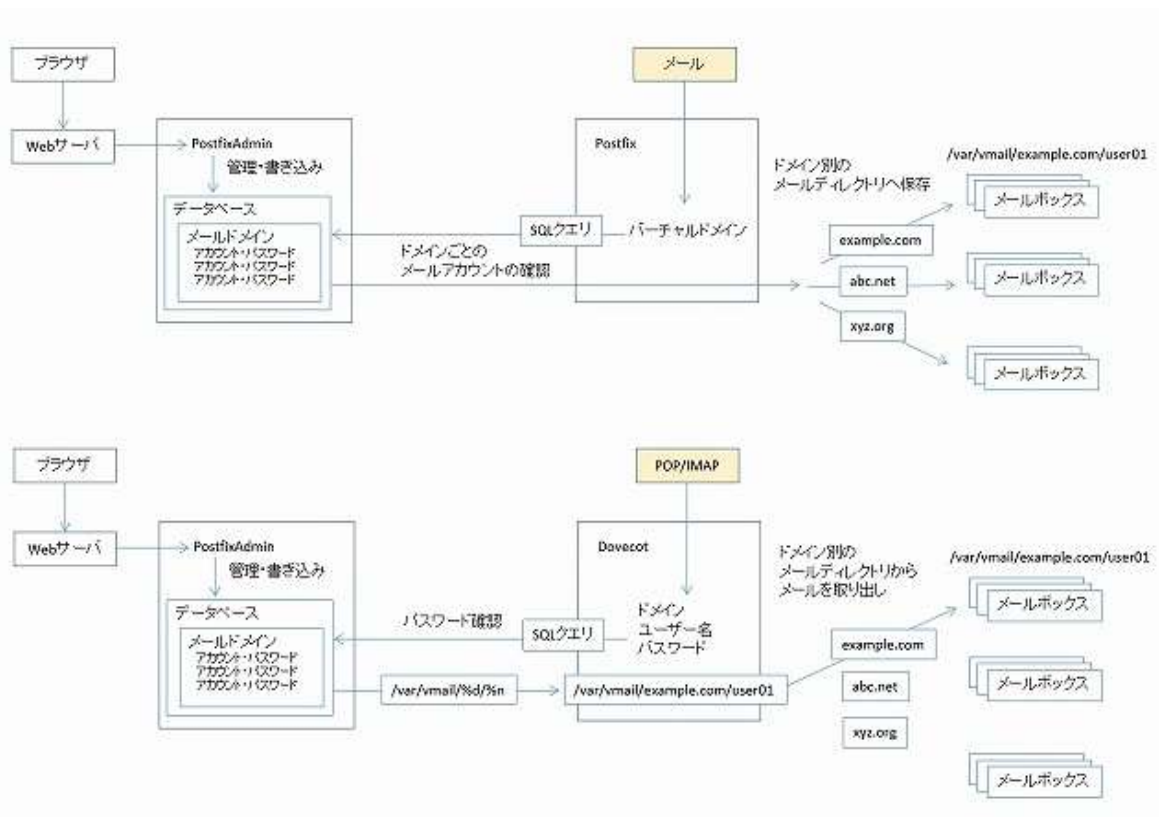


2023/06/24 RockyLinux 9.3

# PostfixAdminでPostfixとDovecotを管理する (MySQL編)

PostfixAdminでメールサーバを管理運用するには、PostfixおよびDovecotの設定変更が欠かせません。

PostfixAdminは、データベースを書き込み、実際の連携はPostfix/Dovecot側からデータベースを参照してもらう、という構成をとります。



このページでは別項「[PostfixAdminのインストール\(MySQL編\)](#)」が終わっているのを前提とし、PostfixとDovecotを設定変更してメールサーバを PostfixAdminと連携させます。

今環境の前提は

- PostfixAdminのデータベースはMySQLを使用している。
- PostfixAdminのインストールと初期化は完了している。
- SELinuxは無効化されている。

1. [PostfixAdminのインストール\(MySQL編\)](#)
2. PostfixAdminでPostfixとDovecotを管理する(MySQL編)
3. [PostfixAdminでクォータ管理をする\(MySQL編\)](#)

