

学習目標

この授業を受講すると，電子メールの基本的な仕組みと，メールサーバの役割を理解し，Postfixを用いて基本的なメールサーバの構築が出来るようになります．具体的には次のようなことが出来るように学習しましょう．

- 電子メールの基本的な仕組みを説明できる
- Postfixによるメールサーバの設定ができる
- DovecotによるIMAP, POP3サーバの設定ができる
- telnetコマンド，メールソフト(ThunderBird) で構築したメールサーバの動作確認を行うことができる

電子メールの仕組みとメールサーバの役割

電子メールは既に一般的な通信手段の一つとして広く普及しています。電子メールの特徴としては、次のような点があげられるでしょう。

- 迅速である(通常すぐに届く)
- 非同期である(電話のように相手を拘束しない)
- データを添付できる(写真、ワープロ・表計算ソフトのデータなどの様々な電子データを添付できる)

電子メールがどのように配信されるのか、その仕組みについて見ていきましょう。電子メールをメーラ(ThunderBirdなど)から送る場合を考えます。メーラには、自分のメールアドレスの他に、メールを送信するためのサーバ(SMTPサーバ)とメールを受信するためのサーバ(POP、IMAPなどのサーバ)を設定します。なお、単にメールサーバという場合は、SMTPサーバを指すのが一般的です。

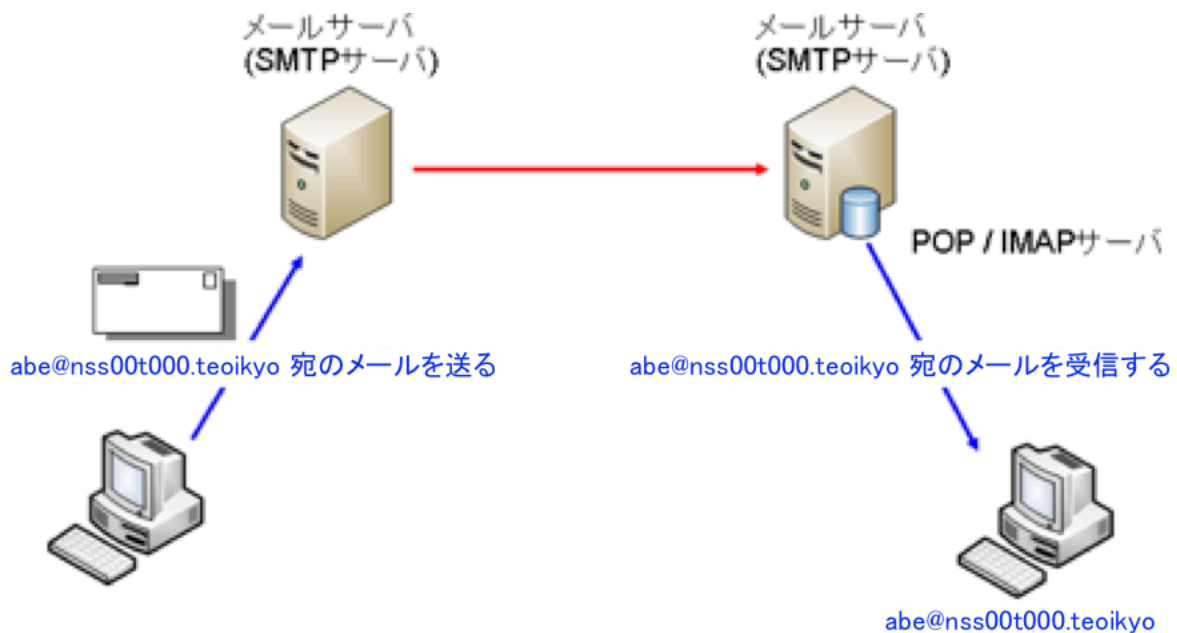


図1 電子メールが送られる仕組み

たとえば、図1のように abe@nss00t000.teoikyo 宛にメールを送る場合、送信者のメーラは、設定されている送信サーバ(SMTPサーバ)に送られます。メールを受け取ったメールサーバ(SMTPサーバ)は、メールを受け取る側のメールサーバ(SMTPサーバ)に対してメールを送ります。送られてきたメールを受け取ったメールサーバは、ユーザ(メールアカウント)毎に用意されているメールスプール(メールを保存する場所)にメールを保存します。

メールを受け取るユーザ(abe@nss00t000.teoikyo)は、自分のメーラに設定してあるメール受信サーバ(POP/IMAPサーバ)に接続して、自分のメールスプールから、送られてきたメールを取得します。ここで出てきた、「SMTP」、「POP」、「IMAP」は、メールを送受信する際に使われるプロトコルです。

SMTP(Simple Mail Transfer Protocol)は、メールサーバ間でメールを転送するときに使われるプロトコルです。

POP(Post Office Protocol)とIMAP(Internet Message Access Protocol)は、メーラが、メールスプールから送られてきたメールを取得するときに使われるプロトコルです。

POPは、メールスプールにあるメールをすべてメーラに送ります。POPを使う場合は、受信したメールはすべてメーラソフトの入ったクライアント(自分のPCなど)に保存します。

一方のIMAPは、サーバ上にメールを残したまま、メールの一覧を取得してから、読みたいメールの情報(本文、添付ファイルなど)だけを指定して取得します。

POPを使う場合は、メーラソフトの入ったクライアント以外ではメールを読むことができません。しかし、一度取得したメールは、ネットワークに接続されていなくても読むことができます。

これに対してIMAPでは、メールはサーバに入ったままです。不特定のクライアントから読みたいときに読みたいメールを取得することが可能です(Webメールなど)。しかし、Webメールなどは、ネットワークに接続されていなければメールを読むことができません。

(IMAPでもダウンロードしたメールを保存しておけば、読むことができます。)

