

Unknown Title

ChatGPT●

Esto es una copia de una conversación entre ChatGPT y Anónimo.

Tú dijiste:

si ya tengo el ldap instalado, como hago lo siguiente: La empresa ficticia TechSolutions S.L., dedicada al desarrollo de software, cuenta con 40 empleados repartidos en 3 departamentos: Desarrollo, Soporte y Administración. Han desplegado un servidor LDAP para centralizar la gestión de usuarios. Ahora necesitan: 1. Que todos los empleados accedan a su correo electrónico corporativo con las mismas credenciales del directorio LDAP. 2. Que cada empleado tenga acceso a un espacio de almacenamiento en la nube con OwnCloud, también validado con LDAP. -Configurar el servidor de correo existente (ejemplo: Postfix + Dovecot en Linux) para que valide a los usuarios de ldap.

ChatGPT Plus

Perfecto 👍 Veamos cómo lograr lo que necesitas paso a paso.
Ya tienes LDAP instalado, así que partimos desde ahí.

El objetivo es **hacer que Postfix y Dovecot autentiquen contra el servidor LDAP**, de modo que los usuarios usen las mismas credenciales del directorio para el correo corporativo y también para OwnCloud.

1. Requisitos previos

Asegúrate de tener:

- Servidor LDAP operativo (por ejemplo, `ldap://ldap.techsolutions.local`)
- Postfix y Dovecot instalados
- Paquetes de soporte LDAP:

bash

```
sudo apt install libsasl2-modules-ldap dovecot-ldap
```

- Un dominio de correo (por ejemplo, `techsolutions.local`)

2. Configurar Postfix con autenticación LDAP

a. Habilitar autenticación SASL (usando Dovecot)

Edita `/etc/postfix/main.cf` y añade o modifica:

bash

```
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_recipient_restrictions =
    permit_sasl_authenticated,
    permit_mynetworks,
    reject_unauth_destination
```

También asegúrate de tener:

bash

```
mydomain = techsolutions.local
myhostname = mail.techsolutions.local
myorigin = /etc/mailname
```

Guarda y recarga:

bash

```
sudo systemctl reload postfix
```

3. Configurar Dovecot para autenticación LDAP

Edita /etc/dovecot/conf.d/10-auth.conf:

bash

```
disable_plaintext_auth = no
auth_mechanisms = plain login
!include auth-ldap.conf.ext
```

Luego edita /etc/dovecot/conf.d/auth-ldap.conf.ext:

bash

```
passdb {
    driver = ldap
    args = /etc/dovecot/dovecot-ldap.conf.ext
}

userdb {
    driver = ldap
    args = /etc/dovecot/dovecot-ldap.conf.ext
}
```

4. Configurar conexión LDAP en Dovecot

Edita /etc/dovecot/dovecot-ldap.conf.ext:

bash

```
hosts = ldap.techsolutions.local
dn = cn=admin,dc=techsolutions,dc=local
dnpass = tu_contraseña_admin_ldap
base = ou=empleados,dc=techsolutions,dc=local
ldap_version = 3
scope = subtree
auth_bind = yes
dnpass = tu_contraseña_admin_ldap

user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))

# Maildir para cada usuario
user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail
default_pass_scheme = SSHA
```

Tip: Si usas `inetOrgPerson` en lugar de `posixAccount`, adapta los atributos (`uid`, `mail`, etc.) según tu esquema LDAP.

Recarga Dovecot:

bash

```
sudo systemctl restart dovecot
```

✓ 5. Prueba de autenticación

Ejecuta:

bash

```
telnet localhost 143
```

y prueba un login:

css

```
a login usuario@techsolutions.local contraseña
```

Si obtienes `OK Logged in`, Dovecot ya está autenticando contra LDAP.

☁ 6. Integrar OwnCloud con LDAP

En la interfaz web de OwnCloud (como administrador):

1. Ve a **Aplicaciones** → **Usuarios externos (LDAP/AD Integration)**

2. Configura:

- Servidor LDAP: `ldap.techsolutions.local`
- Puerto: 389
- DN base: `dc=techsolutions,dc=local`
- Filtro de usuarios: `(objectClass=posixAccount)`
- Filtro de grupos: `(objectClass=posixGroup)`

3. Prueba la conexión y guarda.

Ahora los usuarios podrán iniciar sesión en OwnCloud con las mismas credenciales LDAP.

7. (Opcional) Depuración

- Ver logs:

bash

```
tail -f /var/log/mail.log
tail -f /var/log/dovecot.log
```

- Verifica usuarios:

bash

```
ldapsearch -x -b "ou=empleados,dc=techsolutions,dc=local" "(uid=jgomez)"
```

¿Quieres que te prepare los archivos de configuración listos (por ejemplo, para `/etc/dovecot/dovecot-ldap.conf.ext` y `/etc/postfix/main.cf`) con tus valores de dominio y estructura LDAP específicos?

Solo necesito saber:

- El dominio de correo (¿`techsolutions.local` o `techsolutions.com`?)
- La ruta base LDAP (por ejemplo `dc=techsolutions,dc=local`)
- Cómo se llama tu unidad organizativa de usuarios (`ou=empleados`, `ou=users`, etc.)

Tú dijiste:

mi ldap es villegas.basilio

ChatGPT Plus

Perfecto, entonces tu dominio LDAP es `villegas.basilio`. Con eso podemos ajustar la configuración de Dovecot, Postfix y OwnCloud para que autentiquen correctamente contra tu LDAP.

Te preparo un ejemplo concreto de configuración:

1 Postfix (SMTP) con autenticación SASL vía Dovecot

Archivo `/etc/postfix/main.cf` (solo las partes relevantes):

bash

```
myhostname = mail.villegas.basilio
mydomain = villegas.basilio
```

```
myorigin = /etc/mailname
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_recipient_restrictions =
    permit_sasl_authenticated,
    permit_mynetworks,
    reject_unauth_destination
```

Recarga Postfix:

bash

```
sudo systemctl reload postfix
```

2 Dovecot (IMAP/POP3) con LDAP

a. /etc/dovecot/conf.d/10-auth.conf

bash

```
disable_plaintext_auth = no
auth_mechanisms = plain login
!include auth-ldap.conf.ext
```

b. /etc/dovecot/conf.d/auth-ldap.conf.ext

bash

```
passdb {
    driver = ldap
    args = /etc/dovecot/dovecot-ldap.conf.ext
}

userdb {
    driver = ldap
    args = /etc/dovecot/dovecot-ldap.conf.ext
}
```

c. /etc/dovecot/dovecot-ldap.conf.ext

bash

```
hosts = ldap.villegas.basilio
dn = cn=admin,dc=villegas,dc=basilio
dnpass = TU_CONTRASEÑA_ADMIN_LDAP
base = ou=empleados,dc=villegas,dc=basilio
ldap_version = 3
scope = subtree
auth_bind = yes

user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))

user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail
default_pass_scheme = SSHA
```

Luego recarga Dovecot:

bash

```
sudo systemctl restart dovecot
```

3 OwnCloud con LDAP

1. Instala la app **LDAP/AD Integration** en OwnCloud.
2. Configura en la interfaz web:
 - Servidor LDAP: `ldap.villegas.basilio`
 - Puerto: 389 (o 636 para LDAPs)
 - DN base: `dc=villegas,dc=basilio`
 - Filtro de usuarios: `(objectClass=posixAccount)`
 - Filtro de grupos: `(objectClass=posixGroup)`
3. Testea la conexión y guarda.

4 Probar autenticación

- Dovecot:

bash

```
telnet localhost 143
a login usuario@villegas.basilio CONTRASEÑA
```

- OwnCloud:
Inicia sesión con cualquier usuario LDAP, debería funcionar.
-

Si quieres, puedo **generarte todos los archivos listos para copiar y pegar** con valores exactos de tu dominio y OU, listos para que solo pongas la contraseña LDAP y recargues los servicios. Esto ahorra mucho tiempo y errores.

¿Quieres que haga eso?

Tú dijiste:

empieza desde el principio con la informacion que te he dado

ChatGPT Plus

Perfecto 😎 Vamos a hacerlo desde cero usando tu información: tu dominio LDAP es **villegas.basilio**, y quieres que los 40 empleados puedan:

1. Acceder a su correo corporativo con LDAP.
2. Acceder a OwnCloud con las mismas credenciales LDAP.

Supondré que tu LDAP tiene una estructura típica:

- **Base DN:** dc=villegas,dc=basilio
- **Usuarios:** ou=empleados,dc=villegas,dc=basilio
- **Objetos de usuario:** posixAccount
- **Grupo:** posixGroup

Vamos paso a paso.

1 Preparar el servidor de correo

Instala Postfix y Dovecot (si no están instalados):

bash

```
sudo apt update
sudo apt install postfix dovecot-core dovecot-imapd dovecot-pop3d dovecot-
ldap libsasl2-modules-ldap
```

Durante la instalación de Postfix, elige **"Sitio de Internet"** y configura el dominio:

```
mail.villegas.basilio
```

2 Configurar Postfix para usar Dovecot SASL (LDAP indirecto)

Edita /etc/postfix/main.cf y añade:

bash

```
myhostname = mail.villegas.basilio
mydomain = villegas.basilio
myorigin = /etc/mailname

smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous

smtpd_recipient_restrictions =
    permit_sasl_authenticated,
    permit_mynetworks,
    reject_unauth_destination
```

Recarga Postfix:

bash

```
sudo systemctl reload postfix
```

3 Configurar Dovecot para autenticar con LDAP

a. Habilitar autenticación LDAP

Edita /etc/dovecot/conf.d/10-auth.conf:

bash

```
disable_plaintext_auth = no
auth_mechanisms = plain login
!include auth-ldap.conf.ext
```

b. Configurar auth-ldap

Crea o edita /etc/dovecot/conf.d/auth-ldap.conf.ext:

bash

```
passdb {
    driver = ldap
    args = /etc/dovecot/dovecot-ldap.conf.ext
}

userdb {
    driver = ldap
    args = /etc/dovecot/dovecot-ldap.conf.ext
}
```

c. Configurar conexión LDAP

Crea o edita `/etc/dovecot/dovecot-ldap.conf.ext`:

bash

```
hosts = ldap.villegas.basilio
dn = cn=admin,dc=villegas,dc=basilio
dnpass = TU_CONTRASEÑA_ADMIN_LDAP
base = ou=empleados,dc=villegas,dc=basilio
ldap_version = 3
scope = subtree
auth_bind = yes

user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))

user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail
default_pass_scheme = SSHA
```

Recarga Dovecot:

bash

```
sudo systemctl restart dovecot
```

4 Probar autenticación de correo

bash

```
telnet localhost 143
a login usuario@villegas.basilio CONTRASEÑA
```

Si recibes `OK Logged in`, la autenticación LDAP funciona.

5 Configurar OwnCloud para LDAP

1. Instala la app **LDAP/AD Integration**.
2. Configura:
 - Servidor LDAP: `ldap.villegas.basilio`
 - Puerto: 389 (o 636 para LDAPS)
 - DN base: `dc=villegas,dc=basilio`
 - Filtro de usuarios: `(objectClass=posixAccount)`
 - Filtro de grupos: `(objectClass=posixGroup)`
3. Testea conexión → Guarda.

Ahora todos los empleados podrán usar **las mismas credenciales LDAP** para correo y OwnCloud.

Si quieres, puedo hacer una **versión lista para copiar y pegar de todos los archivos** (`main.cf`, `dovecot-ldap.conf.ext`, `auth-ldap.conf.ext`) usando directamente tu dominio `villegas.basilio`, así solo tendrías que poner la contraseña LDAP y recargar los servicios.

¿Quieres que haga eso?

Tú dijiste:

```
vuelve a empezar, mi estructura es la siguiente: root@LDAP-Ubu:~# ldapsearch -x -b
"dc=villegas,dc=basilio" "(objectclass=*)" # extended LDIF ## LDAPv3 # base <dc=villegas,dc=basilio>
with scope subtree # filter: (objectclass=*) # requesting: ALL ## villegas.basilio dn: dc=villegas,dc=basilio
objectClass: top objectClass: dcObject objectClass: organization o: villegas.basilio dc: villegas #
usuarios, villegas.basilio dn: ou=usuarios,dc=villegas,dc=basilio objectClass: organizationalUnit ou:
usuarios # equipos, villegas.basilio dn: ou=equipos,dc=villegas,dc=basilio objectClass: organizationalUnit
ou: equipos # villegas, usuarios, villegas.basilio dn: cn=villegas,ou=usuarios,dc=villegas,dc=basilio
objectClass: inetOrgPerson objectClass: posixAccount objectClass: shadowAccount loginShell: /bin/bash
homeDirectory: /home/villegas uid: villegas cn: villegas uidNumber: 10002 gidNumber: 10001 sn: villegas
mail: villegas@villegas.basilio # 912345678, usuarios, villegas.basilio dn:
cn=912345678,ou=usuarios,dc=villegas,dc=basilio objectClass: inetOrgPerson objectClass:
posixAccount objectClass: shadowAccount loginShell: /bin/bash homeDirectory: /home/basilio uid: basilio
gidNumber: 10001 sn: basilio mail: basilio@villegas.basilio uidNumber: 1100 cn: 912345678 # sergio,
usuarios, villegas.basilio dn: cn=sergio,ou=usuarios,dc=villegas,dc=basilio objectClass: inetOrgPerson
objectClass: posixAccount objectClass: shadowAccount loginShell: /bin/bash homeDirectory:
/home/sergio uid: sergio cn: sergio uidNumber: 10004 gidNumber: 10001 sn: sergio mail:
sergio@villegas.basilio # svb, usuarios, villegas.basilio dn: cn=svb,ou=usuarios,dc=villegas,dc=basilio
objectClass: inetOrgPerson objectClass: posixAccount objectClass: shadowAccount loginShell: /bin/bash
homeDirectory: /home/svb uid: svb cn: svb uidNumber: 10005 gidNumber: 10001 sn: svb mail:
```

```
svb@villegas.basilio # grupos, villegas.basilio dn: ou=grupos,dc=villegas,dc=basilio objectClass: top
objectClass: organizationalUnit ou: grupos # ASIR1, grupos, villegas.basilio dn:
cn=ASIR1,ou=grupos,dc=villegas,dc=basilio objectClass: posixGroup cn: ASIR1 gidNumber: 20001
memberUid: villegas memberUid: basilio # ASIR2, grupos, villegas.basilio dn:
cn=ASIR2,ou=grupos,dc=villegas,dc=basilio objectClass: posixGroup cn: ASIR2 gidNumber: 20002
memberUid: sergio memberUid: svb # administradores, villegas.basilio dn:
cn=administradores,dc=villegas,dc=basilio objectClass: posixGroup gidNumber: 10000 cn:
administradores memberUid: svb # svb111, villegas.basilio dn: cn=svb111,dc=villegas,dc=basilio
objectClass: posixAccount objectClass: inetOrgPerson objectClass: organizationalPerson objectClass:
person loginShell: /bin/bash homeDirectory: /home/svb111 uid: svb111 cn: svb111 uidNumber: 10006
gidNumber: 10000 sn: svb111 mail: svb111@gmail.com homePhone: 123456789 # search result search:
2 result: 0 Success # numResponses: 13 # numEntries: 12 root@LDAP-Ubu:~#
```

ChatGPT Plus

¡Perfecto! Ahora que tenemos tu **estructura LDAP completa**, podemos hacer una configuración concreta de correo y OwnCloud. 😎

Resumen de tu LDAP:

- Base DN: dc=villegas,dc=basilio
- Unidad organizativa de usuarios: ou=usuarios,dc=villegas,dc=basilio
- Objetos de usuario: inetOrgPerson, posixAccount, shadowAccount
- Correo de usuarios: atributo mail
- Grupo principal: ou=grupos,dc=villegas,dc=basilio

Objetivo: Todos los usuarios LDAP puedan:

1. Acceder a correo (Postfix + Dovecot) con sus credenciales LDAP.
2. Acceder a OwnCloud con las mismas credenciales.

1 Preparar servidor de correo (Postfix + Dovecot)

Instalar paquetes necesarios:

bash

```
sudo apt update
sudo apt install postfix dovecot-core dovecot-imapd dovecot-pop3d dovecot-
ldap libsasl2-modules-ldap
```

Durante la instalación de Postfix:

- Tipo de instalación: **Sitio de Internet**
 - Dominio: villegas.basilio
-

2 Configurar Postfix para usar Dovecot SASL

Edita /etc/postfix/main.cf y añade:

bash

```
myhostname = mail.villegas.basilio
mydomain = villegas.basilio
myorigin = /etc/mailname

smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous

smtpd_recipient_restrictions =
    permit_sasl_authenticated,
    permit_mynetworks,
    reject_unauth_destination
```

Recarga Postfix:

bash

```
sudo systemctl reload postfix
```

3 Configurar Dovecot para autenticación LDAP

a. Habilitar auth-ldap

Edita /etc/dovecot/conf.d/10-auth.conf:

bash

```
disable_plaintext_auth = no
auth_mechanisms = plain login
!include auth-ldap.conf.ext
```

b. Configurar auth-ldap.conf.ext

Crea o edita /etc/dovecot/conf.d/auth-ldap.conf.ext:

bash

```
passdb {  
    driver = ldap  
    args = /etc/dovecot/dovecot-ldap.conf.ext  
}  
  
userdb {  
    driver = ldap  
    args = /etc/dovecot/dovecot-ldap.conf.ext  
}
```

c. Configurar conexión LDAP en dovecot-ldap.conf.ext

bash

```
hosts = ldap.villegas.basilio  
dn = cn=admin,dc=villegas,dc=basilio  
dnpass = TU_CONTRASEÑA_ADMIN_LDAP  
base = ou=usuarios,dc=villegas,dc=basilio  
ldap_version = 3  
scope = subtree  
auth_bind = yes  
  
user_filter = (&(objectClass=posixAccount)(uid=%u))  
pass_filter = (&(objectClass=posixAccount)(uid=%u))  
  
# Mapear atributos LDAP a Dovecot  
user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail  
default_pass_scheme = SSHA
```

Recarga Dovecot:

bash

```
sudo systemctl restart dovecot
```

4 Probar autenticación de correo

bash

```
telnet localhost 143  
a login villegas@villegas.basilio TU_CONTRASEÑA
```

Si recibes OK Logged in, la autenticación LDAP funciona.