PostfixAdminの構築は、並行的に設定する必要があります。

「PostfixAdminのバーチャルドメインの定義」

「Postfixのバーチャルドメインへメール保存する設定」

「Dovecotのバーチャルドメインからメールを取り出す設定」

どれから行ってもいいですが、3つとも完了しないと動作試験もできません。

全体が関係しあっているので、うまく動作しないときは全体的を見直す必要のある点が構築を複雑にしています。

- [1.メールサーバの構築](#)
- [2.Postfixのバーチャルドメインの準備](#)
- [3.Postfixがバーチャルドメインのために使用するSQLクエリファイルを作成する](#)
- [4.PostfixAdmin上でバーチャルドメインの作成](#)
- [5.PostfixAdminからテストメールの送付](#)
- [6.DovecotをPostfixAdminへ連携させる](#)
- [7.動作試験](#)
- [8.デバッグログ](#)
- [9.メールのログインIDにフルアドレスを使いたくない](#)
- オマケ [PostfixAdminのCLI コマンド](#)

---

## 1.メール サーバの構築

PostfixAdminとDovecotをインストールして構成します。

構築トラブルを避けるため、いったんUNIXアカウントによってメールアドレスを管理する従来型のメールサーバを作ります。

RockyLinux9.2では、デフォルトでPostfixがインストールされていませんでした。

PostfixとDovecot、およびMySQLモジュールをインストールしておきます。

ついでにファイアウォールも開放しておきます。

```
#dnf install postfix dovecot -y
#dnf install postfix-mysql
#dnf install dovecot-mysql
#
#firewall-cmd --get-active-zone
public
    interfaces: enp0s3
#firewall-cmd --add-service=smtp --permanent
#firewall-cmd --add-service=pop3 --permanent
#firewall-cmd --reload
```

```
#firewall-cmd --list-service
cockpit dhcpv6-client http pop3 smtp ssh
```

Postfixの設定値の内、最小限の設定項目のみ例示します。

メール中継や送信先振り分けなどを指定したいなら、「[上位メールサーバへ一律転送するメールサーバ](#)」や「[宛先により配送先を振り分けるメールサーバ](#)」を参照してください。

/etc/postfix/main.cf の変更

```
mydestination = $myhostname,localhost.$mydomain,localhost, example.com(仮)
mynetworks = 192.168.1.0/24
inet_interfaces = all
inet_protocols = ipv4
```

ここでは送受信テストのため、mydestinationにexample.comドメインを追記していますが、これはテスト用で後で削除します。

**PostfixAdmin で運用したいドメインは、最終的にmydestinationへ記載されてはいけません**

mydestinationとの関係、これがバーチャルドメインでは重要なポイントです。

Postfixを起動させておきます。

```
#systemctl start postfix
#systemctl enable postfix
```

次にDovecotを構成します。まだDovecot単体で動作させるため、mboxで構成してます(後で変更します)

/etc/dovecot/dovecot.conf の変更

```
protocols = pop3 (とりあえずPOP3を記載、必要に応じてIMAP)
```

/etc/dovecot/conf.d/10-auth.conf の変更

```
disable_plaintext_auth = no
```

/etc/dovecot/conf.d/10-mail.conf の変更

```
mail_location = mbox:/var/empty:INBOX=/var/mail/%u:INDEX=MEMORY
```

```
/etc/dovecot/conf.d/10-ssl.conf の変更
```

```
#ssl = required
```

サービスを起動させます。

```
#systemctl start dovecot
#systemctl enable dovecot
```

これで、従来型のOSアカウントでメールアカウントを管理する、リアルドメインのメールサーバができました。

ここまでで、いったんメールの送受信テストをしておくで安心でしょう。

OS上の一般ユーザー宛にメールを送信し、Dovecotでメールが取り出せるところまでできて成功です。

現時点までの手順で構成されたメールサーバは、以下の前提となっています。

- ・メールアカウントは、OSのアカウントを参照している
- ・メールボックスはMBOX形式
- ・メールの保存先は/var/mail
- ・プロトコルはPOP3

---

## 2. Postfixのバーチャルドメインの準備

Postfixをバーチャルドメインで動作するよう設定を追加します。

PostfixAdminの運用は、バーチャルドメインが前提です。

バーチャルドメインでは、メールアドレスごとのUNIXアカウントが必要ない代わりに、全てのメールボックスの保存作業を一つのユーザーアカウントで行います。

そのため専用のユーザーアカウントを作成します。作成したアカウントのIDも確認しておきます。

```
# useradd -r -g mail vmmail
# usermod -d /var/vmail/ -s /sbin/nologin -G apache vmmail
# id vmmail
uid=974(vmmail) gid=12(mail) groups=12(mail),48(apache)
```

vmailのユーザID 974、mailのグループID 12。  
メモしておきます。このIDは後々、何度も必要となります。

バーチャルドメイン用のメールディレクトリには、/var/mailを流用するのではなく新たに/var/vmailディレクトリを作るのがオーソドックスのようです。

```
# mkdir /var/vmail/  
# chmod 775 /var/vmail/  
# chown vmail:mail /var/vmail/
```