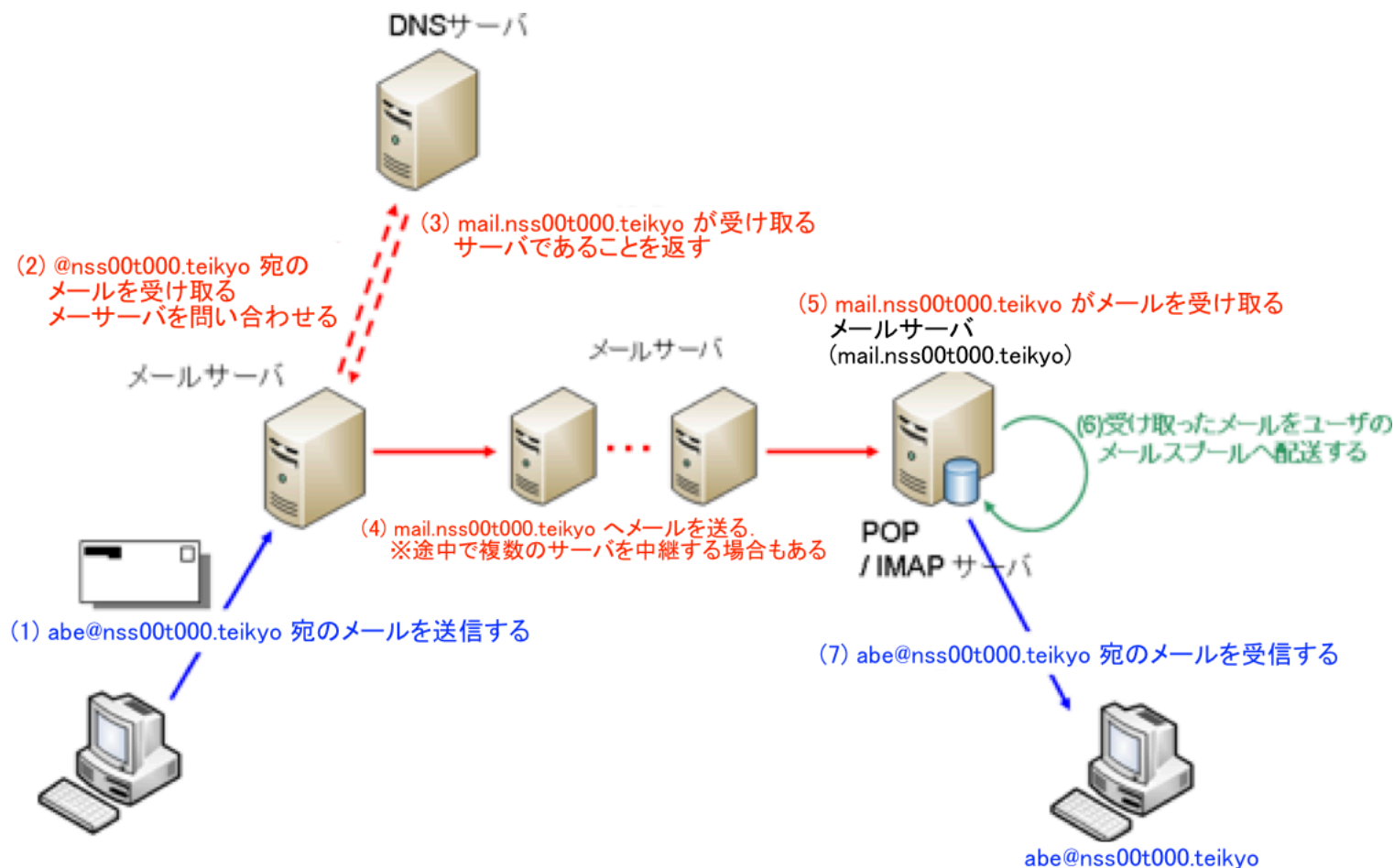


図2 電子メールが送られる詳しい手順

電子メールが送られる手順をもう少し詳しく見ていきましょう。図2のように、**abe@nss00t000.teikyo** 宛にメールを送る場合を考えます。メールを受け取るサーバは **mail.nss00t000.teikyo** というホストであるとします。

(1) まず、メーラがメールサーバへメールを送信します。

(2) メールサーバは、メールアドレスからメールを受信するサーバの情報を取得します。

メールアドレスの@ (アットマーク) より右側のドメイン名が、メールサーバのFQDNであった場合(図2の例では、メールの宛先が **abe@mail.nss00t000.teikyo** となっていた場合は、そのメールサーバのIPアドレスをDNSサーバに問い合わせます。

また、@ の右側がホストではなくドメインを指している場合もあります(図2の **abe@nss00t000.teikyo** の **nss00t000.teikyo** はドメイン(ゾーン)を表しています)。メールでこのようなドメインが指定された場合に、どのサーバへメールを送信したらいいかを、DNSに問い合わせます。DNSには、ドメインとメールサーバを対応させる機能が実装されています。ドメインとメールサーバの対応付けはMX(Mail eXchange)レコードで行います。MXレコードの登録方法については、後で説明します。

(3) DNSサーバは、問い合わせのあったホスト、あるいはドメインのメールサーバのIPアドレスを返します。

(4) メールサーバは、取得したIPアドレス宛てにメールを送信します。メールを送信する際には、複数のサーバが中継する場合もあります。

(5) 最終的に、メールを受け取るメールサーバ(図2では **mail.nss00t000.teikyo**)がメールを受信します。(4)のメールの送信、転送および(5)のメールの受信に用いられるのがSMTPというプロトコルです。また、メールをメールサーバ間で転送(送信、中継、受信)をするプログラムをMTA(Mail Transfer Agent)といいます。

(6) メールサーバは、受信したメールをユーザのメールスプールへ保存します。受信したメールをメールスプールに保存するプログラムをMDA(Mail Delivery Agent)といいます。

(7) メールを受信するユーザは、メールスプールに保存されたメールを、メーラソフトで受信します。メールスプールからメールを取り出す際に用いられるのがPOPやIMAPというプロトコルです。

メールソフトからのPOPやIMAPでの要求に応じて、メールスプールからメールを取り出してメールへ送るサービスを提供するのがPOPあるいはIMAPサーバです。

※POPやIMAPサーバは(6)のMDAとは異なりますので間違えないようにしてください。(1)や(7)で用いられるメールソフトでは、メールの作成、MTAへの送信、POP/IMAPサーバからの受信、受信したメールの表示などを行います。このようなメールソフトをMUA(Mail User Agent)といいます。

この授業では、MTAとしてPostfixを用います。PostfixはMDAとしても機能します。また、POP,IMAPサーバとしては、dovecotを用います。MUAとしては、POPクライアントとしてThunderbirdを用います。

DNSの設定

DNSには、メールアドレスの@の右側のドメイン名から、そのメールを転送すべきサーバを取得するための情報を登録しておくことができます。ここで使われるのがMX(Mail eXchange)レコードです。電子メールのアドレスにサーバのホスト名ではなくドメイン名を使いたい場合は、DNSにMXレコードを登録しておく必要があります。

WebminによるMXレコードの設定

WebブラウザからWebmin(https://localhost:10000/)を起動します。Webminのインデックス画面から、「サーバ」タブを開き、その中の「BIND DNSサーバ」の設定画面を開きます。そこから「既存のDNSゾーン」にある、ゾーンの設定画面を開きます。ここでは例として「nss00t000.teikyo」ドメインの設定画面を開きます。(図3)



図3 「既存のDNSゾーン」にある、ゾーンの設定画面を開く

図4のような「マスターゾーンの編集」画面が開いたら，MXレコードを登録するため「メールサーバ」の設定画面を開きます．



図4 「メールサーバ」の設定画面を開く

図5のような「メールサーバレコード」の設定画面が開きますので，「レコード名」の欄に，メールアドレスで用いるドメイン名(ここではnss00t000.teikyo)を入力します．また，「メールサーバ」の欄に，レコード名に入力したドメイン名がメールアドレスになっているメールを受け取るサーバ(ここではこのサーバ)のドメイン名(ここではsvr01.nss00t000.teikyo)を入力します．「優先度」では，同じドメインへのメールを受信するサーバを複数登録した場合に，どちらのサーバに優先的に送るかを指定します．数字の小さい方が，優先度が高くなります． 入力が終わったら，「作成」ボタンを押します．



図5 「svr01.nss00t000.teikyo」を「nss00t000.teikyo」ドメインのメールサーバとして登録する

画面が図6のように変わり，nss00t000.teikyoドメインのメールサーバとしてsvr01.nss00t000.teikyo が登録されたことが分かります．