5 Configurar OwnCloud para LDAP

1. Instala la app **LDAP/AD Integration**.
2. Configura la conexión:
 - Servidor LDAP: `ldap.villegas.basilio`
 - Puerto: 389
 - DN base: `ou=usuarios,dc=villegas,dc=basilio`
 - Filtro de usuarios: `(objectClass=posixAccount)`
 - Filtro de grupos: `(objectClass=posixGroup)`
3. Testea conexión → Guarda.

Ahora todos los usuarios LDAP pueden iniciar sesión con las mismas credenciales en correo y OwnCloud.

Si quieres, puedo **prepararte todos los archivos listos para copiar y pegar** (`main.cf`, `auth-ldap.conf.ext`, `dovecot-ldap.conf.ext`) ya configurados con tu LDAP (`villegas.basilio`) y tu estructura exacta de usuarios, solo faltaría poner la contraseña del admin LDAP.

¿Quieres que haga eso?

Tú dijiste:

vuelve a empezar teniendo en cuenta que estoy conectado como root así que los comandos no deberán tener el `sudo` delante

ChatGPT Plus

¡Perfecto! Entonces partimos desde cero, **como root**, y vamos a configurar todo paso a paso teniendo en cuenta tu LDAP (`villegas.basilio`) y la estructura exacta que compartiste. No usaré `sudo` en los comandos.

1 Preparar servidor de correo (Postfix + Dovecot)

Instalar paquetes necesarios:

```
bash
```

```
apt update
apt install postfix dovecot-core dovecot-imapd dovecot-pop3d dovecot-ldap
libsasl2-modules-ldap
```

Durante la instalación de Postfix:

- Tipo de instalación: **Sitio de Internet**
- Dominio: `villegas.basilio`

2 Configurar Postfix para autenticación SASL con Dovecot

Editar `/etc/postfix/main.cf` y añadir las siguientes líneas:

bash

```
myhostname = mail.villegas.basilio
mydomain = villegas.basilio
myorigin = /etc/mailname

smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous

smtpd_recipient_restrictions =
    permit_sasl_authenticated,
    permit_mynetworks,
    reject_unauth_destination
```

Recargar Postfix:

bash

```
systemctl reload postfix
```

3 Configurar Dovecot para autenticar usuarios LDAP

a. Habilitar autenticación LDAP

Editar `/etc/dovecot/conf.d/10-auth.conf`:

bash

```
disable_plaintext_auth = no
auth_mechanisms = plain login
!include auth-ldap.conf.ext
```

b. Configurar auth-ldap

Editar o crear /etc/dovecot/conf.d/auth-ldap.conf.ext:

bash

```
passdb {
    driver = ldap
    args = /etc/dovecot/dovecot-ldap.conf.ext
}

userdb {
    driver = ldap
    args = /etc/dovecot/dovecot-ldap.conf.ext
}
```

c. Configurar conexión LDAP en Dovecot

Editar o crear /etc/dovecot/dovecot-ldap.conf.ext:

bash

```
hosts = ldap.villegas.basilio
dn = cn=admin,dc=villegas,dc=basilio
dnpass = TU_CONTRASEÑA_ADMIN_LDAP
base = ou=usuarios,dc=villegas,dc=basilio
ldap_version = 3
scope = subtree
auth_bind = yes

# Filtrar usuarios y contraseñas
user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))

# Mapear atributos LDAP a Dovecot
user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail
default_pass_scheme = SSHA
```

Recargar Dovecot:

bash

```
systemctl restart dovecot
```

4 Probar autenticación de correo

Probar IMAP contra Dovecot:

bash

```
telnet localhost 143  
a login villegas@villegas.basilio TU_CONTRASEÑA
```

Si responde OK Logged in, la autenticación LDAP funciona correctamente.

5 Configurar OwnCloud para usar LDAP

1. Instalar la app **LDAP/AD Integration**.
2. Configuración en la interfaz web de OwnCloud:
 - Servidor LDAP: `ldap.villegas.basilio`
 - Puerto: 389
 - DN base: `ou=usuarios,dc=villegas,dc=basilio`
 - Filtro de usuarios: `(objectClass=posixAccount)`
 - Filtro de grupos: `(objectClass=posixGroup)`
3. Testear conexión → Guardar.

Ahora todos los usuarios pueden iniciar sesión en OwnCloud con las mismas credenciales LDAP que usan para el correo.

Si quieres, puedo generarte **todos los archivos listos para copiar y pegar** (`main.cf`, `auth-ldap.conf.ext`, `dovecot-ldap.conf.ext`) ya configurados con tu dominio `villegas.basilio` y tu estructura LDAP exacta. Solo tendrías que poner la contraseña del admin LDAP y recargar servicios.

¿Quieres que haga eso?

Tú dijiste:

con esto a que te refieres: **5** Configurar OwnCloud para usar LDAP Instalar la app LDAP/AD Integration. Configuración en la interfaz web de OwnCloud: Servidor LDAP: `ldap.villegas.basilio` Puerto:

389 DN base: ou=usuarios,dc=villegas,dc=basilio Filtro de usuarios: (objectClass=posixAccount) Filtro de grupos: (objectClass=posixGroup) Testear conexión → Guardar. Ahora todos los usuarios pueden iniciar sesión en OwnCloud con las mismas credenciales LDAP que usan para el correo.

ChatGPT Plus

Claro, te explico con más detalle y paso a paso lo que significa eso:

En OwnCloud, para que los usuarios puedan **iniciar sesión con sus credenciales LDAP** (las mismas que usan para el correo), no basta con instalar OwnCloud; hay que **conectar OwnCloud a tu servidor LDAP**. Para eso se usa una aplicación dentro de OwnCloud llamada **LDAP / AD Integration**.

1 Instalar la app LDAP/AD Integration

- Accede a OwnCloud como administrador.
- Ve a **Aplicaciones** → **Integraciones** o **Market**.
- Busca **LDAP / AD Integration** y actívala.

Esta app permite que OwnCloud lea tu directorio LDAP y autentique usuarios directamente desde él.

2 Configuración de conexión LDAP

Dentro de la interfaz de OwnCloud, en **Configuración** → **LDAP / AD Integration**, se deben completar estos campos:

1. Servidor LDAP:

- Aquí pones tu servidor LDAP: `ldap.villegas.basilio`

2. Puerto:

- Normalmente 389 para LDAP sin cifrado o 636 para LDAPS (LDAP seguro).

3. DN Base (Base DN):

- Es el punto desde el que OwnCloud empezará a buscar usuarios.
- Para tu LDAP sería: `ou=usuarios,dc=villegas,dc=basilio`

4. Filtro de usuarios:

- Indica qué objetos en LDAP se consideran usuarios.
- En tu caso: `(objectClass=posixAccount)`

5. Filtro de grupos:

- Indica qué objetos en LDAP se consideran grupos.
- En tu LDAP: (objectClass=posixGroup)

6. Credenciales de Bind (opcional pero recomendado):

- Un usuario LDAP con permisos de lectura para buscar usuarios y grupos.
- Por ejemplo: cn=admin,dc=villegas,dc=basilio

3 Testear la conexión

- OwnCloud tiene un botón **"Test connection"**.
- Si OwnCloud puede conectarse y leer usuarios, la prueba será exitosa.

4 Guardar configuración

- Una vez la prueba funciona, guardas la configuración.
- Ahora todos los usuarios que existen en LDAP pueden iniciar sesión en OwnCloud **con las mismas credenciales** que usan para correo u otros servicios LDAP.

En pocas palabras:

Con esto, OwnCloud “confía” en tu LDAP para autenticar usuarios. No necesitas crear usuarios manualmente en OwnCloud; se sincronizan desde LDAP y pueden iniciar sesión automáticamente.

Si quieres, puedo hacerte un **paso a paso visual y exacto para OwnCloud**, con capturas de configuración simuladas usando tu LDAP `villegas.basilio`, para que sea más fácil de aplicar.

¿Quieres que haga eso?

Tú dijiste:

ahora tengo que hacer esto, como lo hago: Pruebas: • Un usuario de LDAP inicia sesión en Thunderbird/Outlook. • Envía un correo a otro usuario LDAP. • Verificar en logs del servidor que la autenticación se ha hecho vía LDAP.

ChatGPT Plus

¡Perfecto! Vamos a detallar **cómo hacer estas pruebas paso a paso** usando tu configuración LDAP (`villegas.basilio`) y tu servidor de correo Postfix + Dovecot.

1 Preparar un usuario de prueba

Supongamos que vas a usar el usuario villegas (ya existe en tu LDAP):

- Usuario: villegas
 - Correo: villegas@villegas.basilio
 - Contraseña: la que tiene en LDAP
-

2 Configurar Thunderbird o Outlook

Thunderbird (ejemplo)

1. Abrir Thunderbird → **Crear nueva cuenta**.
2. Elegir **Cuenta de correo existente**.
3. Introducir:
 - Nombre: Villegas
 - Correo electrónico: villegas@villegas.basilio
 - Contraseña: la contraseña LDAP
4. Seleccionar **Configuración manual**:
 - **IMAP**:
 - Servidor entrante: mail.villegas.basilio
 - Puerto: 143 (o 993 si usas IMAPS)
 - Seguridad: STARTTLS o SSL/TLS
 - Autenticación: Contraseña normal
 - **SMTP**:
 - Servidor saliente: mail.villegas.basilio
 - Puerto: 25 o 587
 - Seguridad: STARTTLS o SSL/TLS
 - Autenticación: Contraseña normal
5. Guardar y probar conexión.

Si la autenticación falla, revisa que Dovecot esté corriendo y que LDAP tenga el usuario correcto.

3 Enviar un correo a otro usuario LDAP

Supongamos que envías a `basilio@villegas.basilio`:

1. Redacta un correo en Thunderbird:
 - De: `villegas@villegas.basilio`
 - Para: `basilio@villegas.basilio`
 - Asunto: Prueba LDAP
 2. Enviar correo.
-

4 Verificar autenticación vía LDAP en los logs

a. Logs de Dovecot (autenticación)

bash

```
tail -f /var/log/dovecot.log
```

Busca entradas como:

sql

```
imap(villegas): Login: user=<villegas@villegas.basilio>, method=PLAIN,
rip=IP_DEL_CLIENTE, tls=...
```

Si ves Login con el usuario correcto, significa que **Dovecot validó la contraseña contra LDAP**.

b. Logs de Postfix (envío de correo)

bash

```
tail -f /var/log/mail.log
```

Busca algo similar a:

swift

```
postfix/smtpd[PID]: [ID]: client=IP_DEL_CLIENTE, sasl_method=PLAIN,
sasl_username=villegas
```

```
postfix/qmgr[PID]: [ID]: from=<villegas@villegas.basilio>, to=
<basilio@villegas.basilio>, status=sent
```

- `sasl_username=villegas` indica que la autenticación fue hecha vía SASL, que en tu caso está vinculada a LDAP a través de Dovecot.
- `status=sent` confirma que el correo se entregó correctamente.

5 Probar recepción

- Inicia sesión con `basilio@villegas.basilio` en Thunderbird u otro cliente IMAP.
- Verifica que haya llegado el correo de villegas.

✓ Resumen

1. Configuras cliente de correo con usuario LDAP.
2. Inicias sesión → Dovecot autentica contra LDAP.
3. Envías correo → Postfix lo procesa y registra la autenticación SASL (LDAP).
4. Verificas en los logs que `sasl_username` coincida con el usuario LDAP.

Si quieres, puedo prepararte **una guía completa de prueba lista para copiar y pegar**, incluyendo **los comandos exactos de verificación en logs y ejemplos de líneas que deberías ver**, para que tengas evidencia de que la autenticación fue LDAP.

¿Quieres que haga eso?

Tú dijiste:

me dice que se encuentran las configuraciones al sondear el servidor pero al iniciar sesion me pone que la contraseña no es valida la cual estoy al 100% seguro de que es usuario. Por cierto, estoy iniciando con el usuario svb111 que es el unico que me acuerdo de la contraseña si o si

ChatGPT Plus

¡Perfecto! Gracias por la información, eso nos ayuda mucho a diagnosticar. 😊

El hecho de que puedas **“sondear” el servidor LDAP** pero al iniciar sesión te diga **contraseña inválida** normalmente indica un problema de **mapa de atributos, filtros LDAP, o esquema de contraseña** en la configuración de Dovecot/Postfix.

Vamos a analizarlo paso a paso:

1 Verificar el usuario en LDAP

Tu usuario svb111 según tu ldapsearch:

makefile

```
dn: cn=svb111,dc=villegas,dc=basilio
objectClass: posixAccount, inetOrgPerson, organizationalPerson, person
loginShell: /bin/bash
homeDirectory: /home/svb111
uid: svb111
cn: svb111
uidNumber: 10006
gidNumber: 10000
sn: svb111
mail: svb111@gmail.com
homePhone: 123456789
```

Observaciones importantes:

- Este usuario **NO está dentro de ou=usuarios**, sino directamente bajo la base DN: dc=villegas,dc=basilio.
- Tu configuración actual de Dovecot (base = ou=usuarios,dc=villegas,dc=basilio) **no incluirá este usuario**, porque svb111 está fuera de la OU usuarios.
- Además, tiene mail: svb111@gmail.com, no @villegas.basilio, lo que puede afectar la correspondencia del login si usas correo como usuario.

2 Qué hacer en la configuración de Dovecot

a. Ajustar el base para incluir todos los usuarios

En /etc/dovecot/dovecot-ldap.conf.ext, cambia:

bash

```
# Antes
base = ou=usuarios,dc=villegas,dc=basilio

# Ahora
base = dc=villegas,dc=basilio
```

De esta forma, Dovecot buscará **todos los usuarios bajo la base DN**, incluyendo svb111.

b. Revisar user_filter y pass_filter

Actualmente tienes:

bash

```
user_filter = (&(objectClass=posixAccount)(uid=%u))  
pass_filter = (&(objectClass=posixAccount)(uid=%u))
```

- Esto está bien si vas a autenticar con uid (por ejemplo, svb111).
- **No funcionará si intentas usar el correo (svb111@gmail.com)** como usuario, porque el atributo uid es svb111.
- Solución: asegúrate de usar **svb111** como usuario en Thunderbird/Outlook, no el correo.

c. Verificar el esquema de contraseña

- En tu LDAP, es probable que las contraseñas estén almacenadas en **userPassword** con hash tipo {SSHA} o {SHA}.
- Dovecot debe coincidir con esto:

bash

```
default_pass_scheme = SSHA
```

- Si tu LDAP tiene otra codificación, cámbialo acorde.

d. Habilitar auth_bind para pruebas

- auth_bind = yes permite que Dovecot intente **bindear directamente con la contraseña del usuario**, evitando problemas de hash.
- Esto funciona bien para tests rápidos.

3 Verificación rápida desde el servidor

Prueba autenticar directamente con LDAP:

bash

```
ldapwhoami -x -D "cn=svb111,dc=villegas,dc=basilio" -W
```

- Te pedirá la contraseña de svb111.
- Si funciona, el problema no es la contraseña, sino la configuración de Dovecot.

4 Ajuste final sugerido de dovecot - ldap.conf.ext

bash

```
hosts = ldap.villegas.basilio
dn = cn=admin,dc=villegas,dc=basilio
dnpass = TU_CONTRASEÑA_ADMIN_LDAP
base = dc=villegas,dc=basilio
ldap_version = 3
scope = subtree
auth_bind = yes

user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))

user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail
default_pass_scheme = SSHA
```

Luego recarga Dovecot:

bash

```
systemctl restart dovecot
```

5 Reintentar login

- En Thunderbird/Outlook usa:
 - Usuario: svb111
 - Contraseña: la misma que recuerdas

Debería funcionar.

Si quieres, puedo **hacer una versión completa de todos los archivos de Postfix y Dovecot lista para tu LDAP**, teniendo en cuenta usuarios como svb111 que no están dentro de ou=usuarios, para que funcione con cualquier usuario real que tengas.

¿Quieres que haga eso?

Tú dijiste:

dame los pasos y configuraciones para cambiar todas las contraseñas de mis usuarios de ldap a usuario

¡Perfecto! Vamos a hacer esto paso a paso. 😎

Objetivo: **Cambiar la contraseña de todos los usuarios de tu LDAP a la misma**, por ejemplo usuario, de forma segura.

Vamos a asumir que estás conectado como root y que tu LDAP es **villegas.basilio**.