このvmailディレクトリの下に、各ドメインのディレクトリが作られ、ユーザメールボックスが保存されます。

Postfixのmain.cfへ先ほど作成したvmailディレクトリと、確認した vmai ユーザーのUIDとGIDを指定します。

virtual\_mailbox\_baseも、virtual\_uid\_mapsもデフォルトでは存在しないと思いますので、追記します。

```
/etc/postfix/main.cf の編集  
  
#メール保存の基本ディレクトリの追加  
virtual_mailbox_base = /var/vmail  
  
#バーチャルドメインの使用アカウントの指定  
virtual_uid_maps = static:974  
virtual_gid_maps = static:12
```

---

### 3.Postfixがバーチャルドメインのために使用するSQLクエリファイルを作成する

Postfixへバーチャルドメインの設定を加えます。

といっても直接バーチャルドメイン情報を記述するのではなく、必要なデータはSQLクエリを使ってPostfixAdminデータベースから読み出す、という設定を行います。

今回想定PostfixAdminではMySQLを使用しているので、PostfixがMySQLを参照できるようにpostfix-sqlを確認します。

どのモジュールがインストール済みかは、コマンドで確認できます。

```
#postconf -m インストール済みのモジュールの確認
btree
cidr
environ
fail
hash
inline
internal
mysql
(略)
```

```
入ってなければ、PostfixとDovecotのMySQLモジュールのインストール
#dnf install postfix-mysql
#dnf install dovecot-mysql
```

前手順のPostfix、Dovecotの構築手順でインストール済みのはずですが、再度確認します。

次に、MySQLからバーチャルドメインのメールエイリアスに関する情報を問い合わせるためのSQLクエリファイルを準備します。

virtual\_mailbox\_domainsに指定するSQLクエリファイルです。

```
/etc/postfix/mysql_virtual_domains_maps.cf の作成

user = postfix
password = password123
hosts = localhost
dbname = postfix
query = SELECT domain FROM domain WHERE domain='%s' AND active = '1'
```

到着したメールがこのサーバ宛のメールか、そうでないかを判断します。mydestinationに似ます。

activeの確認があるので、そのドメインが有効化されている場合だけ着信処理します。

メールアドレスに関する情報を問い合わせるSQLクエリファイルを準備します。  
このSQLファイルはどちらも、virtual\_mailbox\_maps で使用されます。

```
/etc/postfix/mysql_virtual_mailbox_maps.cf の作成

user = postfix
password = password123
hosts = localhost
dbname = postfix
query = SELECT maildir FROM mailbox WHERE username='%s' AND active = '1'
```

```
/etc/postfix/mysql_virtual_alias_domain_mailbox_maps.cf の作成
```

```
user = postfix
password = password123
hosts = localhost
dbname = postfix
query = SELECT maildir FROM mailbox,alias_domain WHERE
alias_domain.alias_domain = '%d' and mailbox.username = CONCAT('%u', '@',
alias_domain.target_domain) AND mailbox.active='1' AND
alias_domain.active='1'
```

user01@example.com宛のメールをどこに保存するかをDBへ確認し、example.com/user01 という値が取得されます。

Postfixのコンフィグで「/var/vmail」ディレクトリがベースとして定義されているので、「/var/vmail/example.com/user01」が保存ディレクトリとして処理されます。これもactiveの確認があるので、もしメールアドレスが無効化されていればメールは受け付けません。

次の3つのファイルはvirtual\_alias\_mapsで使用します。  
メールアドレスの別アドレスへのマッピングに使われます。

/etc/postfix/mysql\_virtual\_alias\_maps.cf の作成

```
user = postfix
password = password123
hosts = localhost
dbname = postfix
query = SELECT goto FROM alias WHERE address='%s' AND active = '1'
```

/etc/postfix/mysql\_virtual\_alias\_domain\_maps.cf の作成

```
user = postfix
password = password123
hosts = localhost
dbname = postfix
query = SELECT goto FROM alias,alias_domain WHERE alias_domain.alias_domain =
'%d' and alias.address = CONCAT('%u', '@', alias_domain.target_domain) AND
alias.active='1' AND alias_domain.active='1'
```