図6 「svr01.nss00t000.teikyo」が「nss00t000.teikyo」ドメインのメールサーバとして登録された

MXレコードの登録が終わったら、「BIND DNSサーバ」の設定画面に戻り、「変更を適用」ボタンを押します (図7)。これで登録した別名が有効になります。



図7 「変更を適用」して登録した別名を有効にする

設定内容の確認

登録したMXレコードはゾーンファイルに登録されています。viエディタなどでゾーンファイル(ここでは /var/named/ /nss00t000.teikyo.hosts)を開くと、図8のような内容となっています。MXレコードで「nss00t000.teikyo.」ドメインのメールサーバとして「svr01.nss00t000.teikyo.」が、優先度10で登録されています。

```
root@svr01:~
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
$ttl 38400
nss00t000.teikyo.      IN      SOA      svr01.nss00t000.teikyo. 00t000xx.stu.teikyo-u.ac.jp. (
                        1594199237
                        10800
                        3600
                        604800
                        38400 )
nss00t000.teikyo.      IN      NS       svr01.nss00t000.teikyo.
svr01.nss00t000.teikyo. IN      A        10.0.2.100
pc01.nss00t000.teikyo. IN      A        10.0.2.121
pc02.nss00t000.teikyo. IN      A        10.0.2.122
www.nss00t000.teikyo.  IN      CNAME    svr01
vhost.nss00t000.teikyo. IN      CNAME    svr01
nss00t000.teikyo.      IN      MX        10 svr01.nss00t000.teikyo.
~
~
~
~
~
~
~
"/var/named/nss00t000.teikyo.hosts" 14L, 436C
```

図8 ゾーンファイル(nss00t000.teikyo.hosts)に登録されたMXレコード

動作確認

登録したMXレコードで、ドメインのメールサーバが取得できるかどうかを、digコマンドで確認してみます。
プロンプトから次のようにdigコマンドを実行します。

```
dig @(DNSサーバのFQDNまたはIPアドレス) (ドメイン) MX
```

DNSサーバがsvr01.nss00t000.teikyo、ドメインがnss00t000.teikyoの場合は次のようなコマンドを実行します。

```
dig @svr01.nss00t000.teikyo nss00t000.teikyo MX
```

digコマンドで「nss00t000.teikyo」のMXレコードを問い合わせた結果を図9に示します。MXレコードから「nss00t000.teikyo。」のメールサーバが「svr01.nss00t000.teikyo。」であり、優先度が10であることが得られています。

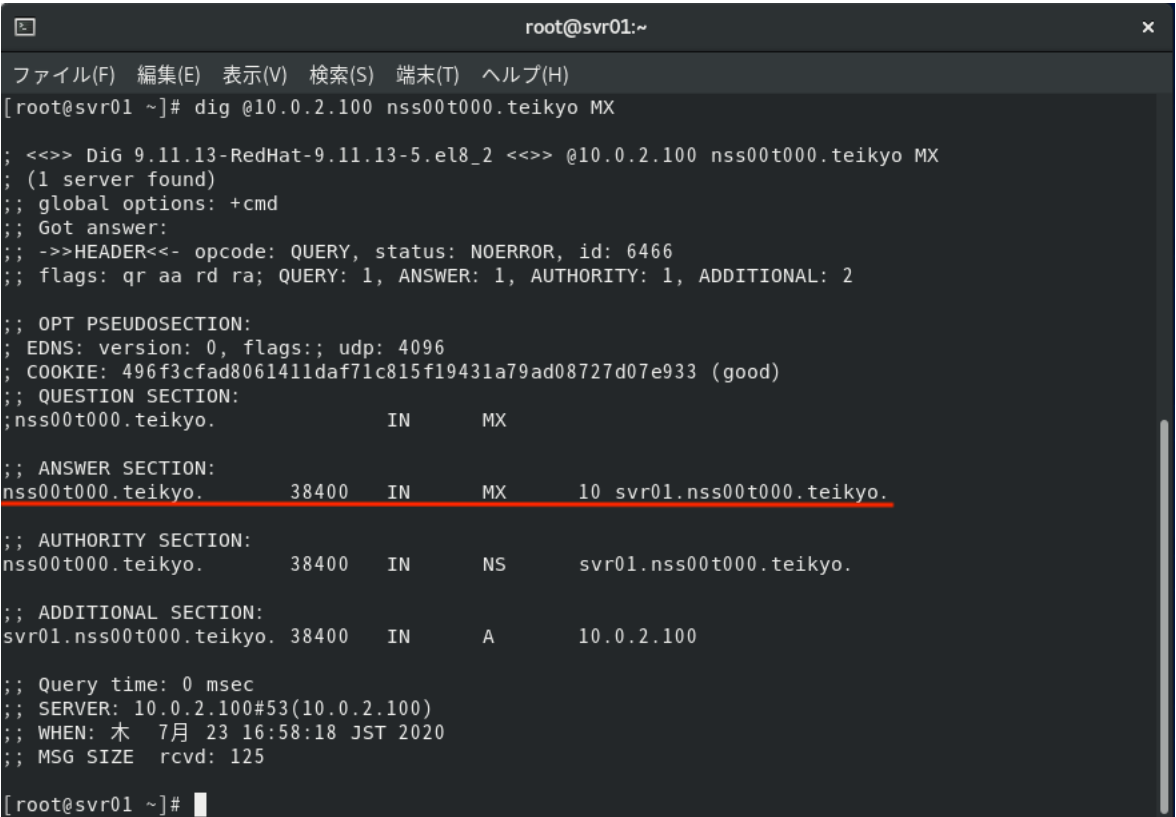


図9 digコマンドで「nss00t000.teikyo」のMXレコードを問い合わせた結果

Postfixによるメールサーバの構築

まず , PostfixによりMATとしてPostfixによるメールサーバを構築していきます . PostfixおよびDovecotもWebminから設定します .

Webminによる設定

Webブラウザを起動し , Webmin(<https://localhost:10000/>)を開きます . Webminのメニューの「サーバ」を開き , その中の「Postfixの設定」画面を開きます(図14) . 「Postfixメールエージェント」の設定画面が開きます . このうちの「一般オプション」と「ローカル配信」の設定画面を開いて設定します .



図14 「Postfixの設定」の「一般オプション」と「ローカル配信」を設定する

図14の「一般オプション」をクリックすると , 図16に示す「一般オプション」の設定画面が開きます . メールサーバの主な設定はこの設定画面から行います . 次の項目を設定します .

- 発信メールで使用するドメイン : 「ドメイン」を選択
- メールを受信するドメイン : localhostの後ろに「, \$mydomain」を追加する(localhostと","で区切る)
- メール受信用のネットワークインターフェース : 「すべて」を選択
- このメールシステムのインターネットホスト名 : このサーバのホスト名(ここではsvr01.nss00t000.teikyo)
- ローカルインターネットドメイン名 : 指定されたドメイン名(ここではnss00t000.teikyo)

☆ 一般オプション

最も便利な一般オプション

発信メールで使用するドメイン

☐ ホスト名を使用

☒ ドメイン名を使用

メールを受信するドメイン

☐ ローカル マシン

☒ ドメイン全体

\$myhostname, localhost.\$mydomain, localhost, \$mydomain

デフォルト Selected below

bounce - Bounced mail

2bounce - Double-bounced mail

delay - Delayed mail

policy - Policy rejected clients

protocol - Client protocol errors

resource - Resource problems

software - Software problems

ポストマスターに通知する問題

その他の一般オプション

ホスト経由でメールを送信

☒ 直接配信

各メッセージの bcc を受信するアドレス

☒ なし

要請を処理するタイムアウト

18000s

デフォルト データベース タイプ

hash

デフォルト メッセージ 配信トランスポート

smtp

バウンス メール の 差出人 アドレス

double-bounce

キュー ディレクトリ の 下の サブ ディレクトリ レベル 数

1

サブ ディレクトリ に 渡って 分割 する キュー ディレクトリ 数

deferred, defer

最大 Received: ヘッダ 数

50

無配信の警告を送信するまでの時間

☒ Postfix default

メール受信用のネットワーク インターフェイス

☒ すべて

☐ localhost

内部 IPC クライアントが切断後のアイドル時間

5s

内部通信チャンネルの入出力タイムアウト

3600s

メール システム 名

Postfix

メールの所有者

postfix

オフィシャル メール システム バージョン

3.3.1

次のサービス要請の待機時間

100s

終了前に処理されるサービス要請の最大数

100

このメール システムのインターネット ホスト 名

☐ デフォルト (システムにより供給)

☒ svr01.nss00t000.teikyo

ローカル インターネット ドメイン 名

☐ デフォルト (システムにより供給)

☒ nss00t000.teikyo

ローカル ネットワーク

☒ デフォルト (すべてネットワークに接続)

☐

自動ローカルネットワーク

Same IP subnet

ポストマスターに bounce (バウンス) の通知を送信...