1 Verificar la ubicación de los usuarios

Según tu `ldapsearch`, tus usuarios están en:

- `ou=usuarios,dc=villegas,dc=basilio` → mayoría de usuarios
- Algunos usuarios fuera de la OU, por ejemplo `svb111,dc=villegas,dc=basilio`

Por eso vamos a hacer un cambio **para todos los usuarios bajo la base DN `dc=villegas,dc=basilio`**.

2 Preparar la contraseña en formato LDAP

LDAP normalmente usa el atributo `userPassword`. Para guardar una contraseña en texto claro de forma segura, podemos convertirla a **SSHA** (Dovecot/Postfix soporta SSHA).

Generar contraseña SSHA desde línea de comandos:

bash

```
slappasswd -s usuario
```

Esto te dará algo como:

```
{SSHA}V1H2lK7v9Jkq2fN8G3tH1kZ5xB1u4pQ=
```

- Esta será la contraseña `usuario` en formato LDAP.
-

3 Crear un archivo LDIF para modificar los usuarios

Ejemplo: `cambiar_contrasenas.ldif`

ldif

```
dn: ou=usuarios,dc=villegas,dc=basilio
changetype: modify
replace: userPassword
userPassword: {SSHA}V1H2lK7v9Jkq2fN8G3tH1kZ5xB1u4pQ=
```

Pero este solo cambiaría **la OU completa**.

Para cambiar cada usuario individualmente, lo mejor es **hacer un script que genere un LDIF por cada usuario**.

4 Script para cambiar la contraseña de todos los usuarios

Supongamos que quieres cambiar la contraseña de todos los usuarios de `dc=villegas,dc=basilio`:

bash

```
#!/bin/bash

# Contraseña nueva en texto claro
NUEVA_PASS="usuario"

# Genera el hash SSHA
HASH=$(slappasswd -s "$NUEVA_PASS")

# Busca todos los usuarios (posixAccount) en LDAP
ldapsearch -x -LLL -b "dc=villegas,dc=basilio" "(objectClass=posixAccount)"
dn | while read DN
do
    if [[ "$DN" == dn:* ]]; then
        USUARIO_DN=$(echo "$DN" | cut -d' ' -f2-)
        echo "Cambiando contraseña de $USUARIO_DN"
        # Crear LDIF temporal
        echo "dn: $USUARIO_DN
changetype: modify
replace: userPassword
userPassword: $HASH
" | ldapmodify -x -D "cn=admin,dc=villegas,dc=basilio" -w
TU_CONTRASEÑA_ADMIN_LDAP
    fi
done
```

- Reemplaza `TU_CONTRASEÑA_ADMIN_LDAP` con la contraseña del admin LDAP.
- Esto iterará sobre **todos los usuarios posixAccount** y cambiará la contraseña a `usuario`.

5 Probar que se aplicó

Prueba autenticar con alguno de los usuarios:

bash

```
ldapwhoami -x -D "cn=svb111,dc=villegas,dc=basilio" -w usuario
```

Si responde con el DN del usuario, la contraseña fue cambiada correctamente.

6 Notas importantes

- **Backup primero:** Siempre haz un backup de LDAP antes de cambiar contraseñas:

bash

```
slapcat -b "dc=villegas,dc=basilio" > backup.ldif
```

- Este procedimiento cambia todas las contraseñas a la misma (usuario).
 - Para seguridad en producción, lo ideal es generar contraseñas individuales aleatorias o temporales y obligar al cambio en el primer login.
-

Si quieres, puedo **prepararte un script listo para copiar y pegar**, que cambie **todas las contraseñas de tus usuarios LDAP a usuario**, incluyendo los usuarios que no están en ou=usuarios (como svb111), de forma automática.

¿Quieres que haga eso?

Tú dijiste:

no me hagas scripts ni nada, dame el código completo para el archivo ldif para todos los usuarios que los sabes. Esta es la contraseña encriptada así que cámbialo:

```
{SSHA}Z+z0eBs6U/c7d/LyWwjTSokzPCnsOLo6
```

ChatGPT Plus

Aquí tienes **el contenido completo** del archivo LDIF con las entradas modify para **todos los usuarios** que aparecen en tu ldapsearch y que identificamos como cuentas de usuario. Sustituye/usa tal cual:

ldif

```
dn: cn=villegas,ou=usuarios,dc=villegas,dc=basilio
changetype: modify
replace: userPassword
userPassword: {SSHA}Z+z0eBs6U/c7d/LyWwjTSokzPCnsOLo6

dn: cn=912345678,ou=usuarios,dc=villegas,dc=basilio
changetype: modify
replace: userPassword
userPassword: {SSHA}Z+z0eBs6U/c7d/LyWwjTSokzPCnsOLo6
```

```
dn: cn=sergio,ou=usuarios,dc=villegas,dc=basilio
changetype: modify
replace: userPassword
userPassword: {SSHA}Z+z0eBs6U/c7d/LyWwjTSokzPCns0Lo6

dn: cn=svb,ou=usuarios,dc=villegas,dc=basilio
changetype: modify
replace: userPassword
userPassword: {SSHA}Z+z0eBs6U/c7d/LyWwjTSokzPCns0Lo6

dn: cn=svb111,dc=villegas,dc=basilio
changetype: modify
replace: userPassword
userPassword: {SSHA}Z+z0eBs6U/c7d/LyWwjTSokzPCns0Lo6
```

Guarda este contenido en un fichero (por ejemplo `cambiar_contrasenass.ldif`) y aplícalo con `ldapmodify` desde tu servidor LDAP cuando quieras. Si prefieres que te indique la línea exacta de comando para aplicarlo, dímelo y la incluyo.

Tú dijiste:

como es el comando de `ldapmodify` completo

ChatGPT Plus

El comando completo para aplicar un archivo LDIF usando `ldapmodify` es el siguiente:

bash

```
ldapmodify -x -D "cn=admin,dc=villegas,dc=basilio" -w
TU_CONTRASEÑA_ADMIN_LDAP -f cambiar_contrasenass.ldif
```

Explicación:

- `-x` → Usa autenticación simple (sin SASL).
- `-D "cn=admin,dc=villegas,dc=basilio"` → DN del usuario administrador de LDAP que tiene permisos para modificar contraseñas.
- `-w TU_CONTRASEÑA_ADMIN_LDAP` → La contraseña del administrador LDAP.
- `-f cambiar_contrasenass.ldif` → Archivo LDIF que contiene las modificaciones (el que creaste con las nuevas contraseñas).

 **Tips importantes antes de ejecutar:**

1. Haz **backup de LDAP** antes de modificar nada:

bash

```
slapcat -b "dc=villegas,dc=basilio" > backup.ldif
```

2. Revisa que el LDIF no tenga errores de sintaxis.

3. Una vez aplicado, prueba un login con alguno de los usuarios:

bash

```
ldapwhoami -x -D "cn=svb111,dc=villegas,dc=basilio" -w usuario
```

Si responde con el DN del usuario, la contraseña se cambió correctamente.

Si quieres, puedo escribir **el comando exacto listo para copiar y pegar con tu usuario admin y el LDIF que generamos**.

¿Quieres que lo haga?

Tú dijiste:

me dice eso root@LDAP-Ubu:~# ldapmodify ldapmodify -x -D "cn=admin,dc=villegas,dc=basilio" -w usuario -f cambiar_contrasenas.ldif Add or modify entries from an LDAP server usage: ldapmodify [options] The list of desired operations are read from stdin or from the file specified by "-f file". Add or modify options:

ChatGPT Plus

Ah, ya veo el problema 😊. El error se debe a que escribiste **ldapmodify ldapmodify** al inicio, eso no es correcto. Solo necesitas **un ldapmodify** al inicio del comando.

El comando correcto sería:

bash

```
ldapmodify -x -D "cn=admin,dc=villegas,dc=basilio" -w usuario -f  
cambiar_contrasenas.ldif
```

Explicación rápida:

- **ldapmodify** → llama al programa que aplica cambios LDAP.
- **-x** → autenticación simple (no SASL).
- **-D "cn=admin,dc=villegas,dc=basilio"** → usuario administrador LDAP.

- `-w usuario` → contraseña del admin LDAP (en este ejemplo, es “usuario”).
- `-f cambiar_contraseñas.ldif` → tu archivo LDIF con las modificaciones.

No debe haber un segundo `ldapmodify` al principio.

Si quieres, puedo **escribirte el comando exacto listo para copiar y pegar con tu LDIF y usuario admin**, para que no haya errores de sintaxis.

¿Quieres que haga eso?

Tú dijiste:

ahora tengo todas las contraseñas bien y estoy intentando iniciar sesión con el usuario villegas, el cual no me deja

ChatGPT Plus

¡Perfecto! Entonces ya sabemos que **las contraseñas se cambiaron correctamente**, pero ahora no puedes iniciar sesión con el usuario villegas. 😊 Esto normalmente tiene que ver con **cómo Dovecot/Postfix está mapeando el usuario y dónde lo busca en LDAP**. Vamos a diagnosticar paso a paso.

1 Verificar que el usuario existe y su DN

Según tu `ldapsearch`:

`makefile`

```
dn: cn=villegas,ou=usuarios,dc=villegas,dc=basilio
uid: villegas
mail: villegas@villegas.basilio
```

- El usuario está **dentro de ou=usuarios**.
- El atributo de login es `uid = villegas`.

💡 Esto significa que **debes usar villegas como usuario en el cliente de correo**, no el correo `villegas@villegas.basilio`, a menos que hayas configurado Dovecot para buscar por mail.

2 Verificar la configuración de Dovecot

Archivo `/etc/dovecot/dovecot-ldap.conf.ext`:

`bash`


```
hosts = ldap.villegas.basilio
dn = cn=admin,dc=villegas,dc=basilio
dnpass = TU_CONTRASEÑA_ADMIN_LDAP
base = ou=usuarios,dc=villegas,dc=basilio
ldap_version = 3
scope = subtree
auth_bind = yes

user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))

user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail
default_pass_scheme = SSHA
```

Puntos críticos:

1. base = ou=usuarios, ... → correcto, incluye al usuario villegas.
2. user_filter = (&(objectClass=posixAccount)(uid=%u)) → busca por uid.
 - Entonces, en el cliente de correo **debes poner villegas como usuario**, no villegas@villegas.basilio.
3. auth_bind = yes → Dovecot intentará bindear directamente con la contraseña del usuario, lo que es correcto.

3 Probar autenticación desde la línea de comandos

Esto nos permite verificar si Dovecot puede autenticar a villegas:

bash

```
dovecot -n
```

- Asegúrate de que auth-ldap está cargado.

Luego, prueba directamente con:

bash

```
ldapwhoami -x -D "cn=villegas,ou=usuarios,dc=villegas,dc=basilio" -w
usuario
```

- Debe responder: dn: cn=villegas,ou=usuarios,dc=villegas,dc=basilio
- Si falla aquí, el problema está **en LDAP**, no en Dovecot.

4 Revisar logs de Dovecot al intentar login

Mientras intentas iniciar sesión desde Thunderbird/Outlook:

bash

```
tail -f /var/log/dovecot.log
```

Busca líneas como:

pgsql

```
imap(villegas): Login failed: user=<villegas>, method=PLAIN,  
rip=IP_DEL_CLIENTE, TLS, unknown user
```

- Si dice `unknown user` → Dovecot **no puede encontrar el DN del usuario**
- Si dice `Password mismatch` → Dovecot **no está verificando la contraseña correctamente**

5 Ajustes comunes

1. Si quieres usar el correo como login (`villegas@villegas.basilio`):

- Cambia `user_filter` y `pass_filter`:

bash

```
user_filter = (&(objectClass=posixAccount)(mail=%u))  
pass_filter = (&(objectClass=posixAccount)(mail=%u))
```

2. Si usas **UID como login** (más seguro y recomendable):

- En el cliente de correo, pon **usuario = villegas**, contraseña = usuario

3. Recarga Dovecot después de cualquier cambio:

bash

```
systemctl restart dovecot
```

Si quieres, puedo **prepararte la configuración de Dovecot lista para usar con todos tus usuarios**, que funcione tanto con UID como con correo, y así no tendrás problemas de login.

¿Quieres que haga eso?

Tú dijiste:

```

root@LDAP-Ubu:~# dovecot -n # 2.3.21 (47349e2482): /etc/dovecot/dovecot.conf # Pigeonhole version
0.5.21 (f6cd4b8e) # OS: Linux 6.14.11-1-pve x86_64 Ubuntu 24.04.3 LTS # Hostname: LDAP-
Ubu.institutodh.net auth_mechanisms = plain login disable_plaintext_auth = no mail_location =
mbox:~/mail:INBOX=/var/mail/%u mail_privileged_group = mail 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 } passdb { args = /etc/dovecot/dovecot-ldap.conf.ext driver = ldap } protocols = "
imap pop3" ssl_cert = </etc/dovecot/private/dovecot.pem ssl_client_ca_dir = /etc/ssl/certs ssl_dh = #
hidden, use -P to show it ssl_key = # hidden, use -P to show it userdb { driver = passwd } userdb { args =
/etc/dovecot/dovecot-ldap.conf.ext driver = ldap } root@LDAP-Ubu:~# ldapwhoami -x -D
"cn=villegas,ou=usuarios,dc=villegas,dc=basilio" -w usuario
dn:cn=villegas,ou=usuarios,dc=villegas,dc=basilio root@LDAP-Ubu:~# root@LDAP-Ubu:~# tail -f
/var/log/dovecot.log tail: cannot open '/var/log/dovecot.log' for reading: No such file or directory tail: no
files remaining root@LDAP-Ubu:~#

```

ChatGPT Plus

¡Perfecto! Con lo que muestras podemos diagnosticar varias cosas:

1. LDAP funciona correctamente:

bash

```

ldapwhoami -x -D "cn=villegas,ou=usuarios,dc=villegas,dc=basilio" -w
usuario

```

- Devuelve el DN del usuario → la contraseña es correcta y LDAP responde.

2. Dovecot tiene configuraciones que debemos ajustar:

En tu salida de dovecot -n:

text

```

passdb {
    driver = pam
}
passdb {
    args = /etc/dovecot/dovecot-ldap.conf.ext
    driver = ldap
}
userdb {
    driver = passwd
}
userdb {
    args = /etc/dovecot/dovecot-ldap.conf.ext
}

```

```
driver = ldap
}
```

Observaciones críticas:

- Tienes **dos passdb**: uno usando pam y otro LDAP.
 - Dovecot primero intenta la primera passdb que encuentra.
 - Por eso puede que esté intentando pam y fallando antes de consultar LDAP.
- Tienes **dos userdb**: uno passwd y otro LDAP.
 - Esto puede generar conflictos si passwd no tiene el usuario villegas.
- Mail location:

text

```
mail_location = mbox:~/mail:INBOX=/var/mail/%u
```

- Esto requiere que exista /home/villegas y que Dovecot pueda acceder al buzón /var/mail/villegas.

3 Ajustes recomendados para que funcione LDAP correctamente

a) Priorizar LDAP sobre PAM

Edita /etc/dovecot/conf.d/10-auth.conf (o tu dovecot-ldap.conf.ext si lo incluyes directamente):

bash

```
!include auth-ldap.conf.ext
# comentamos PAM si no lo necesitamos
#passdb {
#  driver = pam
#}
```

O en dovecot -n debe aparecer **solo LDAP** como passdb.

b) Userdb también solo LDAP

- Comenta userdb { driver = passwd } si no necesitas usuarios del sistema local.
- Dovecot entonces usará solo LDAP para encontrar home, uid, gid y mail.

c) Asegurarte de que los buzones existen

- La ruta de mail: `mbox:~/mail:INBOX=/var/mail/%u`
- Asegúrate de que `/var/mail/villegas` existe y tiene permisos correctos:

bash

```
ls -l /var/mail/
```

Si no existe:

bash

```
touch /var/mail/villegas
chown villegas:mail /var/mail/villegas
chmod 660 /var/mail/villegas
```

Si quieres usar Maildir en lugar de Mbox, podemos cambiar `mail_location` a Maildir (recomendado).

d) Logs de Dovecot

- En Ubuntu 24.04, Dovecot no siempre usa `/var/log/dovecot.log`.
- Puedes revisar los logs por `journalctl`:

bash

```
journalctl -u dovecot -f
```

Esto te mostrará **todos los intentos de login** y errores.

4 Próximo paso recomendado

1. Comenta `passdb { driver = pam }` y `userdb { driver = passwd }`.
2. Asegúrate que `passdb` y `userdb` LDAP apunten a tu `dovecot-ldap.conf.ext`.
3. Reinicia Dovecot:

bash

```
systemctl restart dovecot
```

4. Verifica login con:

bash

```
doveadm auth test villegas
```

- Esto probará la autenticación LDAP directamente desde Dovecot.
 - Debe devolver `passdb: ldap (user villegas): successful`.
-

Si quieres, puedo **escribirte la configuración completa de Dovecot lista para tu LDAP**, con LDAP como único `passdb` y `userdb`, `mailboxes` correctos y `logs` funcionando en `journalctl`, lista para copiar y pegar.

¿Quieres que haga eso?

Tú dijiste:

a ver otra de las cosas que me pasan que igual es algo mas simple es que cuando pongo el correo me envia al dominio `villegas.basilio` pero no me da conectividad. cuando cambio el apartado nombre del servidor por `10.2.8.200` me conecta pero no me funcionan las contraseñas, no se si tiene que ver solo te lo comento

ChatGPT Plus

Ah, eso me da **mucha información útil**. 🧐 Vamos a analizarlo paso a paso.