/etc/postfix/mysql\_virtual\_alias\_domain\_catchall\_maps.cf の作成

```
user = postfix
password = password123
hosts = localhost
dbname = postfix
query = SELECT goto FROM alias,alias_domain WHERE alias_domain.alias_domain =
'%d' and alias.address = CONCAT('@', alias_domain.target_domain) AND
alias.active='1' AND alias_domain.active='1'
```

デフォルトでは「user01@example.comはuser01@example.comへ転送せよ」、とマッピングしてあるだけです。だからそのまま着信処理がされます。

他ドメインのメールボックスへ格納させたり、別のサーバへ転送したりするマッピングで使います。

activeの指定があるので、そのエイリアスが有効な場合だけ処理します。

Postfixのmain.cfを編集し、作ったファイルを指定します。

/etc/postfix/main.cf の編集

```
virtual_mailbox_domains =  
    mysql:/etc/postfix/mysql_virtual_domains_maps.cf  
  
virtual_mailbox_maps =  
    mysql:/etc/postfix/mysql_virtual_mailbox_maps.cf,  
    mysql:/etc/postfix/mysql_virtual_alias_domain_mailbox_maps.cf  
  
virtual_alias_maps =  
    mysql:/etc/postfix/mysql_virtual_alias_maps.cf,  
    mysql:/etc/postfix/mysql_virtual_alias_domain_maps.cf,  
    mysql:/etc/postfix/mysql_virtual_alias_domain_catchall_maps.cf
```

それぞれのマッピングの設定を、SQLite経由で問い合わせるようクエリファイルを指定しています。

次に重要な設定である、mydestinationの確認をします。

mydestinationには、バーチャルドメインで運用するドメインが含まれていては**いけません**。  
削除します。

/etc/postfix/main.cf の編集

```
mydestination = $myhostname,localhost.$mydomain,localhost , 削除(example.com)
```

mydestinationとバーチャルドメインの両方に同じドメインが記載してある場合、mydestinationが優先され、従来通りのリアルドメイン・UNIXアカウント準拠のメール受信が行われます。逆に言えば、1つのメールサーバ内でリアルドメイン運用からPostfixAdminのバーチャルドメイン運用へ移行しようとしている場合は、mydestinationの記述変更が切り替えるトリガーになります。

ここまででPostfixの設定の完了です。

Postfixを再起動すれば設定は全て有効になり、example.com宛のメールはPostfixAdminで作成したバーチャルドメインで受信できるようになります。



```
#systemctl restart postfix
```

Dovecotの設定はまだなので、メールを取り出すことはできません。

---

#### 4.PostfixAdmin上でバーチャルドメインの作成

PostfixAdminで、運用するメールドメインを作成します。

<http://サーバーアドレス/postfixadmin/login.php> ブラウザでアクセスし、PostfixAdminへログインします。

プロトコルがHTTPであることに注意です。



→ 192.168.1.64/postfixadmin/login.php

postfix.admin

こちらからドメインの管理ができます。

ログイン (メールアドレス):  
admin@example.local

パスワード:  
\*\*\*\*\*

言語: 日本語 - Japanese

ログイン

一般ユーザのログインはこちらをクリックしてください。

ドメイン一覧から、新しいドメインを選択します。



必要な情報を入力して、メールアドレスを作成します。

### 新しいドメインの追加

ドメイン	<input type="text" value="example.com"/>
説明	<input type="text"/>
転送数	<input type="text" value="10"/> -1 = 無効   0 = 無制限
アドレス数	<input type="text" value="0"/> -1 = 無効   0 = 無制限
バックアップ MXサーバ	<input type="checkbox"/>
アクティブ	<input checked="" type="checkbox"/>
デフォルト転送先の追加	<input checked="" type="checkbox"/>
Pass expires	<input type="text" value="0"/> Date when password will expire

画面上部のアドレス一覧からメールアドレスの追加を選択します。



必要な情報を入力して、メールアドレスを作成します。

Postfixの連携が正しくできていれば、「ようこそメール」が着信するはずです。

### 新しいメールアドレスの作成

メールアドレス	<input type="text" value="user01"/>
	<input type="text" value="example.com"/>
パスワード	<input type="password" value="....."/>
	POP3/IMAPのパスワード
パスワード (確認)	<input type="password" value="....."/>
名前	<input type="text"/>
	フルネーム
容量制限	<input type="text"/>
	MB
アクティブ	<input checked="" type="checkbox"/>
ようこそメールの送信	<input checked="" type="checkbox"/>