☒ デフォルト

☐

ポストマスターに 2bounce (倍のバウンス) の通知を送信..

☒ デフォルト

☐

ポストマスターにエラーの通知を送信...

☒ デフォルト

☐

メール キュー ディレクトリ

/var/spool/postfix

ポストマスターに遅延の通知を送信...

☒ デフォルト

☐

ロック ファイル ディレクトリ、キュー ディレクトリに相対

pid

ユーザ名/アドレス拡張子のセバレータ

再配置済みマッピング検索テーブル

☒ デフォルト (無効)

☐

メールボックスでカーネルファイルのロックを無効

☐ はい

☒ いいえ

Emailコンテンツフィルタ

☒ 無し

☐

デーモンにトリガーを送信する最大時間

10s

保存して適用

Postfix 設定 に戻る

図16 「一般オプション」での設定内容

「発信メールで使用するドメイン」は、メールを送信する際に自分のメールアドレスの@(アットマーク)以降につけるドメインを指定します。メールサーバのホスト名を付けるか、このサーバが属するドメイン名(FQDNからホスト名を取ったもの)を付けるか、あるいは指定したドメインを用いるかを指定します。

※通常は、メーラなどで送信者のメールアドレスを指定します。そうすると、メーラで指定した送信者のメールアドレスがそのまま発信元となるため、この情報は使われません。

※「**発信メールで使用するドメイン**」の”localhost”が設定したサーバ名”svr01”となっている場合がありますが、どちらでも問題ありません。

「メール受信用のネットワークインターフェース」は、外部から送られてくるメールを受信するネットワークインターフェースのアドレスを指定します。CentOSのデフォルトではlocalhostが指定されていますが、これだと外部からのメールは受信できないため、「すべて」に変更します。

「このメールシステムのインターネットホスト名」は、メールサーバのホスト名を含んだドメイン名を指定します。デフォルトではシステムから取得する設定になっていますが、授業の環境では取得できていないようなので、ここで設定します。

「ローカルインターネットドメイン名」は、このメールサーバの属するドメインを指定します。デフォルトではシステムから取得する設定になっていますが、これも授業の環境では取得できていないため、設定します。すべて入力したら、「保存して適用」ボタンを押します。

図14の「ローカル配信」をクリックすると、図16aに示す「ローカル配信」の設定画面が開きます。次の項目を設定します。

- ユーザーメールボックスファイルのホーム相対パス名：「Maildir/」を入力

☆ ローカル配信

ローカル配信

ローカル配信用のトランスポート名

☒ ローカル

☐

外部コマンドへの配信に使用するシェル

☒ なし (コマンドのダイレクト実行)

☐

転送にリストを検索

\$home/.forward\${recipient_delimiter}\${extension}, \$home/.forward

外部コマンドに有効なメール配信

alias, forward

外部ファイルに有効なメールを配信

alias, forward

配信エージェントのデフォルトの権利

nobody

ユーザーメールボックスファイルのホーム相対パス名

☐ なし (標準の /var/something を使用)

☒

Maildir/

不明の受信者用の宛先アドレス

☒ なし (バウンス メール)

☐

スプールディレクトリ

/var/mail

メールボックス配信の代わりに使用する外部コマンド

オプションの実際に使用するトランスポート

☒ なし

☐

不明な受信者用のオプションのトランスポート

☒ なし

☐

同一ローカル受信者へのパラレル配信の最大数

2

各ローカルメッセージ配信ごとの受信者の最大数

☒ デフォルト

☐

次の場合に Delivered-To: を前に付加...

☒ デフォルト

☐

保存して適用

← Postfix 設定に戻る

図16a 「ローカル配信」での設定内容

Webminによる設定が終わったら，Postfixを起動します．図17の「Postfixメールエージェント」の設定画面の一番下にある「Postfixを開始」ボタンをクリックします．