1 Lo que está pasando con el servidor

- Cuando pones `mail.villegas.basilio` en el cliente de correo, no se conecta.
- Cuando pones la **IP** (`10.2.8.200`), sí conecta, pero las contraseñas no funcionan.

Esto indica dos cosas distintas:

A) Problema de resolución DNS

- `mail.villegas.basilio` **no se resuelve a la IP correcta** en tu red.
- El cliente de correo intenta conectar a un hostname que no existe o no apunta a tu servidor LDAP/IMAP.
- Solución:
 1. Asegúrate de que `mail.villegas.basilio` tenga un **registro A** en tu DNS apuntando a `10.2.8.200`.
 2. Como solución temporal, puedes añadir en cada cliente de correo en su archivo `hosts`:

```
10.2.8.200    mail.villegas.basilio
```

3. Luego prueba de nuevo con `mail.villegas.basilio` como servidor.
-

B) Contraseñas no funcionan con la IP

- Esto normalmente ocurre porque Dovecot **hace validación de SASL/LDAP usando el hostname que recibe en el TLS**.
 - Si conectas por IP, Dovecot puede rechazar la autenticación si:
 - Estás usando **SSL/TLS** y el certificado no coincide con la IP.
 - Estás usando autenticación LDAP y Dovecot hace algún check sobre el hostname.
 - Por eso, con la IP funciona la conexión TCP, pero **LDAP no autentica correctamente**.
-

C) Cómo solucionarlo

1. **Usa siempre el hostname en clientes de correo:** `mail.villegas.basilio`
 - Asegúrate de que resuelva a tu IP: `ping mail.villegas.basilio → 10.2.8.200`
 2. **Certificados SSL/TLS:**
 - Si tienes SSL, el certificado de Dovecot debe tener **como CN o SAN** `mail.villegas.basilio`, no la IP.
 3. **LDAP y Dovecot:**
 - No importa la IP en sí, sino que el cliente se conecte usando el hostname correcto si TLS está habilitado.
 - Además, asegúrate de que la configuración `hosts = ldap.villegas.basilio` en `dovecot-ldap.conf.ext` coincida con lo que Dovecot puede resolver desde el servidor.
-

D) Verificación práctica

Desde el servidor de correo:

bash

```
ping mail.villegas.basilio
```

Desde el cliente de correo:

- Servidor: `mail.villegas.basilio`
- Puerto IMAP: 143 (STARTTLS) o 993 (SSL)
- Usuario: `villegas`
- Contraseña: `usuario`

Si no resuelve el hostname, agrega el registro a `/etc/hosts` temporalmente.

💡 Resumen rápido:

- Con la IP conectas, pero TLS/LDAP fallan → no se recomienda.
 - Con el hostname correcto y DNS funcionando, todo debería funcionar: conexión y autenticación LDAP.
-

Si quieres, puedo hacerte un **diagrama de la configuración final de correo y LDAP**, incluyendo hostnames, IPs, TLS y usuario, para que tengas una referencia clara y segura de cómo debe estar configurado todo.

¿Quieres que haga eso?

Tú dijiste:

no me sale mail.villegas.basilio, me sale villegas.basilio. como cambio el archivo de hosts

ChatGPT Plus

¡Ah, perfecto! Eso explica por qué tu cliente de correo no se conecta correctamente. 😊

Actualmente tu dominio es villegas.basilio, pero quieres que **mail.villegas.basilio apunte a tu servidor**. Esto se puede hacer fácilmente editando el archivo `hosts` en tu servidor o en la máquina cliente.

1 Editar `/etc/hosts` en Linux

1. Abre el archivo `/etc/hosts` con tu editor favorito (como root):

bash

```
nano /etc/hosts
```

2. Añade una línea al final para mapear el hostname al IP de tu servidor:

```
10.2.8.200    mail.villegas.basilio
```

- `10.2.8.200` → IP de tu servidor LDAP/IMAP/Postfix
- `mail.villegas.basilio` → nombre que quieres usar en clientes de correo

3. Guarda y cierra el archivo (`Ctrl+O` → `Enter` → `Ctrl+X` en nano).

4. Prueba que el hostname resuelve:

bash

```
ping mail.villegas.basilio
```

- Debe responder con 10.2.8.200.

2 Editar /etc/hosts en Windows (cliente de correo)

1. Abre **Bloc de notas** como administrador.
2. Edita el archivo:

makefile

```
C:\Windows\System32\drivers\etc\hosts
```

3. Añade la misma línea:

```
10.2.8.200    mail.villegas.basilio
```

4. Guarda y cierra el archivo.
5. Prueba en el cliente:

cmd

```
ping mail.villegas.basilio
```

3 Configurar cliente de correo

- Servidor de entrada (IMAP/POP3): mail.villegas.basilio
- Servidor de salida (SMTP): mail.villegas.basilio
- Usuario: villegas
- Contraseña: usuario

Ahora, el cliente debe poder **conectarse correctamente y autenticar contra LDAP**.

Si quieres, puedo explicarte **cómo configurar Dovecot y Postfix para que funcionen correctamente con mail.villegas.basilio**, incluyendo TLS y autenticación LDAP, de manera que no tengas problemas aunque uses certificados SSL.

¿Quieres que haga eso?

Tú dijiste:

vale acabo de hacer los archivos de hosts, me estoy referenciando con esa ip y estamos en la misma situacion de antes, me conecta pero no me funcionan las contraseñas

ChatGPT Plus

¡Ah, perfecto! Ahora entiendo exactamente la situación. 😊

Tienes **conectividad** con `mail.villegas.basilio` o la IP, pero **la autenticación falla** aunque LDAP funciona bien (probado con `ldapwhoami`). Esto nos dice claramente que el **problema no es la conexión**, sino **cómo Dovecot está consultando LDAP**.

Vamos a revisar los puntos críticos:

1 Prioridad de passdb y userdb en Dovecot

Tu salida de `dovecot -n` mostraba:

text

```
passdb {  
    driver = pam  
}  
passdb {  
    args = /etc/dovecot/dovecot-ldap.conf.ext  
    driver = ldap  
}  
  
userdb {  
    driver = passwd  
}  
userdb {  
    args = /etc/dovecot/dovecot-ldap.conf.ext  
    driver = ldap  
}
```

⚠ Problema: **Dovecot usa primero PAM y Passwd**, antes de LDAP.

- Esto hace que cualquier login intente **PAM o passwd local** y falle, **sin llegar a LDAP**.
- Debes **comentar** los `passdb` y `userdb` locales para que **solo LDAP sea usado**.

2 Configuración correcta de Dovecot

Tu `/etc/dovecot/conf.d/10-auth.conf` debería verse así:

text

```
disable_plaintext_auth = no
auth_mechanisms = plain login

!include auth-ldap.conf.ext
# Comentar o eliminar cualquier otra passdb/userdb local
# passdb { driver = pam }
# userdb { driver = passwd }
```

Y en /etc/dovecot/dovecot-ldap.conf.ext:

text

```
hosts = ldap.villegas.basilio
dn = cn=admin,dc=villegas,dc=basilio
dnpass = TU_CONTRASEÑA_ADMIN_LDAP
base = ou=usuarios,dc=villegas,dc=basilio
ldap_version = 3
scope = subtree
auth_bind = yes

user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))

user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail
default_pass_scheme = SSHA
```

3 Verificar autenticación directamente con Dovecot

Después de reiniciar Dovecot:

bash

```
systemctl restart dovecot
```

Prueba con:

bash

```
doveadm auth test villegas
```

- Esto hará que Dovecot intente autenticar villegas usando **exactamente lo mismo que usa IMAP/POP3**.
- Debe devolver:

makefile

```
passdb: ldap (user villegas): successful
```

Si falla aquí, el problema está **en la configuración LDAP de Dovecot**.

4 Logs de Dovecot

En Ubuntu 24.04 los logs están en `journalctl`:

bash

```
journalctl -u dovecot -f
```

Intenta login desde Thunderbird y revisa la salida.

- `unknown user` → Dovecot **no encuentra al usuario**.
 - `Password mismatch` → la contraseña LDAP **no coincide con lo que Dovecot espera**.
-

5 Puntos a revisar si sigue fallando

1. base en `dovecot-ldap.conf.ext` → Debe incluir el DN correcto (`ou=usuarios,...`).
 2. `user_filter` y `pass_filter` → Si usas **uid como login**, `%u` debe ser `villegas`.
 3. `auth_bind = yes` → permite que Dovecot haga bind directamente con la contraseña del usuario.
 4. `default_pass_scheme = SSHA` → debe coincidir con el formato del password LDAP.
-

Resumen:

- Conectividad = correcta (IP y hostname funcionan).
 - LDAP funciona (`ldapwhoami`).
 - Fallo = **Dovecot intenta PAM/Passwd primero y no llega a LDAP**, o hay **desajuste entre UID y login**.
-

Si quieres, puedo **escribirte la configuración completa de Dovecot lista para tu LDAP**, comentando PAM/passwd, con userdb/passdb LDAP correctamente configurados, y lista para reiniciar y probar `villegas` sin errores. Esto resolvería el problema de forma definitiva.

¿Quieres que haga eso?

Tú dijiste:

```
root@LDAP-Ubu:/home# doveadm auth test villegas Password: passdb: villegas auth failed extra fields:
user=villegas code=temp_fail root@LDAP-Ubu:/home# enseñame que es lo que tengo que comentar del
apartado "Prioridad de passdb y userdb en Dovecot" root@LDAP-Ubu:/home# journalctl -u dovecot -f Oct
21 11:14:25 LDAP-Ubu dovecot[4795]: imap-login: Disconnected: Connection closed (auth service
reported temporary failure): user=<villegas@villegas.basilio>, method=PLAIN, rip=10.2.8.108,
lip=10.2.8.200, session=<30ZITKIBOIAKAghs> Oct 21 11:16:29 LDAP-Ubu auth[4932]:
ldapdb_canonuser_plug_init() failed in sasl_canonuser_add_plugin(): invalid parameter supplied Oct 21
11:16:29 LDAP-Ubu auth[4932]: _sasl_plugin_load failed on sasl_canonuser_init for plugin: ldapdb Oct
21 11:16:29 LDAP-Ubu dovecot[4795]: auth: Error: ldap(/etc/dovecot/dovecot-ldap.conf.ext): Can't
connect to server: ldap.villegas.basilio Oct 21 11:16:29 LDAP-Ubu auth[4933]:
ldapdb_canonuser_plug_init() failed in sasl_canonuser_add_plugin(): invalid parameter supplied Oct 21
11:16:29 LDAP-Ubu auth[4933]: _sasl_plugin_load failed on sasl_canonuser_init for plugin: ldapdb Oct
21 11:16:29 LDAP-Ubu dovecot[4795]: auth-worker(4933): Error: ldap(/etc/dovecot/dovecot-
ldap.conf.ext): Can't connect to server: ldap.villegas.basilio Oct 21 11:16:29 LDAP-Ubu auth[4933]:
pam_unix(dovecot:auth): check pass; user unknown Oct 21 11:16:29 LDAP-Ubu auth[4933]:
pam_unix(dovecot:auth): authentication failure; logname= uid=0 euid=0 tty=dovecot ruser=villegas rhost=
Oct 21 11:16:31 LDAP-Ubu dovecot[4795]: auth: Error: ldap(/etc/dovecot/dovecot-ldap.conf.ext): Can't
connect to server: ldap.villegas.basilio
```

ChatGPT Plus

Perfecto, con los logs y el resultado de `doveadm auth test` queda claro qué está pasando:

1. **Dovecot no puede conectarse a `ldap.villegas.basilio`** → Can't connect to server: `ldap.villegas.basilio`.
2. Mientras tanto, intenta `pam_unix` y falla porque el usuario `villegas` **no existe en el sistema local**.

Esto coincide con el síntoma: **conectas a la IP y al hostname, pero la autenticación LDAP no funciona**.

Paso 1: Prioridad de passdb y userdb

Actualmente en tu `dovecot -n` aparece:

text

```
passdb {
    driver = pam
}
passdb {
    args = /etc/dovecot/dovecot-ldap.conf.ext
    driver = ldap
}
```

```
userdb {  
    driver = passwd  
}  
userdb {  
    args = /etc/dovecot/dovecot-ldap.conf.ext  
    driver = ldap  
}
```

Qué hay que hacer:

1. **Comentar los passdb y userdb locales** (pam y passwd) para que **solo LDAP sea usado**.
2. El resultado debería quedar así:

text

```
# passdb {  
#    driver = pam  
# }  
passdb {  
    args = /etc/dovecot/dovecot-ldap.conf.ext  
    driver = ldap  
}  
  
# userdb {  
#    driver = passwd  
# }  
userdb {  
    args = /etc/dovecot/dovecot-ldap.conf.ext  
    driver = ldap  
}
```

Esto se hace en los archivos de configuración de Dovecot, normalmente en `/etc/dovecot/conf.d/10-auth.conf` o directamente en `dovecot.conf`.

Paso 2: Verificar que Dovecot puede resolver LDAP

El error principal ahora es:

vbnet

```
Error: ldap(/etc/dovecot/dovecot-ldap.conf.ext): Can't connect to server:  
ldap.villegas.basilio
```

Esto significa que **Dovecot no puede resolver el hostname LDAP** o no puede conectarse al puerto 389.

Soluciones:

1. Probar conectividad desde el servidor Dovecot:

bash

```
ping ldap.villegas.basilio
telnet ldap.villegas.basilio 389
```

- Si no responde → Dovecot no puede conectarse al hostname.
- Como solución rápida puedes usar la **IP del servidor LDAP** en `dovecot-ldap.conf.ext`:

text

```
hosts = 10.2.8.200
```

2. Asegúrate de que el firewall permita conexión al puerto 389 (LDAP).

Paso 3: Reiniciar Dovecot y probar

1. Reinicia Dovecot:

bash

```
systemctl restart dovecot
```

2. Prueba la autenticación con:

bash

```
doveadm auth test villegas
```

- Si devuelve `passdb: ldap (user villegas): successful` → todo listo.
- Si sigue fallando → revisa que `hosts` apunte a algo que Dovecot pueda resolver y que el puerto 389 esté abierto.

Resumen

1. Comentar `passdb driver=pam` y `userdb driver=passwd` para que solo LDAP sea usado.
2. En `dovecot-ldap.conf.ext`, asegurarte que `hosts` apunta a algo resolvable (IP o hostname válido).

3. Reiniciar Dovecot y probar con `doveadm auth test`.

Si quieres, puedo escribir **el archivo `dovecot-ldap.conf.ext` completo listo para tu LDAP**, con IP que funciona, passdb/userdb LDAP único, y listo para reiniciar y probar villegas.

¿Quieres que haga eso?

Tú dijiste:

no encuentro el archivo para Paso 1: Prioridad de passdb y userdb

ChatGPT Plus

¡Perfecto! No te preocupes, te guío paso a paso para localizarlo y modificarlo. 😊

1 Archivos donde se define passdb/userdb

En Dovecot, la configuración de autenticación suele estar repartida en:

1. **`/etc/dovecot/dovecot.conf`** → archivo principal, puede incluir otros archivos.
2. **`/etc/dovecot/conf.d/10-auth.conf`** → normalmente aquí están definidas las secciones passdb y userdb.
3. **`/etc/dovecot/conf.d/auth-ldap.conf.ext`** → solo contiene la configuración LDAP (host, base DN, filtros).

Lo que viste con `dovecot -n` muestra **la configuración final combinada** de todos estos archivos.

2 Cómo encontrar los passdb y userdb

Ejecuta este comando:

bash

```
grep -R "passdb" /etc/dovecot/  
grep -R "userdb" /etc/dovecot/
```

- Esto te mostrará **en qué archivos están definidos**.
- Probablemente te saldrá algo como:

swift


```
/etc/dovecot/conf.d/10-auth.conf:passdb { driver = pam }  
/etc/dovecot/conf.d/10-auth.conf:passdb { args = /etc/dovecot/dovecot-  
ldap.conf.ext driver = ldap }  
/etc/dovecot/conf.d/10-auth.conf:userdb { driver = passwd }  
/etc/dovecot/conf.d/10-auth.conf:userdb { args = /etc/dovecot/dovecot-  
ldap.conf.ext driver = ldap }
```

3 Qué editar

1. Abre `/etc/dovecot/conf.d/10-auth.conf`:

bash

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

2. Comenta las secciones locales (pam y passwd) y deja solo LDAP:

text

```
# passdb {  
#   driver = pam  
# }  
passdb {  
    args = /etc/dovecot/dovecot-ldap.conf.ext  
    driver = ldap  
}  
  
# userdb {  
#   driver = passwd  
# }  
userdb {  
    args = /etc/dovecot/dovecot-ldap.conf.ext  
    driver = ldap  
}
```

3. Guarda (Ctrl+O) y cierra (Ctrl+X).
-

4 Reiniciar Dovecot

bash

```
systemctl restart dovecot
```

5 Probar autenticación

bash

```
doveadm auth test villegas
```

- Si devuelve `passdb: ldap (user villegas): successful` → problema resuelto.

Si quieres, puedo **escribirte paso a paso un comando que haga el backup de 10-auth.conf, comente PAM/Passwd y deje solo LDAP**, para que no tengas que buscarlo a mano.

¿Quieres que haga eso?

Tú dijiste:

me sale eso, sigo sin encontrar el archivo

ChatGPT Plus

¡Ah, perfecto! Ahora entiendo: en tu sistema **probablemente no existe un 10-auth.conf editable**, o Dovecot está usando la configuración predeterminada combinada en `/etc/dovecot/dovecot.conf` y los archivos incluidos desde `/etc/dovecot/conf.d/`.

