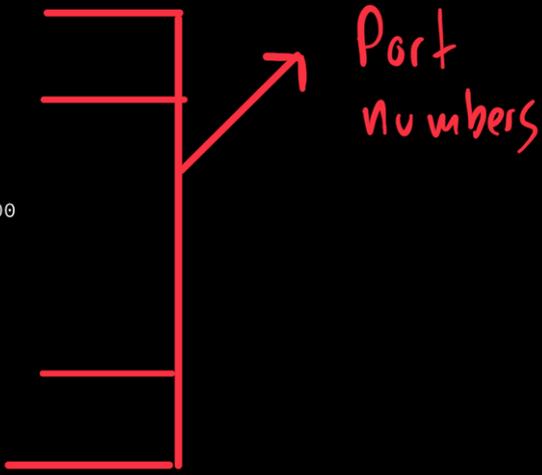
*Configuración de Sieve en Dovecot 2.4 (90-sieve.conf)*

## Probando la nueva configuración

Es posible que tenga la tentación de simplemente iniciar el servidor nuevamente después de redefinir la configuración 2.4. Pero debe tener mucho cuidado, ya que una configuración incorrecta podría causar que los clientes pierdan el acceso a todos los correos electrónicos y/o los obligue a volver a descargar todos sus buzones. En su lugar, opté por un enfoque gradual en el que cambié los números de puerto IMAP (y POP3) a algo arbitrario. De esta manera, pude probar la nueva configuración por mí mismo mientras evitaba que los clientes reales se conectaran al servidor. Esto me permitió corregir algunos errores de escritura, refinar parte de la nueva sintaxis de variables del servidor y lidiar con algunos mensajes de registro extraños. Puede controlar los números de puerto del servidor desde

el archivo '10-master.conf'.

```
1 #
2 # Dovecot 2.4 /etc/dovecot/conf.d/10-master.conf
3 #
4
5 service imap-login {
6   inet_listener imap {
7     port = 143
8   }
9   inet_listener imaps {
10    port = 993
11    ssl = yes
12  }
13
14  process_min_avail = 100
15  vsz_limit = 1024M
16 }
17
18 service pop3-login {
19   inet_listener pop3 {
20     port = 110
21   }
22   inet_listener pop3s {
23     port = 995
24     ssl = yes
25   }
26 }
```



Port numbers

*Configurar los números IMAP y POP del servidor para habilitar la depuración privada/gradual de la nueva configuración*

## Conclusión

Las interrupciones como esta son raras, gracias a la estabilidad de Debian GNU/Linux, pero no imposibles. Existen copias de seguridad diarias y un plan de recuperación ante desastres probado para el servicio de correo electrónico Lemmid Online de mi empresa. Si bien esta vez no se requirió una conmutación por error completa, los ejercicios previos resultaron cruciales para adaptar rápidamente la configuración.

Este incidente fue un recordatorio de que incluso con la automatización y la IA, el conocimiento profundo del sistema, la preparación y la depuración manual siguen siendo herramientas esenciales cuando los componentes críticos de la infraestructura cambian inesperadamente y sin rutas de migración automatizadas.

## Configuración de muestra de Dovecot 2.4

Si encuentra esta publicación buscando una solución para su propio servidor de correo roto, es posible que estos archivos de configuración de muestra de Dovecot 2.4 le resulten útiles. Los encontré durante mi propio trabajo de reparación y podrían ahorrarle algo de tiempo para descubrir la nueva sintaxis: <https://source.willem.com/dovecot-2.4-sample-config/>