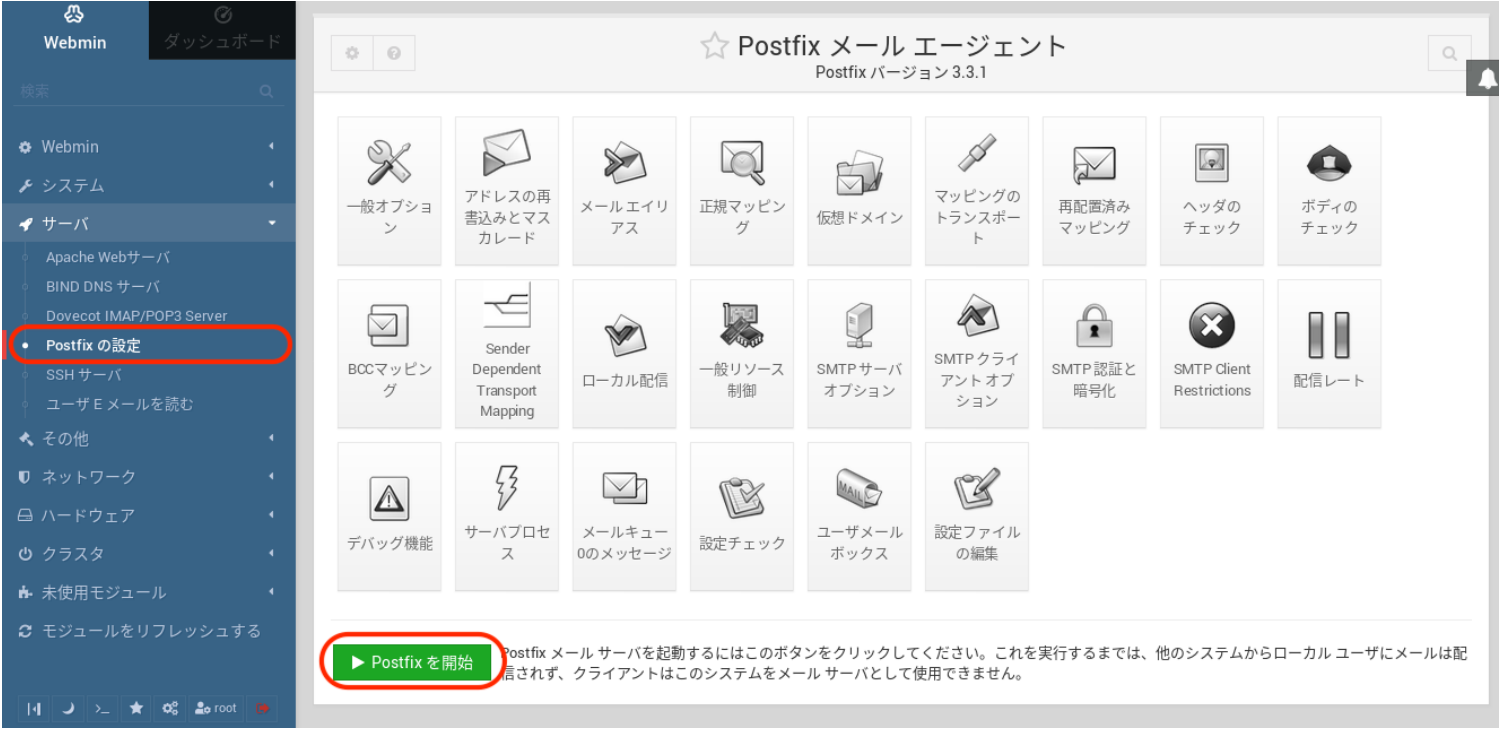


図17 「Postfixを開始」ボタンをクリックして，Postfixを起動する

設定内容の確認

Postfixの設定は， /etc/postfix/ディレクトリ内に保存されています．Webminから行った設定は，main.cfファイルに書き込まれます．main.cfファイルの内容をviエディタなどで開いて設定内容を確認します．図18にmain.cfの内容の一部を示します．

```

      .
      .
      .

# RECEIVING MAIL

# The inet_interfaces parameter specifies the network interface
# addresses that this mail system receives mail on. By default,
# the software claims all active interfaces on the machine. The
# parameter also controls delivery of mail to user@[ip.address].
#
# See also the proxy_interfaces parameter, for network addresses that
# are forwarded to us via a proxy or network address translator.
#
# Note: you need to stop/start Postfix when this parameter changes.
#
#inet_interfaces = all
#inet_interfaces = $myhostname
#inet_interfaces = $myhostname, localhost
#
# Enable IPv4, and IPv6 if supported
inet_protocols = all

# The proxy_interfaces parameter specifies the network interface
# addresses that this mail system receives mail on by way of a
# proxy or network address translation unit. This setting extends
# the address list specified with the inet_interfaces parameter.
#
# You must specify your proxy/NAT addresses when your system is a
# backup MX host for other domains, otherwise mail delivery loops
# will happen when the primary MX host is down.
#
#proxy_interfaces =
#proxy_interfaces = 1.2.3.4

      .
      .
      .

# Specify a list of host or domain names, /file/name or type:table
# patterns, separated by commas and/or whitespace. A /file/name
# pattern is replaced by its contents; a type:table is matched when
# a name matches a lookup key (the right-hand side is ignored).
# Continue long lines by starting the next line with whitespace.
#
# See also below, section "REJECTING MAIL FOR UNKNOWN LOCAL USERS".
#
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain,
# mail.$mydomain, www.$mydomain, ftp.$mydomain

# REJECTING MAIL FOR UNKNOWN LOCAL USERS
#
# The local_recipient_maps parameter specifies optional lookup tables
# with all names or addresses of users that are local with respect
# to $mydestination, $inet_interfaces or $proxy_interfaces.
#
# If this parameter is defined, then the SMTP server will reject
# mail for unknown local users. This parameter is defined by default.
#

      .
      .
      .
```


・
・
・

```
# Use TLS if this is supported by the remote SMTP server, otherwise
# plaintext (opportunistic TLS outbound).
#
smtp_tls_security_level = may
meta_directory = /etc/postfix
shlib_directory = /usr/lib64/postfix
mynetworks_style = subnet
mydomain = nss00t000.teikyo
myorigin = $mydomain
myhostname = svr01.nss00t000.teikyo
home_mailbox = Maildir/
```

図18 /etc/postfix/main.cf の内容(一部)

- inet_interfaceは「メール受信用のネットワークインターフェイス」で設定した内容です。Webminからの設定では「localhost」となっていた設定を「すべて」に変更しました。「すべて」はPostfixのデフォルトの設定(inet_interfaceでオプションを設定しなかったとき有効になる設定)なので、inet_interfaceの設定がすべてコメントアウトされて無効になっています。(デフォルトのallが有効となる)
- mydestinationは、「メールを受信するドメイン」で設定した内容です。最後に、Webminからの設定で追加した\$mydomainが追加されています。(localhostでなくsvr01となっていた場合は、svr01と表示されます)
- mydomainは、「ローカルインターネットドメイン名」で設定したこのサーバの属するドメインです。
- myoriginは、「発信メールで使用するドメイン」で設定した内容です。ドメイン名を使用を選択したため、\$mydomainが設定されています。\$は、(2)で設定したmydomainの内容を参照する(変数として利用する)ための記号です。
- myhostnameは、「このメールシステムのインターネットホスト名」で設定したこのサーバのホスト名(ホストを含んだFQDN)です。
- home_mailboxは、「ローカル配信」で設定したユーザメールボックスファイルのホーム相対パス名(Maildir/)です。

動作確認

Postfixを設定して起動したら，Postfixが正しく動作しているかどうかを確認してみましょう．

図19にtelnetコマンドによりSMTPのポートに接続してSMTPのコマンドを打ってみます．試しに，以前に作成したユーザ kato から abe 宛てにメールを送信してみましょう．図19の赤の下線部分を入力します．

```
[root@svr01 ~]# telnet svr01.nss00t000.teikyo smtp (1)
Trying 10.0.2.100...
Connected to svr01.nss00t000.teikyo.
Escape character is '^]'.
220 svr01.nss00t000.teikyo ESMTP Postfix
helo svr01.nss00t000.teikyo (2)
250 svr01.nss00t000.teikyo
mail from:<kato@nss00t000.teikyo> (3)
250 2.1.0 Ok
rcpt to:<abe@nss00t000.teikyo> (4)
250 2.1.5 Ok
data (5)
354 End data with <CR><LF>.<CR><LF>
hello postfix! (6)
. (7)
250 2.0.0 Ok: queued as 312D0207111
quit (8)
221 2.0.0 Bye
Connection closed by foreign host.
[root@svr01 ~]#
```

(1)telnetコマンドでPostfixサーバのSMTPポートに接続します．プロンプトからtelnetコマンドを次のように実行します．

```
telnet (接続先のサーバのアドレスorFQDN) smtp
```

図19の例では，telnetに続けて接続先のサーバ(ここではsvr01.nss00t000.teikyo)を指定し，その後で接続するポート(ここではsmtpの53番ポート)を指定しています．telnetコマンドを実行すると，図19のように「Trying 10.0.2.100...」などのメッセージが表示されます．メッセージの4行目に「220 svr01.nss00t000.teikyo ESMTP Postfix」というメッセージが返ってきています．これがPostfixから返ってきたSMTPの応答メッセージです．先頭の「220」は，SMTP接続が確立され，準備ができたという応答コードです．

(2)メッセージが止まったら，次のようなコマンドを打ちます．

```
helo (サーバのFQDN)
```

heloはSMTPセッションを開始するコマンドです．heloに続けて，サーバのFQDN(ここではsvr01.nss00t000.teikyo)を記述します．Postfixは「250 svr01.nss00t000.teikyo」というメッセージを返しています．「250」はSMTPのコマンドの実行が成功したことを返す応答コードです．

(3) 続けて、mail コマンドを打ちます。

```
mail from:<送信元のメールアドレス>
```

「mail from:<送信元のメールアドレス>」はメールの送信者を指定するコマンドです。アドレスは" "<">"で囲んで指定します。ここでは、kato@nss00t000.teikyo を指定しています。mail コマンドの実行後に、「250」で始まる応答が返されています。