No te preocupes, hay otra manera de solucionarlo. 😎

1 Buscar exactamente dónde están definidos passdb y userdb

Ejecuta:

bash

```
grep -R "driver = " /etc/dovecot/
```

Esto te mostrará **todos los lugares donde se definen passdb y userdb**, por ejemplo:

swift

```
/etc/dovecot/dovecot.conf:passdb { driver = pam }  
/etc/dovecot/dovecot.conf:passdb { args = /
```

Tú dijiste:

```
root@LDAP-Ubu:/home# grep -R "passdb" /etc/dovecot/ grep -R "userdb" /etc/dovecot/  
/etc/dovecot/dovecot-ldap.conf.ext:# This file is commonly accessed via passdb {} or userdb {} section in  
/etc/dovecot/dovecot-ldap.conf.ext:# If you wish to avoid two LDAP lookups (passdb + userdb), you can
```

use /etc/dovecot/dovecot-sql.conf.ext:# This file is commonly accessed via passdb {} or userdb {} section in /etc/dovecot/dovecot-sql.conf.ext:# For the sql passdb module, you'll need a database with a table that /etc/dovecot/dovecot-sql.conf.ext:# passdb query to retrieve the password. It can return fields:

/etc/dovecot/dovecot-sql.conf.ext:# If you wish to avoid two SQL lookups (passdb + userdb), you can use /etc/dovecot/conf.d/10-auth.conf:# blocking passdb and userdb queries (eg. MySQL and PAM). They're /etc/dovecot/conf.d/10-auth.conf:# You can have multiple passdbs and userdbs. This is useful if you want to /etc/dovecot/conf.d/auth-passwdfile.conf.ext:passdb { /etc/dovecot/conf.d/auth-deny.conf.ext:# Users can be (temporarily) disabled by adding a passdb with deny=yes. /etc/dovecot/conf.d/auth-deny.conf.ext:# The deny passdb should always be specified before others, so it gets /etc/dovecot/conf.d/auth-deny.conf.ext:# Example deny passdb using passwd-file. You can use any passdb though. /etc/dovecot/conf.d/auth-deny.conf.ext:passdb { /etc/dovecot/conf.d/auth-checkpassword.conf.ext:passdb { /etc/dovecot/conf.d/auth-checkpassword.conf.ext:# passdb lookup should return also userdb info /etc/dovecot/conf.d/auth-dict.conf.ext:passdb { /etc/dovecot/conf.d/auth-master.conf.ext:# By adding master=yes setting inside a passdb you make the passdb a list /etc/dovecot/conf.d/auth-master.conf.ext:# Example master user passdb using passwd-file. You can use any passdb though. /etc/dovecot/conf.d/auth-master.conf.ext:passdb { /etc/dovecot/conf.d/auth-master.conf.ext: # be looked up from passdb that it really exists. pass=yes does that. /etc/dovecot/conf.d/auth-sql.conf.ext:passdb { /etc/dovecot/conf.d/auth-sql.conf.ext:# "prefetch" user database means that the passdb already provided the /etc/dovecot/conf.d/auth-static.conf.ext:# Static passdb. Included from 10-auth.conf. /etc/dovecot/conf.d/auth-static.conf.ext:#passdb { /etc/dovecot/conf.d/auth-static.conf.ext:#passdb { /etc/dovecot/conf.d/auth-ldap.conf.ext:passdb { /etc/dovecot/conf.d/auth-ldap.conf.ext:# "prefetch" user database means that the passdb already provided the /etc/dovecot/conf.d/auth-system.conf.ext:passdb { /etc/dovecot/conf.d/auth-system.conf.ext:#passdb { /etc/dovecot/conf.d/auth-system.conf.ext:#passdb { /etc/dovecot/conf.d/auth-system.conf.ext: # Normally static userdb handles this by doing a passdb lookup. This works /etc/dovecot/conf.d/auth-system.conf.ext: # with most passdbs, with PAM being the most notable exception. If you do /etc/dovecot/conf.d/auth-system.conf.ext: # the args in which case the passdb lookup is skipped. /etc/dovecot/dovecot-dict-auth.conf.ext:# This file is commonly accessed via passdb {} or userdb {} section in /etc/dovecot/dovecot-dict-auth.conf.ext:# The "passdb" and "userdb" keys are JSON objects containing key/value pairs, /etc/dovecot/dovecot-dict-auth.conf.ext:key passdb { /etc/dovecot/dovecot-dict-auth.conf.ext: key = passdb/%u /etc/dovecot/dovecot-dict-auth.conf.ext: # isn't specified at all (even as empty), the passdb/userdb lookup fails with /etc/dovecot/dovecot-dict-auth.conf.ext:# All the key/value pairs inside the object are added as passdb fields. /etc/dovecot/dovecot-dict-auth.conf.ext:passdb_objects = passdb /etc/dovecot/dovecot-dict-auth.conf.ext:#passdb_fields { /etc/dovecot/dovecot-ldap.conf.ext:# This file is commonly accessed via passdb {} or userdb {} section in /etc/dovecot/dovecot-ldap.conf.ext:# dovecot-ldap.conf.ext for userdb (it can even be a symlink, just as long as /etc/dovecot/dovecot-ldap.conf.ext:# the filename is different in userdb's args). That way one connection is used /etc/dovecot/dovecot-ldap.conf.ext:# If you wish to avoid two LDAP lookups (passdb + userdb), you can use /etc/dovecot/dovecot-ldap.conf.ext:# userdb prefetch instead of userdb ldap in dovecot.conf. In that case you'll /etc/dovecot/dovecot-ldap.conf.ext:# also have to include user_attrs in pass_attrs field prefixed with "userdb_" /etc/dovecot/dovecot-ldap.conf.ext:# homeDirectory=userdb_home,uidNumber=userdb_uid,gidNumber=userdb_gid /etc/dovecot/dovecot-sql.conf.ext:# This file is commonly accessed via passdb {} or userdb {} section in /etc/dovecot/dovecot-sql.conf.ext:# directories, you can use the static userdb module to generate the home

/etc/dovecot/dovecot-sql.conf.ext:# If you prefer to use the sql userdb module, you'll want to add fields
 /etc/dovecot/dovecot-sql.conf.ext:# userdb query to retrieve the user information. It can return fields:
 /etc/dovecot/dovecot-sql.conf.ext:# home or mail directory fits to a template string, you could use userdb
 static /etc/dovecot/dovecot-sql.conf.ext:# If you wish to avoid two SQL lookups (passdb + userdb), you
 can use /etc/dovecot/dovecot-sql.conf.ext:# userdb prefetch instead of userdb sql in dovecot.conf. In that
 case you'll /etc/dovecot/dovecot-sql.conf.ext:# also have to return userdb fields in password_query
 prefixed with "userdb_" /etc/dovecot/dovecot-sql.conf.ext:# home AS userdb_home, uid AS userdb_uid,
 gid AS userdb_gid \ /etc/dovecot/conf.d/10-director.conf: unix_listener director-userdb {
 /etc/dovecot/conf.d/10-director.conf: #auth_socket_path = director-userdb /etc/dovecot/conf.d/10-
 auth.conf:# blocking passdb and userdb queries (eg. MySQL and PAM). They're /etc/dovecot/conf.d/10-
 auth.conf:# You can have multiple passdbs and userdbs. This is useful if you want to
 /etc/dovecot/conf.d/10-auth.conf:# own them. For single-UID configuration use "static" userdb.
 /etc/dovecot/conf.d/auth-passwdfile.conf.ext:userdb { /etc/dovecot/conf.d/10-mail.conf:# System user and
 group used to access mails. If you use multiple, userdb /etc/dovecot/conf.d/10-
 mail.conf:#auth_socket_path = /var/run/dovecot/auth-userdb /etc/dovecot/conf.d/auth-
 checkpassword.conf.ext:# passdb lookup should return also userdb info /etc/dovecot/conf.d/auth-
 checkpassword.conf.ext:userdb { /etc/dovecot/conf.d/auth-checkpassword.conf.ext:# Standard
 checkpassword doesn't support direct userdb lookups. /etc/dovecot/conf.d/auth-
 checkpassword.conf.ext:# If you need checkpassword userdb, the checkpassword must support
 /etc/dovecot/conf.d/auth-checkpassword.conf.ext:#userdb { /etc/dovecot/conf.d/auth-dict.conf.ext:userdb {
 /etc/dovecot/conf.d/10-master.conf: # auth_socket_path points to this userdb socket by default. It's
 typically /etc/dovecot/conf.d/10-master.conf: # get the results of everyone's userdb lookups.
 /etc/dovecot/conf.d/10-master.conf: # userdb lookups will succeed only if the userdb returns an "uid" field
 that /etc/dovecot/conf.d/10-master.conf: unix_listener auth-userdb { /etc/dovecot/conf.d/90-quota.conf:#
 from userdb. It's also possible to give mailbox-specific limits, for example /etc/dovecot/conf.d/auth-
 sql.conf.ext:# needed information and there's no need to do a separate userdb lookup.
 /etc/dovecot/conf.d/auth-sql.conf.ext:#userdb { /etc/dovecot/conf.d/auth-sql.conf.ext:userdb {
 /etc/dovecot/conf.d/auth-sql.conf.ext:# by using userdb static instead of userdb sql, for example:
 /etc/dovecot/conf.d/auth-sql.conf.ext:#userdb { /etc/dovecot/conf.d/auth-static.conf.ext:#userdb {
 /etc/dovecot/conf.d/auth-ldap.conf.ext:# needed information and there's no need to do a separate userdb
 lookup. /etc/dovecot/conf.d/auth-ldap.conf.ext:#userdb { /etc/dovecot/conf.d/auth-ldap.conf.ext:userdb {
 /etc/dovecot/conf.d/auth-ldap.conf.ext:# If you don't have any user-specific settings, you can avoid the
 userdb LDAP /etc/dovecot/conf.d/auth-ldap.conf.ext:# lookup by using userdb static instead of userdb
 ldap, for example: /etc/dovecot/conf.d/auth-ldap.conf.ext:#userdb { /etc/dovecot/conf.d/auth-
 system.conf.ext:# PAM is typically used with either userdb passwd or userdb static.
 /etc/dovecot/conf.d/auth-system.conf.ext:userdb { /etc/dovecot/conf.d/auth-system.conf.ext:#userdb {
 /etc/dovecot/conf.d/auth-system.conf.ext: # Can return anything a userdb could normally return. For
 example: /etc/dovecot/conf.d/auth-system.conf.ext: # LDA and LMTP needs to look up users only from
 the userdb. This of course /etc/dovecot/conf.d/auth-system.conf.ext: # doesn't work with static userdb
 because there is no list of users. /etc/dovecot/conf.d/auth-system.conf.ext: # Normally static userdb
 handles this by doing a passdb lookup. This works /etc/dovecot/dovecot-dict-auth.conf.ext:# This file is
 commonly accessed via passdb {} or userdb {} section in /etc/dovecot/dovecot-dict-
 auth.conf.ext:iterate_prefix = userdb/ /etc/dovecot/dovecot-dict-auth.conf.ext:# Should iteration be
 disabled for this userdb? If this userdb acts only as a /etc/dovecot/dovecot-dict-auth.conf.ext:# The

"passdb" and "userdb" keys are JSON objects containing key/value pairs, /etc/dovecot/dovecot-dict-auth.conf.ext:key userdb { /etc/dovecot/dovecot-dict-auth.conf.ext: key = userdb/%u /etc/dovecot/dovecot-dict-auth.conf.ext: key = userdb/%u/quota /etc/dovecot/dovecot-dict-auth.conf.ext: # isn't specified at all (even as empty), the passdb/userdb lookup fails with /etc/dovecot/dovecot-dict-auth.conf.ext:userdb_objects = userdb /etc/dovecot/dovecot-dict-auth.conf.ext:userdb_fields { /etc/dovecot/dovecot-dict-auth.conf.ext: mail = maildir:%{dict:userdb.home}/Maildir root@LDAP-Ubu:/home#

ChatGPT Plus

¡Perfecto! Con esto ya lo tenemos claro. 😎

Lo que ves en el grep indica que **la definición que carga PAM/Passwd está en el archivo:**

swift

```
/etc/dovecot/conf.d/auth-system.conf.ext
```

y también se carga mediante:

swift

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

Pero normalmente, **para priorizar LDAP**, no necesitas tocar todos los archivos de Dovecot, solo debes **evitar que Dovecot use PAM y Passwd local**. Esto se hace **comentando o eliminando los passdb y userdb locales**, dejando solo los de LDAP.

Pasos exactos para tu sistema

1. Edita el archivo principal de autenticación:

bash

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

2. Busca estas líneas (aproximadamente por donde estaban tus comentarios):

text

```
!include auth-system.conf.ext
!include auth-ldap.conf.ext
```

3. **Comenta la línea que carga auth-system.conf.ext** (que es la que contiene PAM/Passwd):

text

```
# !include auth-system.conf.ext
!include auth-ldap.conf.ext
```

- Esto hará que **solo se use LDAP para passwd y userdb**.
- No necesitas tocar cada archivo individual (auth-system.conf.ext, etc.).

4. Guarda y cierra (Ctrl+O → Enter → Ctrl+X).

5. Reinicia Dovecot:

bash

```
systemctl restart dovecot
```

6. Prueba de nuevo:

bash

```
doveadm auth test villegas
```

- Ahora debería devolver algo como:

makefile

```
passwd: ldap (user villegas): successful
```

Nota importante:

Si `doveadm auth test` sigue diciendo `Can't connect to server: ldap.villegas.basilio`, **el problema ahora es la resolución del hostname o el firewall**, no Dovecot.

Para solucionarlo temporalmente, puedes poner en tu `/etc/dovecot/dovecot-ldap.conf.ext`:

text

```
hosts = 10.2.8.200
```

Así Dovecot se conecta directo a la IP del servidor LDAP.

Si quieres, puedo escribir **la configuración completa de Dovecot lista para tu LDAP**, comentando PAM/Passwd, usando LDAP por IP, y lista para reiniciar y probar villegas sin errores.

¿Quieres que haga eso?

Tú dijiste:

acabo de comentar las lineas esas, acabo de hacer esta prueba y me sale esto: root@LDAP-Ubu:/home# systemctl restart dovecot root@LDAP-Ubu:/home# dovecadm auth test villegas Password: passdb: villegas auth failed extra fields: user=villegas code=temp_fail root@LDAP-Ubu:/home#

ChatGPT Plus

¡Perfecto! Ahora estamos avanzando: 

El hecho de que dovecadm auth test villegas devuelva passdb: villegas auth failed con code=temp_fail significa **dos cosas clave**:

1. **Dovecot ya no está usando PAM/Passwd** → buen paso, porque antes ni llegaba a LDAP.
2. **Ahora el fallo es “temporal” y relacionado con LDAP**, no con Dovecot en sí.

Miremos tu log anterior:

vbnet

```
Error: ldap(/etc/dovecot/dovecot-ldap.conf.ext): Can't connect to server:
ldap.villegas.basilio
```

Esto indica que **Dovecot no puede conectarse al servidor LDAP**.

◆ Qué revisar y solucionar

1 Conexión al servidor LDAP

Prueba desde tu servidor Dovecot:

bash

```
ping ldap.villegas.basilio
```

Si no responde, Dovecot no podrá autenticarse. Esto explica temp_fail.

- Como solución rápida (y funcional), usa la IP directa en dovecot-ldap.conf.ext:

text

```
hosts = 10.2.8.200
```

2 Verifica el puerto LDAP (389)

bash

```
telnet 10.2.8.200 389
```

- Debes poder conectarte.
 - Si no → revisa firewall/iptables o configuración de OpenLDAP.
-

3 Configuración de dovecot-ldap.conf.ext

Asegúrate de que tenga algo así:

text

```
hosts = 10.2.8.200
dn = cn=admin,dc=villegas,dc=basilio
dnpass = usuario      # tu contraseña de admin LDAP
base = ou=usuarios,dc=villegas,dc=basilio
ldap_version = 3
scope = subtree
auth_bind = yes

user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))

user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail
default_pass_scheme = SSHA
```

- auth_bind = yes → hace que Dovecot haga bind directamente con la contraseña del usuario.
 - default_pass_scheme = SSHA → debe coincidir con tus contraseñas en LDAP.
-



4 Reinicia Dovecot y prueba de nuevo

bash

```
systemctl restart dovecot
doveadm auth test villegas
```

- Si devuelve passdb: ldap (user villegas): successful → autenticación LDAP funciona correctamente.
-

Resumen de por qué fallaba antes:

- PAM/Passwd bloqueaba LDAP → comentamos. 
 - Dovecot no podía resolver ldap.villegas.basilio → temp_fail. 
 - Solución temporal: usar IP (hosts = 10.2.8.200).
-

Si quieres, puedo darte **el archivo dovecot-ldap.conf.ext completo listo para copiar**, con todo configurado para tu servidor LDAP y usuarios villegas, svb111, etc., de forma que dovecadm auth test funcione al primer intento.

¿Quieres que haga eso?

Tú dijiste:

dovecot-ldap.conf.ext no existe telenet desde el cliente si funciona

ChatGPT Plus

¡Ah, perfecto! Esto aclara todo. 😊

Si dovecot-ldap.conf.ext **no existe**, significa que Dovecot no tiene la configuración LDAP lista para usar. Por eso aunque la conectividad LDAP funciona (telnet desde cliente al puerto 389), **Dovecot no sabe a qué servidor conectarse ni cómo buscar los usuarios** → passdb: temp_fail.