これでPostfixAdminのデータベース上に、バーチャルドメインの設定とメールアドレスが作成されました。

---

## 5.PostfixAdminからテストメールの送付

テストメールは、PostfixAdminの画面上部、「メール送信」からも送信できます。



`/var/log/maillog` を監視しつつ、テストメールを送信してみます。

### メール送信

差出人: admin@example.local

宛先:

件名:

本文:

/var/vmailディレクトリの中身を確認し、ドメイン名のディレクトリとユーザーディレクトリが作成されていれば成功です。

```
#ls /var/vmail  
example.com
```

```
#ls /var/vmail/example.com  
user01
```

バーチャルドメインはその性質上、問答無用でMailDir形式となります。

---

## 6.DovecotをPostfixAdminへ連携させる

PostfixAdminのDocuments/[DOVECOT.TXT](#)を参考に設定します。

Dovecotもバーチャルドメインのディレクトリに対応するようコンフィグを修正していきます。

```
/etc/dovecot/conf.d/10-mail.conf の編集
```

```
mail_location = maildir:/var/vmail/%d/%n

mail_uid = vmail
mail_gid = mail

mail_privileged_group = mail

first_valid_uid = 974
last_valid_uid = 974
```

first\_valid\_uidはデフォルトで1000になっています。

ということは、vmailユーザー(今回は974)は、明示的にログインできない設定になっています。

firstだけでもいいのですが、lastも指定するとDovecotにログインできるユーザーIDが絞られ、UNIXアカウントのユーザーは、メールを取り出すことができなくなります。(UNIXユーザーアカウントIDは通常1000以上)

設定ミスなどで、バーチャルドメイン以外のユーザーがログインできてしまうのを防げます。

次のはSMTP-Authの設定です。

社内で運用するメールサーバなど、SMTP-AUTHが不要なら追加する必要はありません。

Postfixが送信認証で、Dovecotの認証情報へアクセスする設定です。

```
/etc/dovecot/conf.d/10-master.conf の編集
```

```
service auth {

    # Postfix smtp-auth
    unix_listener /var/spool/postfix/private/auth {
        mode = 0666
        user = postfix
        group = postfix
    }

}
```

認証情報の問い合わせ先をOSから、SQLへ変更

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

```
#!/include auth-system.conf.ext #コメントアウト  
!include auth-sql.conf.ext #コメントイン
```

問い合わせファイルに、auth-sql.conf.extが指定されています。  
これは/etc/dovecot/conf.d/auth-sql.conf.extファイルを指定しています。

/etc/dovecot/conf.d/auth-sql.conf.extファイルの中を見えます。

```
/etc/dovecot/conf.d/auth-sql.conf.extの中身  
  
passdb {  
    driver = sql  
    args = /etc/dovecot/dovecot-sql.conf.ext  
}  
  
userdb {  
    driver = sql  
    args = /etc/dovecot/dovecot-sql.conf.ext  
}
```

コメントを外せばこれだけです。  
パスワードの問い合わせ、ユーザアカウントの問い合わせは、dovecot-sql.conf.extファイルを指定していることが分かります。

dovecot-sql.conf.extファイルを作成します。  
ユーザー情報、パスワード情報をPostfixAdminのデータベースから取り出す一番肝心なファイルです。  
今回はMySQLを使用しているので、接続ユーザー名、パスワードも指定してあります。

```
/etc/dovecot/dovecot-sql.conf.ext の作成  
  
connect = host=localhost dbname=postfix user=postfix password=password123  
driver = mysql  
  
password_query = \  
    SELECT \  
        username AS user, \  
        password \  
    FROM \  
        mailbox \  
    WHERE username = '%u' AND active='1'  
  
user_query = \  
    SELECT \  
        CONCAT('/var/vmail/', maildir) AS home \  
    FROM \  
        mailbox \  
    WHERE username = '%u' AND active='1'
```

```
iterate_query = SELECT username as user FROM mailbox WHERE active = '1'
```

connectの引数内でイコール前後にスペースを含めると接続に失敗してしまいます。  
dovecot-sql.conf.extファイルは、中身を変更したらDovecotをリロードするか再起動するまで有効にならないので注意です。