(4) 続けて、rcpt コマンドを打ちます。

```
rcpt to:<送り先のメールアドレス>
```

「rcpt to:<送り先のメールアドレス>」は、メールの送り先を指定するコマンドです。ここでは、abe@nss00t000.teikyo を指定しています。

(5) の「data」は、メールのデータを送信するコマンドです。Postfixからの応答の「354」は、メールの入力開始を知らせています。(6)のようにメール本文を入力し(複数行入力できます)、入力が終わったところで(7)のように"."(ピリオド)だけの行を入力します。これが送信データの入力が終わったことを表しています。(7)の次の行では、メールが送信のキューに送られたことを返しています。

(8)入力が終了したら「quit」と入力して、SMTP接続を切ります。

Postfixがメールを送信したことが、ログファイルに記録されていますので確認してみましょう。/var/log/maillog ファイルをviエディタなどで開くと、ファイルの最後に図20のように、Postfixからのログが記録されています。

```
Jul 23 19:02:56 svr01 postfix/smtpd[95110]: connect from svr01[10.0.2.100]
Jul 23 19:03:40 svr01 postfix/smtpd[95110]: 312D0207111: client=svr01[10.0.2.100]
Jul 23 19:04:03 svr01 postfix/cleanup[95123]: 312D0207111: message-id=<20200723100340.312D0207111@svr01.nss00t000.teikyo>
Jul 23 19:04:03 svr01 postfix/qmgr[85216]: 312D0207111: from=<kato@nss00t000.teikyo>, size=348, nrcpt=1 (queue active)
Jul 23 19:04:03 svr01 postfix/local[95124]: 312D0207111: to=<abe@nss00t000.teikyo>, relay=local, delay=40, delays=40/0.02/0/0, dsn=2.0.0, status=sent (delivered to maildir)
Jul 23 19:04:03 svr01 postfix/qmgr[85216]: 312D0207111: removed
Jul 23 19:04:07 svr01 postfix/smtpd[95110]: disconnect from svr01[10.0.2.100] helo=1 mail=1 rcpt=1 data=1 quit=1 commands=5
```

図20 メールログ(/var/log/maillog)の内容を確認する

- サーバ(ここでは192.168.0.10)に接続しています。
- kato@nss00t000.teikyoからメールが送信されています。
- abe@nss00t000.teikyo宛てにメールが送信されています。
- 「status=sent (delivered to mailbox)」となり、メールがメールボックスに送信されて、メールの送信が成功したことが分かります。

動作確認

送信されたメールは、ユーザのメールスプールに保存されます。メールスプールディレクトリは、ユーザのホームディレクトリ内の Maildir ディレクトリです。メールを受け取ったユーザ **abe** のメールは、`/home/abe/Maildir/` ディレクトリ内のファイルに保存されます。viエディタなどで `/home/abe/Maildir` ディレクトリ内のファイル(図21では1595498643.Vfd00I207b19M303673.svr01)を開いてみると、図21のような内容が表示されます。

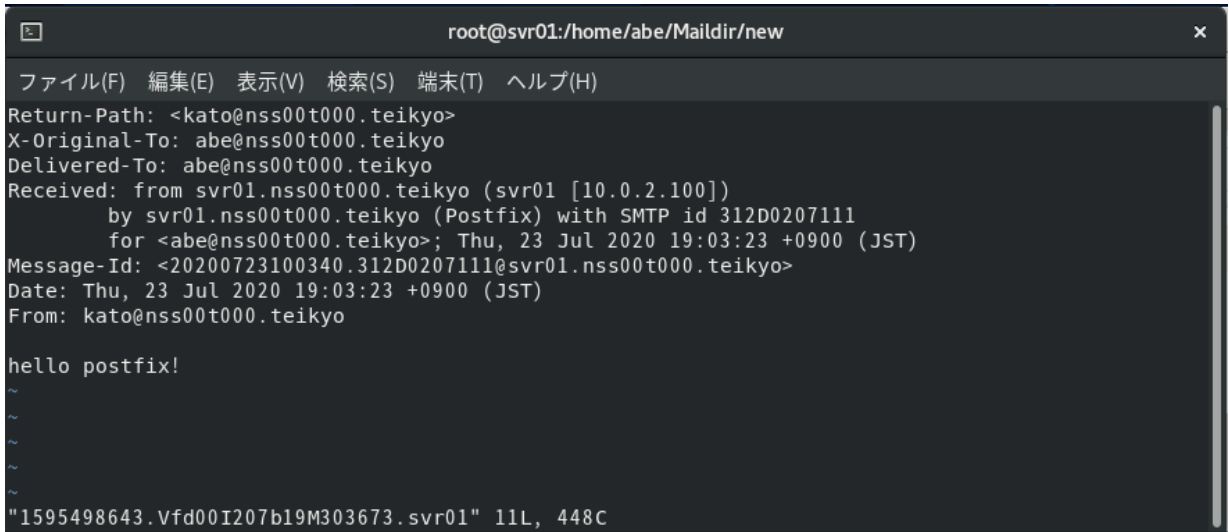


図21 ユーザ **abe** のメールスプールの内容

kato@svr01.nss00t000.teikyo から、abe@nss00t000.teikyo 宛てに、「hello! postfix!」という内容のメールが送られていることが分かります。

このように、Postfixはメールの転送と、メールスプール(メールボックス)へのメールの転送を行います。すなわち、MTAとMDAの役割を果たしたことが分かります。

DovecotによるPOP/IMAPサーバの設定

Postfixにより，送られてきたメールを受信して，メールスプールへ保存するところまでの設定はできました．しかし，Outlook Expressなどのメーラや，Webメールなどでメールを受信するためには，メールスプールからメーラなどへメッセージを取り出すためのサービスを設定する必要があります．これらのサービスを提供するのがPOPやIMAPです．

この章では，POP /IMAPサーバの設定を行います．CentOSには，POP/IMAPサーバとしてDovecot(ダブコット)がインストールされていますので，これを使うことにします．

Webminによる設定

Webminのインデックス画面の「サーバ」メニューの中から，「Dovecot IMAP/POP3 Server」を選択し，「Networking and Protocols」と「SSL Configuration」の設定画面から設定します(図22)．