♦ Cómo solucionarlo: crear dovecot-ldap.conf.ext

1. Crea el archivo:

bash

```
nano /etc/dovecot/dovecot-ldap.conf.ext
```

2. Copia y pega esta configuración adaptada a tu LDAP:

text

```
hosts = 10.2.8.200
dn = cn=admin,dc=villegas,dc=basilio
dnpass = usuario
base = ou=usuarios,dc=villegas,dc=basilio
ldap_version = 3
scope = subtree
auth_bind = yes

user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))

user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail
default_pass_scheme = SSHA
```

Explicación rápida:

- `hosts` → IP del servidor LDAP
- `dn` y `dnpass` → usuario administrador de LDAP
- `base` → contenedor donde están los usuarios
- `auth_bind=yes` → Dovecot hace bind directo con la contraseña del usuario
- `default_pass_scheme=SSHA` → coincide con tus contraseñas LDAP

3. Guarda y cierra (`Ctrl+O` → `Enter` → `Ctrl+X`).

♦ Verifica que Dovecot use este archivo

En tu `/etc/dovecot/conf.d/10-auth.conf` debe aparecer:

text

```
passdb {  
    args = /etc/dovecot/dovecot-ldap.conf.ext  
    driver = ldap  
}  
  
userdb {  
    args = /etc/dovecot/dovecot-ldap.conf.ext  
    driver = ldap  
}
```

- Asegúrate de **comentar cualquier** `passdb driver=pam` o `userdb driver=passwd`, como hicimos antes.

♦ Reinicia Dovecot y prueba

bash

```
systemctl restart dovecot  
doveadm auth test villegas
```

- Si todo está correcto, ahora deberías ver:

makefile

```
passdb: ldap (user villegas): successful
```

Si quieres, puedo escribir **el conjunto completo de archivos de configuración de Dovecot** que funcione directamente con tu LDAP (`villegas.basilio`), incluyendo `passdb/userdb LDAP` y

mailboxes, listo para copiar y pegar. Esto solucionaría **todos los problemas de autenticación** de golpe.

¿Quieres que haga eso?

Tú dijiste:

Password: passdb: villegas auth failed extra fields: user=villegas code=temp_fail root@LDAP-Ubu:/home#

ChatGPT Plus

Perfecto, gracias por el detalle. 😊

Ahora esto es **muy claro**: `passdb: villegas auth failed` con `code=temp_fail` significa que **Dovecot no puede comunicarse correctamente con el servidor LDAP**. No es problema de usuario/contraseña, sino de **conexión/configuración LDAP dentro de Dovecot**.

Aunque el `telnet` al puerto 389 funciona, Dovecot necesita **una configuración LDAP explícita** y un archivo `dovecot-ldap.conf.ext` para saber:

- a qué servidor LDAP conectarse,
- con qué usuario DN para hacer binds,
- cuál es el contenedor base de usuarios,
- cómo filtrar usuarios y contraseñas,
- qué esquema de hash de contraseña usar (SSHA en tu caso).

♦ Próximo paso: crear `dovecot-ldap.conf.ext` completo

Crea este archivo **como root**:

`bash`

```
nano /etc/dovecot/dovecot-ldap.conf.ext
```

Pega lo siguiente:

`text`

```
hosts = 10.2.8.200
dn = cn=admin,dc=villegas,dc=basilio
dnpass = usuario
base = ou=usuarios,dc=villegas,dc=basilio
ldap_version = 3
scope = subtree
```

```
auth_bind = yes

user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))

user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail
default_pass_scheme = SSHA
```

- Cambia dnpass por la contraseña del administrador LDAP (usuario en tu caso).
- hosts usa la IP de tu servidor LDAP (10.2.8.200) porque Dovecot a veces no resuelve correctamente ldap.villegas.basilio.

Guarda y cierra (Ctrl+O, Enter, Ctrl+X).

♦ Asegúrate que Dovecot use este archivo

En /etc/dovecot/conf.d/10-auth.conf, verifica que solo esté esto:

text

```
passdb {
    args = /etc/dovecot/dovecot-ldap.conf.ext
    driver = ldap
}

userdb {
    args = /etc/dovecot/dovecot-ldap.conf.ext
    driver = ldap
}
```

- Comenta cualquier línea que cargue PAM o passwd local:

text

```
# !include auth-system.conf.ext
```

♦ Reinicia Dovecot y prueba

bash

```
systemctl restart dovecot
doveadm auth test villegas
```

- Resultado esperado:

makefile

```
passdb: ldap (user villegas): successful
```

Esto **resuelve el temp_fail**.

Si quieres, puedo escribir **todo el conjunto de archivos de configuración de Dovecot listos para LDAP**, incluyendo mailboxes y SSL/TLS, de modo que tu servidor de correo quede funcional con los usuarios LDAP villegas, svb111, etc.

¿Quieres que haga eso?

Tú dijiste:

he comentado la linea: `!include auth-system.conf.ext #!include auth-sql.conf.ext #!include auth-ldap.conf.ext #!include auth-passwdfile.conf.ext #!include auth-checkpassword.conf.ext #!include auth-static.conf.ext` y ahora me sale esto: `root@LDAP-Ubu:/home# systemctl restart dovecot root@LDAP-Ubu:/home# doveadm auth test villegas Password: Fatal: Couldn't connect to auth socket root@LDAP-Ubu:/home#`

ChatGPT Plus

¡Ah! Ahora estamos en terreno conocido. 😊

El error:

vbnet

```
Fatal: Couldn't connect to auth socket
```

significa que **Dovecot ya no tiene ningún passdb/userdb activo**, porque comentaste todas las líneas que incluyen los módulos de autenticación, incluyendo LDAP.

⚠ Comentaste:

text

```
!include auth-ldap.conf.ext
```

y eso **era justamente lo que configuraba la conexión LDAP**. Al comentar todo, **Dovecot ya no tiene forma de autenticar usuarios**, por eso falla al intentar usar `doveadm auth`.

◆ Cómo solucionarlo

1. **Descomenta al menos el módulo LDAP** para autenticación:

text

```
!include auth-ldap.conf.ext
```

- Deja comentadas las que no uses (auth-system.conf.ext, auth-sql.conf.ext, etc.).
 - Así solo Dovecot usará LDAP para passdb y userdb.
2. Asegúrate que /etc/dovecot/conf.d/auth-ldap.conf.ext **exista** y tenga la configuración correcta (o crea /etc/dovecot/dovecot-ldap.conf.ext y apunta a él desde auth-ldap.conf.ext).

Por ejemplo, en /etc/dovecot/conf.d/auth-ldap.conf.ext:

text

```
passdb {  
    driver = ldap  
    args = /etc/dovecot/dovecot-ldap.conf.ext  
}  
  
userdb {  
    driver = ldap  
    args = /etc/dovecot/dovecot-ldap.conf.ext  
}
```

Y /etc/dovecot/dovecot-ldap.conf.ext:

text

```
hosts = 10.2.8.200  
dn = cn=admin,dc=villegas,dc=basilio  
dnpass = usuario  
base = ou=usuarios,dc=villegas,dc=basilio  
ldap_version = 3  
scope = subtree  
auth_bind = yes  
  
user_filter = (&(objectClass=posixAccount)(uid=%u))  
pass_filter = (&(objectClass=posixAccount)(uid=%u))  
  
user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail  
default_pass_scheme = SSHA
```

3. Reinicia Dovecot:


bash

```
systemctl restart dovecot
```

4. Prueba:

```
bash
```

```
doveadm auth test villegas
```

- Ahora debería devolver `passdb: ldap (user villegas): successful.` 

Resumen:

- No puedes comentar todos los `!include`, porque entonces Dovecot **no tiene ningún mecanismo de autenticación** → falla el socket auth.
- Debes mantener **al menos LDAP** (y opcionalmente static o passwd si quieres).

Si quieres, puedo darte **la configuración mínima de Dovecot lista para tu LDAP**, solo con `auth-ldap` activo y funcionando desde cero, lista para reiniciar y autenticar `villegas` y todos los usuarios.

¿Quieres que haga eso?

Tú dijiste:

```
volvemos a esto habiendo comentado todas menos la de ldap: root@LDAP-Ubu:/home# systemctl
restart dovecot root@LDAP-Ubu:/home# doveadm auth test villegas Password: passdb: villegas auth
failed extra fields: user=villegas code=temp_fail root@LDAP-Ubu:/home# GNU nano 7.2
/etc/dovecot/conf.d/auth-ldap.conf.ext # Authentication for LDAP users. Included from 10-auth.conf. # #
<doc/wiki/AuthDatabase.LDAP.txt> passdb { driver = ldap # Path for LDAP configuration file, see
example-config/dovecot-ldap.conf.ext args = /etc/dovecot/dovecot-ldap.conf.ext } # "prefetch" user
database means that the passdb already provided the # needed information and there's no need to do a
separate userdb lookup. # <doc/wiki/UserDatabase.Prefetch.txt> #userdb { # driver = prefetch #} userdb {
driver = ldap args = /etc/dovecot/dovecot-ldap.conf.ext # Default fields can be used to specify defaults
that LDAP may override #default_fields = home=/home/virtual/%u } GNU nano 7.2 /etc/dovecot/dovecot-
ldap.conf.ext # There are also other special fields which can be returned, see #
http://wiki2.dovecot.org/PasswordDatabase/ExtraFields #pass_attrs = uid=user,userPassword=password
# If you wish to avoid two LDAP lookups (passdb + userdb), you can use # userdb prefetch instead of
userdb ldap in dovecot.conf. In that case you'll # also have to include user_attrs in pass_attrs field
prefixed with "userdb_" # string. For example: #pass_attrs = uid=user,userPassword=password,\ #
homeDirectory=userdb_home,uidNumber=userdb_uid,gidNumber=userdb_gid # Filter for password
lookups #pass_filter = (&(objectClass=posixAccount)(uid=%u)) # Attributes and filter to get a list of all
users #iterate_attrs = uid=user #iterate_filter = (objectClass=posixAccount) # Default password scheme. "
{scheme}" before password overrides this. # List of supported schemes is in:
http://wiki2.dovecot.org/Authentication #default_pass_scheme = CRYPT # By default all LDAP lookups
are performed by the auth master process. # If blocking=yes, auth worker processes are used to perform
```

the lookups. # Each auth worker process creates its own LDAP connection so this can # increase parallelism. With blocking=no the auth master process can # keep 8 requests pipelined for the LDAP connection, while with blocking=yes # each connection has a maximum of 1 request running. For small systems the # blocking=no is sufficient and uses less resources. #blocking = no hosts = ldap.villegas.basilio dn = cn=admin,dc=villegas,dc=basilio dnpass = usuario base = dc=villegas,dc=basilio ldap_version = 3 scope = subtree auth_bind = yes # Filtrar usuarios y contrase as user_filter = (&(objectClass=posixAccount)(uid=%u)) pass_filter = (&(objectClass=posixAccount)(uid=%u)) # Mapear atributos LDAP a Dovecot user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail default_pass_scheme = SSHA

ChatGPT Plus

¡Perfecto! Gracias por compartir los archivos. Ahora está todo mucho más claro. 😎

Actualmente tu `dovecot-ldap.conf.ext` tiene **un par de problemas clave** que están causando `temp_fail`:

1 hosts = ldap.villegas.basilio

- Dovecot no puede resolver correctamente `ldap.villegas.basilio`, aunque ping funcione a veces.
- La forma más segura es usar la **IP del servidor LDAP**:

text

```
hosts = 10.2.8.200
```

Esto elimina cualquier problema de DNS/hostname.

2 base = dc=villegas,dc=basilio

- Actualmente estás poniendo la **base en el root DN** del dominio, pero tus usuarios están en `ou=usuarios`.
- Cambia base a:

text

```
base = ou=usuarios,dc=villegas,dc=basilio
```

- Esto hace que Dovecot solo busque usuarios donde realmente están, evitando búsquedas que fallan.

3 user_attrs

Actualmente tienes:

text

```
user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail
```

- Los nombres a la izquierda deben ser los **atributos de Dovecot** (home, uid, gid, mail) y a la derecha los **atributos LDAP** (homeDirectory, uidNumber, gidNumber, mail).
- Está correcto, solo asegúrate de no mezclar nombres.

4 default_pass_scheme

- Está correcto (SSHA), que coincide con las contraseñas que pusiste.

✓ Archivo dovecot-ldap.conf.ext corregido

text

```
hosts = 10.2.8.200
dn = cn=admin,dc=villegas,dc=basilio
dnpass = usuario
base = ou=usuarios,dc=villegas,dc=basilio
ldap_version = 3
scope = subtree
auth_bind = yes

user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))

user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail
default_pass_scheme = SSHA
```

5 Pasos finales

1. Guarda el archivo /etc/dovecot/dovecot-ldap.conf.ext.
2. Reinicia Dovecot:

bash

```
systemctl restart dovecot
```

3. Prueba autenticación:

bash

```
doveadm auth test villegas
```

- Ahora debería devolver:

makefile

```
passdb: ldap (user villegas): successful
```

Resumen del problema anterior:

- hosts apuntaba a un nombre que Dovecot no resolvía correctamente.
- base estaba en dc=villegas,dc=basilio en lugar de ou=usuarios,..., así que Dovecot no encontraba los usuarios.

Con estas correcciones, la autenticación LDAP debería funcionar correctamente para todos los usuarios.

Si quieres, puedo escribir **la configuración final de auth-ldap.conf.ext + dovecot-ldap.conf.ext lista para copiar**, de manera que puedas reiniciar Dovecot y autenticar **villegas, svb111 y todos los demás usuarios** sin errores.

¿Quieres que haga eso?

Tú dijiste:

```
root@LDAP-Ubu:/home# dovecadm auth test villegas Password: passdb: villegas auth succeeded extra fields: user=villegas
```

ChatGPT Plus

¡Excelente! 🎉

Eso significa que **Dovecot ya autentica correctamente al usuario villegas vía LDAP.** ✅

Ahora todo está funcionando como debería:

- passdb: ldap → Dovecot está usando LDAP.
 - auth succeeded → la contraseña funciona perfectamente.
 - Puedes repetir la prueba con cualquier otro usuario LDAP (svb111, basilio, etc.) y debería funcionar igual.
-

Próximos pasos recomendados

1. **Prueba con un cliente de correo** (Thunderbird/Outlook) usando:

- Usuario: villegas@villegas.basilio
- Contraseña: la que pusiste en LDAP (usuario)
- Servidor IMAP: 10.2.8.200 (o mail.villegas.basilio si lo resolviste en /etc/hosts)
- Servidor SMTP: el mismo host

2. **Verifica envío/recepción** de correo entre usuarios LDAP.

3. **Revisa logs de Dovecot** mientras pruebas:

bash

```
journalctl -u dovecot -f
```

- Deberías ver líneas de imap-login: Login: user=<villegas@villegas.basilio>... authenticated

4. **Mantener seguridad:**

- Considera habilitar disable_plaintext_auth = yes y SSL/TLS para IMAP/SMTP.
- Esto protegerá las contraseñas LDAP al conectarse desde clientes de correo.

Si quieres, puedo prepararte **una guía completa para que Dovecot + Postfix + LDAP quede listo para todos los usuarios de tu empresa**, incluyendo IMAP, SMTP, SSL y OwnCloud, todo centralizado en LDAP.

¿Quieres que haga eso?

Tú dijiste:

en el cliente me sigue saliendo que no se puede iniciar sesion por configuracion, usuario o contraseña erroneo. estas seguro de que el usuario (correo) es villegas@villegas.basilio?

ChatGPT Plus

¡Ah! Ahora lo tenemos que aclarar bien, porque aquí suele estar el malentendido. 😊

En Dovecot con LDAP, **el nombre de usuario que Dovecot espera depende de tu configuración de user_filter y pass_filter.**

Actualmente tienes en /etc/dovecot/dovecot-ldap.conf.ext:

text

```
user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))
```

- %u → es **exactamente lo que el usuario escribe en el cliente**.
- En tu LDAP, el usuario se llama:

makefile

```
uid: villegas
mail: villegas@villegas.basilio
```

Por tanto, **Dovecot espera que el usuario escriba villegas, no villegas@villegas.basilio**, porque tu filtro está usando uid=%u y no mail=%u.

♦ Opciones

1. **Usar solo el UID** (recomendado con tu configuración actual):

- Usuario en el cliente: villegas
- Contraseña: usuario

2. **Usar el correo completo como login** (villegas@villegas.basilio):

- Cambia el filtro LDAP a usar el atributo mail:

text

```
user_filter = (&(objectClass=posixAccount)(mail=%u))
pass_filter = (&(objectClass=posixAccount)(mail=%u))
```

- Con esto, Dovecot buscará usuarios por el correo en lugar de por el UID.

💡 Resumen:

- Con tu configuración actual (uid=%u) → el login debe ser **villegas**
- Si quieres usar el correo completo como login → cambia los filtros a mail=%u

Si quieres, puedo **escribirte el dovecot-ldap.conf.ext listo para que funcione con el correo completo como usuario**, así tus clientes pueden poner villegas@villegas.basilio.

¿Quieres que haga eso?

Tú dijiste:

Si quieres, puedo escribirte el `dovecot-ldap.conf.ext` listo para que funcione con el correo completo como usuario, así tus clientes pueden poner `villegas@villegas.basilio`. si, hazlo porfa

ChatGPT Plus

¡Perfecto! 🤓 Vamos a dejarlo listo para que tus usuarios puedan iniciar sesión usando **su correo completo** (`villegas@villegas.basilio`) en cualquier cliente de correo.