設定がDovecotの設定は以上です。サービスを再起動します。

```
#systemctl restart dovecot
```

もし上記SMTP-Authを設定している場合、/var/spool/postfix/private/authという連携ファイルが生成されています。

---

## 7.動作試験

メールクライアントでPOPサーバ、SMTPサーバを指定して、メールの送受信試験を行います。  
バーチャルドメインなのでPOPのログインIDには「**user01@example.com**」とフルアドレスを指定します。

※もし既存のユーザー環境に対応するという意味で「POP3のログインアカウントはフルドメインを指定せず、user01というユーザー名だけでログインしたい」という要件がある場合、dovecot-sql.conf.extファイルを書き換えることで当面の対応は可能です。  
(「9.メールのログインIDにフルアドレスを使いたくない」で後述します)

メールクライアントで送受信試験をするか、telnetで接続試験をします。

```
#telnet localhost 110
+OK Dovecot ready.
>user user01@example.com
+OK
>pass user01passwd
+OK Logged in.
>list
+OK 1 messages:
1 653
.
>retr 1
```



```
ようこそ  
これはテストメールです。
```

これで、バーチャルドメインの受信が確認できました。

バーチャルドメインへの着信が確認でき、バーチャルドメインからメールを取り出すことが出来れば、PostfixAdminによる管理は成功していると言えます。

クォータの設定を必要としているなら、[PostfixAdminでクォータ管理をする\(MySQL編\)](#)に続きます。

ここまでの手順で正常に動作しない、特に取り出しに失敗する場合は、以下のデバッグログから原因を追いかけることになるでしょう。

---

## 補足情報

---

## 8.デバッグログ

一発で正常動作すればいいのですが、PostfixAdminは幅広く設定をするので、設定ミスをしやすく、ミスした箇所を特定するのも手間がかかります。適切な調査方法が必要です。

### コンフィグ調査コマンド

```
Postfixのコンフィグパラメータを全て表示/変更部分だけ表示  
#postconf  
#postconf -n
```

```
Dovecotのコンフィグパラメータを表示  
#dovecot -n 1
```

### Dovecotのデバッグログモード

```
#vi /etc/dovecot/local.conf  
  
auth_verbose = yes  
auth_debug = yes  
  
#systemctl restart dovecot
```

このデバッグログ状態でメールの送受信をすれば、`/var/log/maillog`に詳細なログが残ります。SQLクエリの結果がどうなっているのか、どこのディレクトリへメールを取りに行っているかを確認できて、とても便利です。

例えば、「**Dovecotに正常にログインできているのに、新着メールが発見できない**」というトラブルが発生したとします。

メール取り出しなのでDovecotが原因だと思われるます。切り分けのためデバッグログを有効にします。

```
/etc/dovecot/local.conf の編集
```

```
auth_verbose = yes
auth_debug = yes
```

```
#systemctl restart dovecot
```

それから再度メールを受信してログを出力します。

```
/var/mail/maillog
```

```
(略)
```

```
auth-worker(4039): Debug: mysql(localhost): Finished query 'SELECT
CONCAT('/var/mail/', maildir) AS home FROM mailbox WHERE username =
'user01@example.net' AND active='1'' in 0 msecs
```

よく見ると、メールがあるはずの`/var/vmail`ではなく、デフォルトの`/var/mail`へアクセスしているのが確認できます。

どこかの設定項目が誤っていると判断できます。

Dovecotのコンフィグコマンドを確認し、`/var/mail`が残っていないかを確認します。

```
#dovecot -n 1 | grep /var/mail
(出力なし)
#dovecot -n 1
(略)
userdb {
  args = /etc/dovecot/dovecot-sql.conf.ext
  driver = sql
}
```

コンフィグに`/var/mail`の記述は見つかりませんでした。

`userdb`に`dovecot-sql.conf.ext`ファイルが指定されているので、そちらも確認します。

パスの指定ミスを見つけました。修正します。

```
/etc/dovecot/dovecot-sql.conf.ext の作成
(略)
user_query = \
    SELECT \
        CONCAT('/var/mail/', maildir) AS home \
    FROM \
        mailbox \
    WHERE username = '%u' AND active='1'
```

/var/vmailと/var/mailの中も確認します。どちらもexample.com/user01がありました。

```
#ls /var/mail/
example.com Postfixがメール着信で作成した正しいディレクトリ
#ls /var/mail/example.com
user01

#ls /var/vmail
example.com Dovecotがログインで作成した空のディレクトリ
#ls /var/vmail/example.com
user01
```