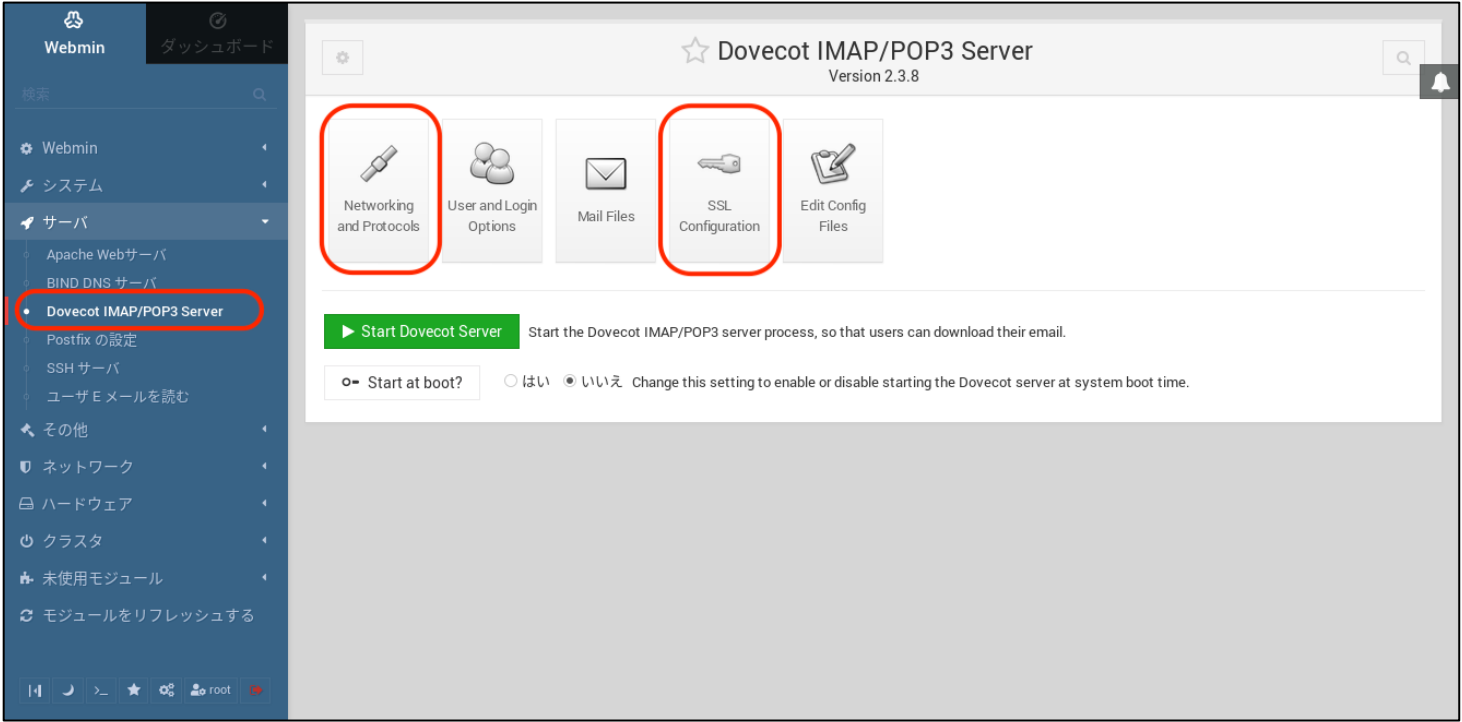


図22 「Dovecot IMAP/POP3 Server」から「Networking and Protocols」を開く

図23bの「Networking and Protocols」の設定画面では，次の項目を設定します．(なお，この授業ではSSLで接続する設定は行いません．また，IPv6も設定しません．)

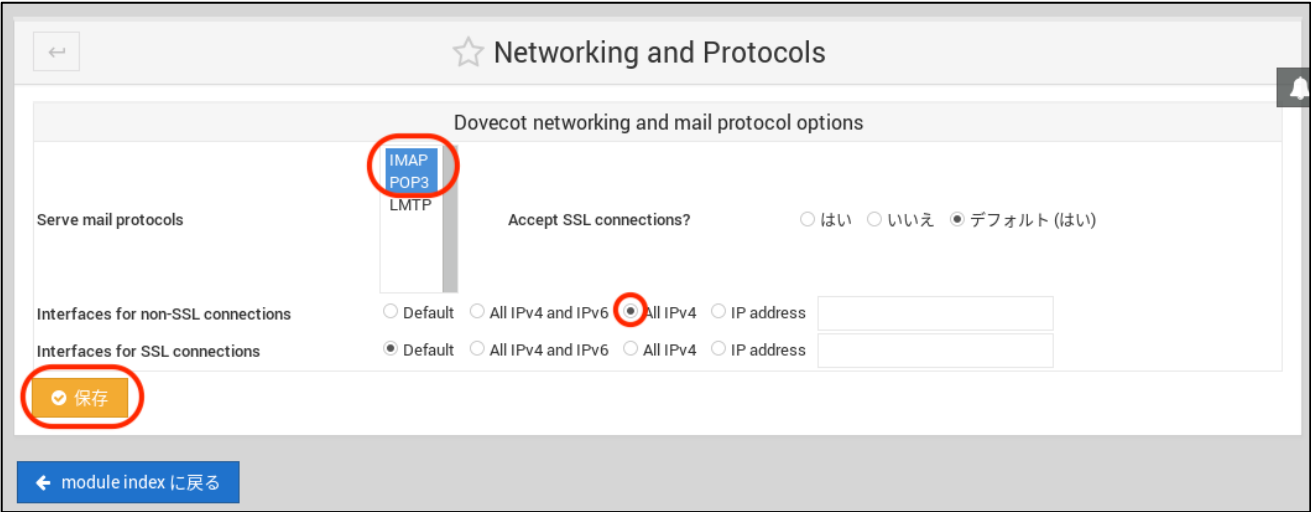


図23b 「Networking and Protocols」における設定

- Server mail protocols：「IMAP」と「POP3」を選択します
- Interface for non-SSL connections：「All IPv4」を選択します

「Server mail protocols」では，Dovecotが提供するサービスを選択します．ここでは，IMAPサーバおよびPOP3サーバとして設定しています．

「Interface for non-SSL connections」では，Dovecotサーバが接続に接続することができるインターフェースの種類やIPアドレスを設定します．ここでは，すべてのIPv4を設定しています．

図23cの「SSL Configuration」の設定画面では，次の項目を設定します．

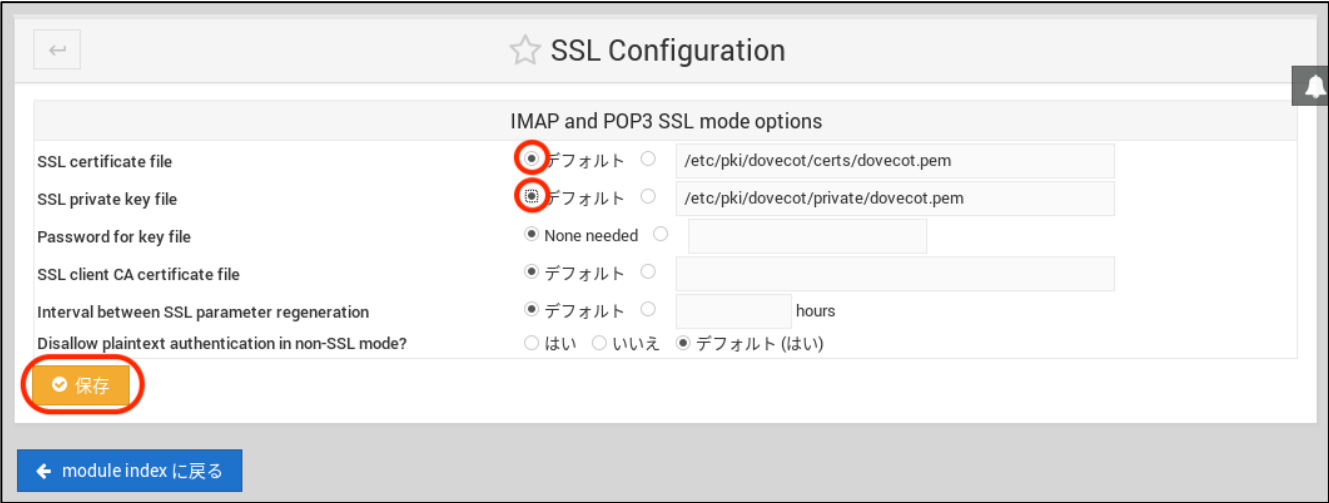


図23c 「SSL Configuration」における設定

- SSL certificate file：「デフォルト」に設定
- SSL private key file：「デフォルト」に設定

入力が終わったら「保存」ボタンを押します．すると「Dovecot IMAP/POP3 Server」の設定画面に戻りますので，「Start Dovecot Server」ボタンを押して，Dovecotサーバを起動します．