Archivo `/etc/dovecot/dovecot-ldap.conf.ext`

text

```
# Servidor LDAP
hosts = 10.2.8.200
dn = cn=admin,dc=villegas,dc=basilio
dnpass = usuario
base = ou=usuarios,dc=villegas,dc=basilio
ldap_version = 3
scope = subtree
auth_bind = yes

# Buscar usuario y contraseña usando el correo completo
user_filter = (&(objectClass=posixAccount)(mail=%u))
pass_filter = (&(objectClass=posixAccount)(mail=%u))

# Mapear atributos LDAP a Dovecot
user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid,mail=mail

# Contraseña encriptada en SSHA
default_pass_scheme = SSHA
```

♦ Pasos a seguir

1. Guarda este archivo como `/etc/dovecot/dovecot-ldap.conf.ext`.
2. Asegúrate de que en `/etc/dovecot/conf.d/auth-ldap.conf.ext` tengas:

text

```
passdb {
    driver = ldap
    args = /etc/dovecot/dovecot-ldap.conf.ext
}

userdb {
```

```
driver = ldap
args = /etc/dovecot/dovecot-ldap.conf.ext
}
```

3. Reinicia Dovecot:

bash

```
systemctl restart dovecot
```

4. Prueba la autenticación en terminal:

bash

```
doveadm auth test villegas@villegas.basilio
```

- Debe devolver:

makefile

```
passdb: ldap (user villegas): successful
```

5. Ahora, en Thunderbird/Outlook, usa:

- Usuario: villegas@villegas.basilio
- Contraseña: la que asignaste (usuario)
- Servidor IMAP: 10.2.8.200 (o mail.villegas.basilio si lo resolviste en /etc/hosts)
- Servidor SMTP: igual

Con esto, **todos tus clientes podrán autenticarse usando el correo completo**, y no tendrás que pedir a los usuarios que recuerden solo el UID.

Si quieres, puedo darte **la configuración completa final de Dovecot + Postfix lista para producción**, lista para que todos los usuarios LDAP envíen y reciban correo correctamente.

¿Quieres que haga eso?

Tú dijiste:

```
root@LDAP-Ubu:/home# dovecot auth test villegas@villegas.basilio Password: passdb:
villegas@villegas.basilio auth succeeded extra fields: user=villegas
original_user=villegas@villegas.basilio root@LDAP-Ubu:/home# pero me sigue diciendo en el cliente
que no es correcto el usuario o contraseña. ademas, ahora tarda mucho menos en cargar para darme
ese resultado
```

¡Perfecto! Esto es un **gran avance** 😎

Ahora Dovecot **sí reconoce el correo completo** (villegas@villegas.basilio) y autentica correctamente contra LDAP:

graphql

```
passdb: villegas@villegas.basilio auth succeeded
```

- user=villegas → el UID real en LDAP
- original_user=villegas@villegas.basilio → lo que puso el cliente

Esto significa que **el problema ya no está en Dovecot**, sino probablemente en **la configuración del cliente de correo o el servidor SMTP/IMAP**.

♦ Cosas a revisar en el cliente

1. **Usuario en el cliente:** villegas@villegas.basilio ✓
2. **Contraseña:** la que asignaste en LDAP (usuario) ✓
3. **Servidor IMAP:** debe coincidir exactamente con el que Dovecot reconoce.
 - Si antes funcionaba con la IP (10.2.8.200) y no con mail.villegas.basilio, usa **la IP** para asegurarte de que se conecta al servidor correcto.
 - Si quieres usar mail.villegas.basilio, asegúrate de que **DNS o /etc/hosts del cliente** resuelva correctamente.
4. **Puerto:**
 - IMAP: 143 (sin SSL) o 993 (SSL/TLS)
 - SMTP: 25, 587 (STARTTLS) o 465 (SMTPS)
5. **Seguridad/SSL:**
 - Si el cliente fuerza SSL y tu Dovecot no tiene certificados configurados, puede fallar la autenticación aunque Dovecot funcione.
 - Para pruebas rápidas, usa **sin SSL** para asegurarte de que el login funciona.

♦ Verifica logs de Dovecot en tiempo real mientras intentas login

bash

```
journalctl -u dovecot -f
```

- Si el cliente se conecta, verás algo como:

pgsql

```
imap-login: Login: user=<villegas@villegas.basilio> method=PLAIN  
rip=IP_DEL_CLIENT lip=10.2.8.200
```

- Si no ves nada → el cliente **no llega al servidor Dovecot** (problema de host/puerto/firewall).



Conclusión:

- La autenticación LDAP ya funciona correctamente.
- El error que ve el cliente ahora es **configuración de servidor en el cliente o conectividad**, no Dovecot/LDAP.

Si quieres, puedo prepararte **una guía exacta de configuración del cliente (Thunderbird/Outlook)** usando tu LDAP + Dovecot, con los valores de servidor, puerto y seguridad correctos, para que inicie sesión sin problemas.

¿Quieres que haga eso?

Tú dijiste:

```
root@LDAP-Ubu:/home# journalctl -u dovecot -f Oct 21 11:39:43 LDAP-Ubu dovecot[5062]: imap-login:  
Disconnected: Connection closed: SSL_accept() failed: error:0A000412:SSL routines::ssl3 alert bad  
certificate: SSL alert number 42 (no auth attempts in 0 secs): user=<>, rip=10.2.8.108, lip=10.2.8.200,  
TLS handshaking: SSL_accept() failed: error:0A000412:SSL routines::ssl3 alert bad certificate: SSL  
alert number 42, session=<AITwp6lBytEKAgHs> Oct 21 11:39:43 LDAP-Ubu dovecot[5062]: imap-login:  
Disconnected: Connection closed: SSL_accept() failed: error:0A000412:SSL routines::ssl3 alert bad  
certificate: SSL alert number 42 (no auth attempts in 0 secs): user=<>, rip=10.2.8.108, lip=10.2.8.200,  
TLS handshaking: SSL_accept() failed: error:0A000412:SSL routines::ssl3 alert bad certificate: SSL  
alert number 42, session=<Qovwp6lBztEKAgHs> Oct 21 11:39:59 LDAP-Ubu dovecot[5062]: imap-login:  
Disconnected: Aborted login by logging out (no auth attempts in 0 secs): user=<>, rip=10.2.8.108,  
lip=10.2.8.200, session=<h33pqKlBfolKAgHs> Oct 21 11:39:59 LDAP-Ubu dovecot[5062]: imap-login:  
Disconnected: Connection closed (no auth attempts in 0 secs): user=<>, rip=10.2.8.108, lip=10.2.8.200,  
session=<amrxqKlBjllKAgHs> Oct 21 11:40:13 LDAP-Ubu dovecot[5062]: auth: Error:  
ldap(villegas,10.2.8.108,<Bdm4qalBDN4KAgHs>): user not found from userdb Oct 21 11:40:13 LDAP-  
Ubu dovecot[5062]: imap(5149): Error: auth-master: login: request [708837377]: Login auth request  
failed: Authenticated user not found from userdb, auth lookup id=708837377 (auth connected 1 msecs  
ago, request took 1 msecs, client-pid=5148 client-id=1) Oct 21 11:40:13 LDAP-Ubu dovecot[5062]: imap-  
login: Disconnected: Internal login failure (pid=5148 id=1): user=<villegas>, method=PLAIN,  
rip=10.2.8.108, lip=10.2.8.200, mpid=5149, session=<Bdm4qalBDN4KAgHs> Oct 21 11:40:13 LDAP-Ubu  
dovecot[5062]: auth: Error: ldap(villegas,10.2.8.108,<jsy5qalBDt4KAgHs>): user not found from userdb
```


Oct 21 11:40:13 LDAP-Ubu dovecot[5062]: imap(5151): Error: auth-master: login: request [3432906753]: Login auth request failed: Authenticated user not found from userdb, auth lookup id=3432906753 (auth connected 1 msecs ago, request took 0 msecs, client-pid=5150 client-id=1) Oct 21 11:40:13 LDAP-Ubu dovecot[5062]: imap-login: Disconnected: Internal login failure (pid=5150 id=1): user=<villegas>, method=PLAIN, rip=10.2.8.108, lip=10.2.8.200, mpid=5151, session=<jsy5qalBDt4KAghs>

ChatGPT Plus

¡Ah, ahora todo tiene sentido! 😊

Los logs muestran dos problemas diferentes:

1 SSL / TLS en IMAP

scss

```
SSL_accept() failed: error:0A000412:SSL routines::sslv3 alert bad certificate
```

- El cliente está intentando conectarse **usando SSL/TLS**, pero Dovecot **no tiene un certificado válido** para ese host, o el cliente no confía en el certificado.
- Por eso **el login ni siquiera llega a autenticarse**, se corta durante el handshake SSL.

Soluciones rápidas para pruebas:

1. Conectar **sin SSL** en el cliente, usando IMAP puerto 143 (STARTTLS desactivado temporalmente).
2. O generar un certificado autofirmado y configurarlo en Dovecot:

text

```
ssl = yes
ssl_cert = </etc/dovecot/private/dovecot.pem
ssl_key = </etc/dovecot/private/dovecot.key
```


- Asegúrate de que el cliente confíe en ese certificado.

2 Usuario no encontrado en userdb

pgsql

```
user not found from userdb
```

- Esto ocurre porque Dovecot recibe un usuario diferente al que espera en LDAP.

- En tu último test `doveadm auth test villegas@villegas.basilio` funcionó 
- Pero en el cliente, parece que está enviando **solo villegas** en vez de `villegas@villegas.basilio`, o el mapping de `original_user` → `user` falla.

Con tu configuración actual de `dovecot-ldap.conf.ext` para usar correo completo:


text

```
user_filter = (&(objectClass=posixAccount)(mail=%u))
pass_filter = (&(objectClass=posixAccount)(mail=%u))
```

- Debes asegurarte de que el **cliente realmente envíe villegas@villegas.basilio como nombre de usuario**.

♦ Qué hacer ahora

1. **Prueba sin SSL/TLS** en Thunderbird/Outlook (puerto 143) para ver si el login funciona.
2. Si funciona, entonces el problema era solo el certificado SSL.
3. Verifica que el usuario escrito en el cliente sea exactamente el correo (`villegas@villegas.basilio`).
4. Una vez funcione sin SSL, puedes generar un certificado válido para tu dominio y habilitar SSL/TLS correctamente.

 Nota: Es **muy normal** que `doveadm auth test` funcione pero el cliente falle si hay **problemas de TLS o usuario mal escrito**.

Si quieres, puedo escribirte **los pasos exactos para habilitar SSL/TLS autofirmado en Dovecot**, de modo que los clientes puedan conectarse con correo completo y contraseña LDAP sin errores de certificado.

¿Quieres que haga eso?

Tú dijiste:

lo estoy haciendo sin ssl y en el puerto 143, solo he hecho un intento en ssl para ver si funcionaba

ChatGPT Plus

Perfecto, gracias por aclararlo. Entonces podemos descartar el problema SSL/TLS, y lo que tenemos ahora es **que Dovecot no encuentra al usuario en userdb cuando el cliente intenta iniciar sesión**.

♦ Lo que muestran los logs

vbnet

```
auth: Error: ldap(villegas,10.2.8.108,<Bdm4qalBDN4KAghs>): user not found from userdb
```

- Fíjate que el cliente está enviando villegas como nombre de usuario, **no villegas@villegas.basilio**.
- En tu configuración de Dovecot con LDAP usando mail=%u, **espera exactamente el correo completo** (villegas@villegas.basilio).
- Por eso falla: Dovecot recibe villegas, busca mail=villegas en LDAP, **no encuentra nada**, y devuelve error user not found from userdb.

♦ Cómo solucionarlo

Tienes dos opciones:

Opción 1: Configurar el cliente para que use el correo completo

- Usuario: villegas@villegas.basilio
- Contraseña: usuario
- Servidor: 10.2.8.200
- Puerto: 143

✓ Esta es la forma correcta según tu configuración actual de Dovecot/LDAP (mail=%u).

Opción 2: Cambiar Dovecot para aceptar UID como login

Si quieres que el cliente use villegas como login:

1. Modifica dovecot-ldap.conf.ext:

text

```
user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))
```

2. Reinicia Dovecot:

bash

```
systemctl restart dovecot
```

3. Ahora el login en el cliente debe ser:

- Usuario: villegas
- Contraseña: usuario

💡 **Clave:** El valor que el cliente envía **debe coincidir exactamente con %u** en los filtros LDAP.

Si quieres, puedo hacer **una tabla con las dos configuraciones posibles**, para que sepas exactamente qué usar en Dovecot y qué poner en Thunderbird/Outlook para todos los usuarios.

¿Quieres que haga eso?

Tú dijiste:

pero es que en thunderbird no me deja poner solo villegas de usuario, tengo que poner un correo valido

ChatGPT Plus

¡Ah! Ahora está claro por qué falla: Thunderbird **obliga a poner un correo válido** en la casilla de usuario. 😊

Entonces tu configuración actual de Dovecot/LDAP **usando el correo completo (mail=%u) es la correcta**, pero el cliente debe enviar exactamente el correo que está en LDAP.

♦ Qué revisar

1. En tu LDAP, el correo es:

graphql

```
mail: villegas@villegas.basilio
```

2. En Dovecot, tu `dovecot-ldap.conf.ext` está así:

text

```
user_filter = (&(objectClass=posixAccount)(mail=%u))
pass_filter = (&(objectClass=posixAccount)(mail=%u))
```

- Esto significa que Dovecot **espera villegas@villegas.basilio como login.** ✅
- 3. Asegúrate de que el **campo “Nombre de usuario” en Thunderbird** sea exactamente:

CSS

```
villegas@villegas.basilio
```

4. Servidor IMAP/SMTP:

- Host: 10.2.8.200 o mail.villegas.basilio (si DNS/hosts resuelve)
- Puerto IMAP: 143 (sin SSL/TLS para pruebas)
- Puerto SMTP: 25 o 587 (STARTTLS desactivado para pruebas)

♦ Qué verificar en Dovecot mientras Thunderbird intenta login

bash

```
journalctl -u dovecot -f
```

- Si el login llega, verás algo como:

pgsql

```
imap-login: Login: user=<villegas@villegas.basilio> method=PLAIN  
rip=IP_DEL_CLIENT lip=10.2.8.200
```

- Si no ves nada, el cliente **ni siquiera llega al servidor** (firewall, puerto incorrecto, o hostname mal resuelto).

💡 Conclusión:

- Tu LDAP y Dovecot ya están listos para usar correos como login.
- Si Thunderbird sigue diciendo “usuario o contraseña incorrecta”, el problema **ya no es Dovecot**, sino que el cliente:

1. No envía exactamente villegas@villegas.basilio como usuario, o
2. No llega al servidor correcto (puerto/host/firewall).

Si quieres, puedo escribir **la configuración exacta para Thunderbird/Outlook paso a paso**, usando villegas@villegas.basilio y LDAP + Dovecot, que funcione sin errores.

¿Quieres que haga eso?

Tú dijiste:

```
root@LDAP-Ubu:/home# journalctl -u dovecot -f Oct 21 11:44:28 LDAP-Ubu dovecot[5062]: imap-login:  
Disconnected: Connection closed: SSL_accept() failed: error:0A000412:SSL routines::ssl3 alert bad  
certificate: SSL alert number 42 (no auth attempts in 0 secs): user=<>, rip=10.2.8.108, lip=10.2.8.200,  
TLS handshaking: SSL_accept() failed: error:0A000412:SSL routines::ssl3 alert bad certificate: SSL  
alert number 42, session=<WHD5uKIBCucKAghs> Oct 21 11:44:29 LDAP-Ubu dovecot[5062]: imap-  
login: Disconnected: Aborted login by logging out (no auth attempts in 0 secs): user=<>, rip=10.2.8.108,  
lip=10.2.8.200, session=</sb9uKIBYJYKAghs> Oct 21 11:44:29 LDAP-Ubu dovecot[5062]: pop3-login:
```

Disconnected: Connection closed: SSL_accept() failed: error:0A000412:SSL routines::ssl3 alert bad certificate: SSL alert number 42 (no auth attempts in 0 secs): user=<>, rip=10.2.8.108, lip=10.2.8.200, TLS handshaking: SSL_accept() failed: error:0A000412:SSL routines::ssl3 alert bad certificate: SSL alert number 42, session=<dUwEualButMKAGhs> Oct 21 11:44:30 LDAP-Ubu dovecot[5062]: pop3-login: Disconnected: Aborted login by logging out (no auth attempts in 0 secs): user=<>, rip=10.2.8.108, lip=10.2.8.200, session=<WI0KualBLKAKAGhs> Oct 21 11:44:53 LDAP-Ubu dovecot[5062]: imap-login: Disconnected: Aborted login by logging out (no auth attempts in 0 secs): user=<>, rip=10.2.8.108, lip=10.2.8.200, session=<sYI2uqlBrlwKAGhs> Oct 21 11:44:54 LDAP-Ubu dovecot[5062]: imap-login: Disconnected: Connection closed (no auth attempts in 1 secs): user=<>, rip=10.2.8.108, lip=10.2.8.200, session=<fF1+uqlBrowKAGhs> Oct 21 11:45:12 LDAP-Ubu dovecot[5062]: imap-login: Disconnected: Connection closed (auth failed, 3 attempts in 14 secs): user=<svb@mail.villegas.basilio>, method=PLAIN, rip=10.2.8.108, lip=10.2.8.200, session=<bje8uqlBRoUKAGhs> Oct 21 11:45:21 LDAP-Ubu dovecot[5062]: imap-login: Disconnected: Aborted login by logging out (no auth attempts in 0 secs): user=<>, rip=10.2.8.108, lip=10.2.8.200, session=<aHUcvKIB0IAKAGhs> Oct 21 11:45:22 LDAP-Ubu dovecot[5062]: imap-login: Disconnected: Connection closed (no auth attempts in 1 secs): user=<>, rip=10.2.8.108, lip=10.2.8.200, session=<1JgkvKIB3oAKAGhs> Oct 21 11:45:29 LDAP-Ubu dovecot[5062]: imap-login: Disconnected: Connection closed (auth failed, 3 attempts in 6 secs): user=<svb@mail.villegas.basilio>, method=PLAIN, rip=10.2.8.108, lip=10.2.8.200, session=<JSc8vKIB4oAKAGhs>

ChatGPT Plus

¡Ahora los logs muestran claramente lo que pasa! 🧐

Hay **dos tipos de entradas distintas**:

1 SSL_accept() failed: ssl3 alert bad certificate

pgsql

```
imap-login: Disconnected: Connection closed: SSL_accept() failed:
error:0A000412:SSL routines::ssl3 alert bad certificate
```

- Esto sucede cuando el cliente intenta conectarse **usando SSL/TLS**, pero Dovecot **no tiene un certificado válido** o el cliente **no confía en el certificado**.
- **Importante:** este error ocurre **antes de cualquier intento de autenticación**, por eso no llega a passdb.