Dovecotはログインに成功した時に、ディレクトリを作成してしまうので、エラーではなく「メールボックスが空である」という動作をしていたので、発見が遅れました。  
/var/mail/example.comのほうは、削除しておきましょう。

こういう感じでデバッグログを使って切り分けていきます。  
もちろん、調査が終わったら、デバッグモードはコメントアウトして、戻しておきましょう。

---

## 9.メールのログインIDにフルアドレスを使いたくない

PostfixAdminはテナント型、つまりレンタルサーバのように複数の人が共用でメールサーバを運用するスタイルを想定していると思われます。  
そのためログインドメインを判断するためPOPのユーザーIDはuser01@example.comのような、フルアドレスを使用します。

しかしPostfixAdminはメールアカウントの管理が簡単にできるので、ドメインを1つしか運用しない企業でも用いられます。

ログインIDがuser01@example.comといった形式に変更されると、既存のメールクライアント全て設定変更する必要があります。

切り替え作業を減らしたいなど、当面はメールクライアントが使用するログインIDは「user01」のままにしておきたい、という要望もあります。

そういう場合はdovecot-sql.conf.extのクエリを変更して一時しのぎできます。

/etc/dovecot/dovecot-sql.conf.ext の編集

```
connect = host=localhost dbname=postfix user=postfix password=postfix
driver = mysql
```

#SQLクエリの変更箇所

```
password_query = \
    SELECT \
        username AS user, \
        password \
    FROM \
        mailbox \
    WHERE username = '%u' AND active='1'
```

**変更箇所 ↓** ドメイン名を追加し、%uを%nとしています

```
WHERE username = '%n@example.com' AND active = '1'
```

```
user_query = \
    SELECT \
        CONCAT('/var/vmail/', maildir) AS home \
    FROM \
        mailbox \
    WHERE username = '%u' AND active='1'
```

**変更箇所 ↓** ドメイン名を追加し、%uを%nとしています

```
WHERE username = '%n@example.com' AND active = '1'
```

```
iterate_query = SELECT username as user FROM mailbox WHERE active = '1'
```

**ポイントはドメイン名を追加し、%uを、%nとすることです。見落としやすいです。**

SQLファイルの変更後は、Dovecotをリロードするか再起動しないと反映されません。

こうすることで、ユーザがuser01というログインIDを送信してきた場合、SQLクエリの中でuser01@example.comという文字列へ変更してくれます。

もし今後、PostfixAdminで別のバーチャルドメインを運用しなくてはいけない時が来たら、そのときの選択肢は二つです。

- ・SQLクエリを元へ戻し、社内のメールクライアントの設定を全て変更する。
- ・このSQLクエリをさらに改造して、「ログインIDが@を含んでいなかったらexample.comと見なし、@を含んでいたらそのまま利用する」というcase文を組み込む。

これで当座の対応ができます。

ポイントは、メールクライアントからログインID user01を使用しても、user01@example.comを使用しても、サーバ上では%nにより、user01として判断されます。

現在のパソコンの設定変更することはやめて、これから新規導入するパソコンのメール設定時に

user01@example.comというログインIDを指定するように指定すれば、いずれ全てのパソコンがログインID user01@example.comになっていくはずです。

---

## オマケ PostfixAdmin のCLIコマンド

PostfixAdminはCLIが用意されています。  
GUIでやりづらい作業はコマンドを流して対応できます。

```
# cd /srv/postfixadmin/scripts
```

コマンドオプションの表示

```
# ./postfixadmin-cli help
# ./postfixadmin-cli mailbox help
# ./postfixadmin-cli mailbox add help
```

メールアドレスの新規作成(所有者名user05、クォータ100MB、ようこそメール送付)

```
# ./postfixadmin-cli mailbox add user05@example.com --password user05p@ss --password2 user05p@ass --name user05 --quota 100 --welcome-mail
```

クォータサイズの変更

```
# ./postfixadmin-cli mailbox update user01@example.com --quota 300
```

メールアカウントを無効化

```
# ./postfixadmin-cli mailbox update user01@example.com --active 0
(ドメインの最大容量制限があるとコマンドで警告が出ますが、無効化には影響ありません)
```

メールアカウントの削除

```
# ./postfixadmin-cli mailbox delete user06@example.com
```