✅ Solución rápida:

- Para pruebas, usa **puerto 143 sin SSL/TLS**.
- O genera un certificado autofirmado y configúralo en Dovecot para TLS.

2 Autenticación fallida

pgsql

```
imap-login: Disconnected: Connection closed (auth failed, 3 attempts in 6 secs): user=<svb@mail.villegas.basilio>, method=PLAIN
```

- Aquí el cliente sí llega a enviar el login, pero Dovecot **rechaza la contraseña o no encuentra al usuario en LDAP**.
 - Como ya confirmaste con `doveadm auth test` que la autenticación funciona, esto **apunta a un problema de coincidencia entre %u y lo que el cliente envía**.
- ♦ Cosas a verificar:
1. El **usuario que envía el cliente** debe coincidir exactamente con `mail` en LDAP.
 - Ejemplo: `svb@villegas.basilio`
 - En los logs aparece: `svb@mail.villegas.basilio` → fíjate en ese `mail.` extra, eso **hace que falle**.
 2. Revisa la configuración del cliente:
 - Usuario: **exactamente** `svb@villegas.basilio`
 - Servidor IMAP: `10.2.8.200` (o `mail.villegas.basilio` si DNS/hosts funciona)
 - Puerto: `143` sin SSL/TLS para pruebas
 3. Si quieres usar SSL/TLS después, asegúrate de:
 - Certificado válido para `mail.villegas.basilio`
 - El cliente confíe en el certificado
 - Puerto IMAP `993` (IMAPS)

💡 **Clave:** El error “auth failed” se produce **porque el usuario enviado desde el cliente no coincide con %u en `user_filter/pass_filter`**.

Actualmente tus filtros LDAP son:

text

```
user_filter = (&(objectClass=posixAccount)(mail=%u))
pass_filter = (&(objectClass=posixAccount)(mail=%u))
```

- `%u` debe ser exactamente el correo en LDAP, sin `mail.` extra.
- Corrige la casilla de usuario en Thunderbird/Outlook para que coincida **exactamente**: `svb@villegas.basilio`.

Si quieres, puedo escribirte **una configuración de cliente completa paso a paso** para Thunderbird y Outlook, asegurando que use el correo completo, la IP correcta y que funcione con tu Dovecot + LDAP tal como está.

¿Quieres que haga eso?

Tú dijiste:

lo de mail lo estoy haciendo queriendo a ver si funciona, tambien he añadido eso al archivo de hosts. lo de ssl pasa porque la primera vez que se intenta conectar, se conecta por ssl. las demás se conecta por el 143 que es el que yo estoy poniendo. ç

ChatGPT Plus

¡Ah, perfecto! Entonces queda claro:

- El problema SSL inicial era solo la **primera conexión automática**, no afecta tus pruebas actuales.
- Ahora que estás usando puerto 143, **el handshake SSL ya no entra en juego**. ✓
- El verdadero detalle que nos queda es la **coincidencia exacta entre el usuario que envía el cliente y lo que Dovecot espera en LDAP**.

♦ Qué hacer ahora

1. Usa siempre el correo completo como usuario en el cliente:

graphql

```
villegas@villegas.basilio
svb@villegas.basilio
basilio@villegas.basilio
```

- Esto coincide con tu configuración en `dovecot-ldap.conf.ext`:

text

```
user_filter = (&(objectClass=posixAccount)(mail=%u))
pass_filter = (&(objectClass=posixAccount)(mail=%u))
```

2. Servidor IMAP en el cliente:

- IP directa: `10.2.8.200` → funciona seguro
- `O mail.villegas.basilio` → asegúrate de que resuelva correctamente en `/etc/hosts` del cliente

3. **Puerto IMAP:** 143 (sin SSL/TLS para pruebas)

4. **Contraseña:** la que asignaste en LDAP (usuario según tus cambios recientes)

♦ Verifica con Dovecot logs en tiempo real

bash

```
journalctl -u dovecot -f
```

- Cuando el cliente intente login correctamente, deberías ver:

pgsql

```
imap-login: Login: user=<villegas@villegas.basilio> method=PLAIN  
rip=IP_DEL_CLIENT lIP=10.2.8.200
```

- Si no ves eso → el cliente **no está llegando al servidor** (firewall, puerto, hostname).
-

💡 Resumen:

- Tu Dovecot ya autentica correctamente contra LDAP usando **correo completo**.
- Ahora el foco está en **configurar bien el cliente**: usuario exacto, servidor/IP, puerto correcto.
- Una vez funcione sin SSL/TLS, podrás luego habilitar SSL con un certificado válido y puerto 993.

Si quieres, puedo armarte **la configuración exacta de Thunderbird y Outlook lista para conectarse** a tu servidor LDAP+Dovecot sin errores de login ni SSL.

¿Quieres que haga eso?

Tú dijiste:

eso es lo que me sale con un solo intento: Oct 21 11:48:18 LDAP-Ubu auth[5181]:
ldapdb_canonuser_plug_init() failed in sasl_canonuser_add_plugin(): invalid parameter supplied Oct 21
11:48:18 LDAP-Ubu auth[5181]: _sasl_plugin_load failed on sasl_canonuser_init for plugin: ldapdb Oct
21 11:48:32 LDAP-Ubu dovecot[5062]: imap-login: Disconnected: Connection closed (auth failed, 3
attempts in 14 secs): user=<svb@mail.villegas.basilio>, method=PLAIN, rip=10.2.8.108, lip=10.2.8.200,
session=<hjWjxqlBBtoKAghs>

ChatGPT Plus


¡Perfecto, esto explica todo! 😊

Los logs muestran **dos problemas distintos**:

1 ldapdb_canonuser_plug_init() failed

SCSS

```
ldapdb_canonuser_plug_init() failed in sasl_canonuser_add_plugin(): invalid
parameter supplied
_sasl_plugin_load failed on sasl_canonuser_init for plugin: ldapdb
```

- Esto **no es un fallo de Dovecot/LDAP normal**.
- Es causado por un **plugin de SASL (ldapdb)** que **no tiene parámetros correctos o no se necesita**.
- Si **no estás usando SASL con LDAP para autenticación externa**, puedes ignorar estas líneas.
- La autenticación real la hace Dovecot vía `passdb ldap` y `userdb ldap`. 

2 auth failed por usuario no encontrado

pgsql

```
imap-login: Disconnected: Connection closed (auth failed, 3 attempts in 14
secs): user=<svb@mail.villegas.basilio>, method=PLAIN
```

- Aquí el cliente **envía `svb@mail.villegas.basilio`** como nombre de usuario.
- Tu LDAP tiene:

graphql

```
mail: svb@villegas.basilio
```

- Nota el **mail. extra en el cliente** → Dovecot no encuentra coincidencia en `pass_filter = (&(objectClass=posixAccount)(mail=%u))`.

♦ Qué hacer

1. En el cliente (Thunderbird/Outlook) asegúrate de que el **usuario sea exactamente**:

CSS

```
svb@villegas.basilio
```

- Sin el `mail.` que está apareciendo actualmente.
- Esto es la razón por la que el login falla, aunque Dovecot esté bien configurado.

2. Servidor: `10.2.8.200` (IMAP 143 para pruebas).

3. Contraseña: la que asignaste en LDAP.

💡 Resumen clave:

- La línea de `ldapdb_canonuser` es solo un warning de SASL, **no bloquea autenticación**.
 - El verdadero bloqueo es **coincidencia exacta del usuario**.
 - Dovecot ya funciona con `%u = correo completo`.
 - Ajusta el cliente y debería funcionar al primer intento.
-

Si quieres, puedo escribirte **el paso a paso exacto para configurar Thunderbird y Outlook** usando tus usuarios LDAP con correo completo para que ya no haya errores.

¿Quieres que haga eso?

Tú dijiste:

```
esto es lo que me salta de 1 solo registro: Oct 21 11:50:10 LDAP-Ubu dovecot[5062]: auth: Error:
ldap(svb,10.2.8.108,</ZlbzalBdtgKAghs>): user not found from userdb Oct 21 11:50:10 LDAP-Ubu
dovecot[5062]: imap(5201): Error: auth-master: login: request [1386610689]: Login auth request failed:
Authenticated user not found from userdb, auth lookup id=1386610689 (auth connected 2 msecs ago,
request took 2 msecs, client-pid=5200 client-id=1) Oct 21 11:50:10 LDAP-Ubu dovecot[5062]: imap-login:
Disconnected: Internal login failure (pid=5200 id=1): user=<svb>, method=PLAIN, rip=10.2.8.108,
lip=10.2.8.200, mpid=5201, session=</ZlbzalBdtgKAghs>
```

ChatGPT Plus

¡Ahora sí, esto lo deja totalmente claro! 🤖

El log dice:

pgsql

```
user not found from userdb
user=<svb>
```

♦ Qué está pasando

1. El cliente envía **solo svb** como usuario.
2. Tu `dovecot - ldap.conf.ext` está configurado así:

text

```
user_filter = (&(objectClass=posixAccount)(mail=%u))
pass_filter = (&(objectClass=posixAccount)(mail=%u))
```

- %u toma exactamente lo que envía el cliente (svb).
- En LDAP, el correo completo es:

graphql

```
mail: svb@villegas.basilio
```

- **No coincide**, así que Dovecot no encuentra al usuario en userdb → falla el login.

♦ Cómo solucionarlo

Tienes dos opciones:

Opción 1 Usar el correo completo como usuario en el cliente

- Usuario: svb@villegas.basilio
- Contraseña: la que asignaste en LDAP
- IMAP: 10.2.8.200 o mail.villegas.basilio
- Puerto: 143 (sin SSL/TLS para pruebas)

✓ Es la forma más limpia y segura.

Opción 2 Cambiar Dovecot para aceptar solo uid como login

Si realmente quieres que el cliente pueda poner solo svb:

1. Modifica dovecot-ldap.conf.ext:

text

```
user_filter = (&(objectClass=posixAccount)(uid=%u))
pass_filter = (&(objectClass=posixAccount)(uid=%u))
```

2. Guarda y reinicia Dovecot:

bash

```
systemctl restart dovecot
```

- Ahora %u será el uid en LDAP (svb, villegas, sergio, etc.)

- Entonces el login funcionará con solo el nombre de usuario, sin correo completo.

💡 **Nota:** Muchos clientes modernos obligan a poner un correo válido, por eso la **Opción 1 es la recomendada** si quieres que Thunderbird/Outlook funcione sin cambios extra.

Si quieres, puedo escribirte **la configuración completa de Dovecot lista para aceptar login con correo completo** que funcione con tus clientes tal cual, paso a paso.

¿Quieres que haga eso?

Tú dijiste:

```
root@LDAP-Ubu:/home# journalctl -u dovecot -f Oct 21 11:51:29 LDAP-Ubu dovecot[5060]: master:
Warning: Killed with signal 15 (by pid=5235 uid=0 code=kill) Oct 21 11:51:30 LDAP-Ubu systemd[1]:
dovecot.service: Deactivated successfully. Oct 21 11:51:30 LDAP-Ubu systemd[1]: Stopped
dovecot.service - Dovecot IMAP/POP3 email server. Oct 21 11:51:30 LDAP-Ubu systemd[1]:
dovecot.service: Consumed 1.529s CPU time, 8.7M memory peak, 0B memory swap peak. Oct 21
11:51:30 LDAP-Ubu systemd[1]: Starting dovecot.service - Dovecot IMAP/POP3 email server... Oct 21
11:51:30 LDAP-Ubu dovecot[5240]: master: Dovecot v2.3.21 (47349e2482) starting up for imap, pop3
(core dumps disabled) Oct 21 11:51:30 LDAP-Ubu systemd[1]: Started dovecot.service - Dovecot
IMAP/POP3 email server. Oct 21 11:51:35 LDAP-Ubu auth[5246]: ldapdb_canonuser_plug_init() failed in
sasl_canonuser_add_plugin(): invalid parameter supplied Oct 21 11:51:35 LDAP-Ubu auth[5246]:
_sasl_plugin_load failed on sasl_canonuser_init for plugin: ldapdb Oct 21 11:51:49 LDAP-Ubu
dovecot[5242]: imap-login: Disconnected: Connection closed (auth failed, 3 attempts in 14 secs): user=
<svb@villegas.basilio>, method=PLAIN, rip=10.2.8.108, lip=10.2.8.200, session=<Q/Vs0qlBeJEKAghs>
```

ChatGPT Plus

Perfecto, ahora el log nos dice claramente:

sql

```
user=<svb@villegas.basilio>
auth failed
```

♦ Interpretación

1. El usuario que envía el cliente es correcto (svb@villegas.basilio).
 2. Dovecot intenta autenticarlo contra LDAP, pero la autenticación falla.
 3. La línea de SASL ldapdb_canonuser sigue siendo solo un warning, **no afecta la autenticación real**.
-

♦ Qué revisar

Dado que `doveadm auth test svb@villegas.basilio` funciona, pero el cliente no, significa que **el problema no es LDAP**, sino la **configuración del cliente**:

1. **Usuario:** `svb@villegas.basilio`
 2. **Servidor IMAP:** `10.2.8.200` (no `villegas.basilio` si el DNS no resuelve)
 3. **Puerto:** 143 (sin SSL/TLS para pruebas)
 4. **Autenticación:** PLAIN o LOGIN
-

♦ Causas comunes por las que falla en el cliente

- Thunderbird/Outlook intenta conectar con SSL/TLS aunque pongas el puerto 143. → falla si no hay certificado válido.
 - Username mal escrito (por ejemplo, `svb@mail.villegas.basilio` en vez de `svb@villegas.basilio`).
 - Caché de credenciales previa del cliente → borrar contraseña guardada y volver a probar.
 - Firewall o NAT que interfiere con la conexión.
-

💡 Prueba rápida en cliente:

- Configura **sin SSL** (puerto 143)
- Usuario completo `svb@villegas.basilio`
- Contraseña: la que pusiste en LDAP (usuario)
- Servidor: `10.2.8.200`

Si funciona, entonces **el siguiente paso** será activar SSL/TLS correctamente (puerto 993) con certificado válido.

Si quieres, puedo prepararte **una guía paso a paso para configurar Thunderbird y Outlook** exactamente para tu servidor LDAP+Dovecot con usuarios tipo correo completo. Esto eliminará todos estos errores de login.

¿Quieres que haga eso?

Tú dijiste:

vale, literalmente con la configuración por defecto pero poniendo en el apartado nombre de la configuración avanzada "svb" me ha dejado

¡Ah, perfecto! 😊

Eso explica todo: tu Dovecot todavía estaba configurado para usar el **uid** como login, no el correo completo. Por eso:

- Con `%u = svb` → coincide con `uid=svb` en LDAP → autenticación correcta ✅
- Con `%u = svb@villegas.basilio` → antes no coincidía → fallaba

♦ Conclusión

- Dovecot ya funciona correctamente con LDAP.
- Puedes usar los usuarios **con su UID** (svb, villegas, sergio) o configurar Dovecot para aceptar el **correo completo** si prefieres.
- El warning de `ldapdb_canonuser` es solo un **mensaje de SASL**, no bloquea la autenticación.

💡 Tip para el futuro:

Si quieres que todos los clientes modernos (Thunderbird/Outlook) solo tengan que poner **correo completo**, solo habría que:

1. Cambiar `pass_filter` y `user_filter` en `/etc/dovecot/dovecot-ldap.conf.ext` para usar `mail=%u` en vez de `uid=%u`.
2. Reiniciar Dovecot.
3. Probar `doveadm auth test correo@villegas.basilio`.

Si quieres, puedo hacer un **resumen final con la configuración mínima funcional de Dovecot + LDAP** para que cualquier usuario pueda iniciar sesión sin complicaciones.

¿Quieres que haga eso?

ChatGPT puede cometer errores. Considera verificar la información importante. Ver [preferencias de cookies](#).