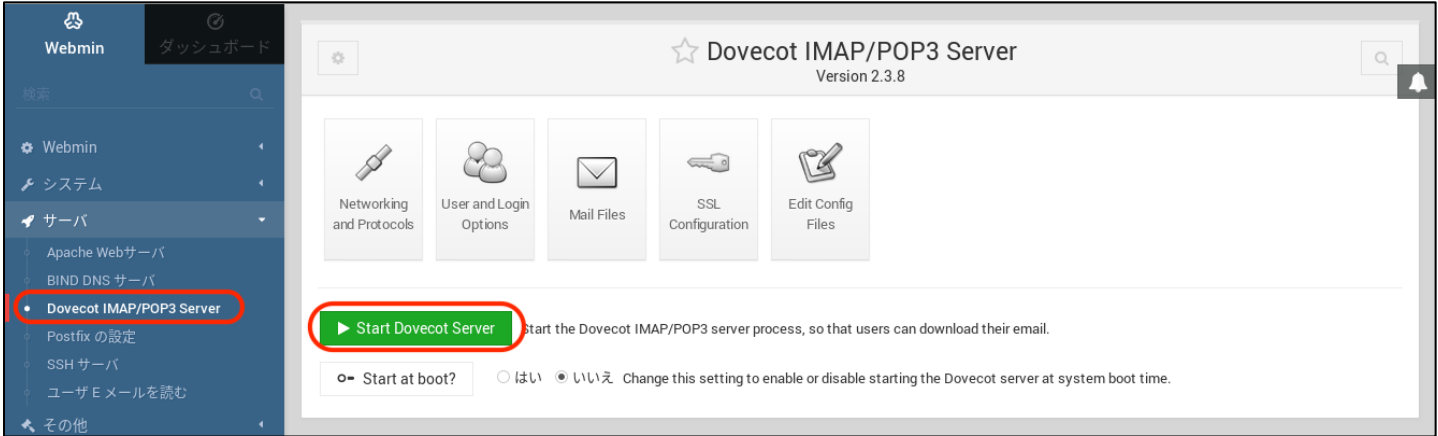


図24 「Start Dovecot Server」ボタンを押して，Dovecotを起動する

ここで , PostfixおよびDovecotのサービスがサーバ起動時に開始されるように設定します .
※PostfixおよびDovecotについては, テキスト作成時点でメニューの「システム」の中にある「起動およびシャットダウン」からの設定がうまくいかないようなので , [端末]から"systemctl"コマンドで設定します (図25) .

```
systemctl enable (サービス名).service
```

```
[root@svr01 ~]# systemctl enable postfix.service  
Created symlink /etc/systemd/system/multi-user.target.wants/postfix.service → /usr/lib/systemd/system/postfix.service.  
[root@svr01 ~]# systemctl enable dovecot.service  
[root@svr01 ~]#
```

図25 [端末]から**systemctl**コマンドで**Postfix**と**Dovecot**がサーバ起動時開始されるように設定する

- Postfixを起動時に開始する : systemctl enable postrix.service
- Dovecotを起動時に開始する : systemctl enable dovecot.service

設定内容の確認

Dovecotの設定内容は、`/etc/dovecot/dovecot.conf` に保存されています。viエディタなどで開いてみると、図26のような内容が表示されます。

「`protocols`」は、Webminで設定した「Server mail protocols」の内容です。imapとpop3が設定されています。

「`listen`」は、「Interface for non-SSL connections」の設定内容です。「`*`」は「All IPv4」を表しています。「All IPv6 and IPv4」なら「`:::`」と設定します。

```

      .
      .
      .
# Default values are shown for each setting, it's not required to uncomment
# those. These are exceptions to this though: No sections (e.g. namespace {})
# or plugin settings are added by default, they're listed only as examples.
# Paths are also just examples with the real defaults being based on configure
# options. The paths listed here are for configure --prefix=/usr
# --sysconfdir=/etc --localstatedir=/var
#
# Protocols we want to be serving.
#protocols = imap pop3 lmtp submission
protocols = imap pop3
#
# A comma separated list of IPs or hosts where to listen in for connections.
# "*" listens in all IPv4 interfaces, ":::" listens in all IPv6 interfaces.
# If you want to specify non-default ports or anything more complex,
# edit conf.d/master.conf.
#listen = *, ::
listen = *
#
# Base directory where to store runtime data.
#base_dir = /var/run/dovecot/
#
# Name of this instance. In multi-instance setup doveadm and other commands
# can use -i <instance_name> to select which instance is used (an alternative
# to -c <config_path>). The instance name is also added to Dovecot processes
# in ps output.
#instance_name = dovecot
      .
      .
      .

```

図26 Dovecotの設定ファイル(`/etc/dovecot/dovecot.conf`)の内容

動作確認

Dovecotが動作しているかどうか，telnetコマンドで確認してみましょう．POP3およびIMAPポートにそれぞれ接続して見ます．

POP3

まず，pop3のポートに接続します．ユーザ abe のメールスプールの内容を取得してみます．図27の下線の部分を入力します．

```
[root@svr01 ~]# telnet svr01.nss00t000.teikyo pop3 (1)
Trying 10.0.2.100...
Connected to svr01.nss00t000.teikyo.
Escape character is '^]'.
+OK Dovecot ready.
user abe (2)
+OK
pass teikyo2020 (3)
+OK Logged in.
list (4)
+OK 1 messages:
1 459
.
retr 1 (5)
+OK 459 octets
Return-Path: <kato@nss00t000.teikyo>
X-Original-To: abe@nss00t000.teikyo
Delivered-To: abe@nss00t000.teikyo
Received: from svr01.nss00t000.teikyo (svr01 [10.0.2.100])
        by svr01.nss00t000.teikyo (Postfix) with SMTP id 312D0207111
        for <abe@nss00t000.teikyo>; Thu, 23 Jul 2020 19:03:23 +0900 (JST)
Message-Id: <20200723100340.312D0207111@svr01.nss00t000.teikyo>
Date: Thu, 23 Jul 2020 19:03:23 +0900 (JST)
From: kato@nss00t000.teikyo

hello postfix!
.
quit (6)
+OK Logging out.
Connection closed by foreign host.
[root@svr01 ~]#
```

図27 telnetコマンドでサーバのPOP3ポートに接続する

(1)telnetコマンドでDovecotサーバのPOP3ポートに接続します．プロンプトからtelnetコマンドを次のように実行します．(ここではsvr01.nss00t000.teikyo に接続しています．)

```
telnet (サーバのアドレスorFQDN) pop3
```

telnetコマンドを実行すると，いくつかメッセージが表示されたあと，「+OK Dovecot ready」というメッセージが返ってきます．これがDovecatから返ってきたPOP3の応答メッセージです．エラーがなければ「+OK」，エラー場ある場合は「-ERR」から始まるメッセージが返ってきます．

(2)「+OK Dovecot ready.」が表示されてメッセージが止まったら，次のようなコマンドを打ちます．

```
user (ユーザ名)
```

userは認証のためのユーザ名を指定するコマンドです．ここでは，ユーザとして abe を指定しています．

(3) 「+OK」 が返ってきたら，次のようなコマンドを打ちます．

```
pass (パスワード)
```

passは認証のためのパスワードを指定するコマンドです．ここでは，ユーザ abeのパスワードを指定しています．

(4) 次に，list コマンドを実行します．list コマンドは，メールのメッセージ番号と，メッセージのサイズを返します．

(5) list 命令で取得したメッセージを表示するのが retr コマンドです．retr に続けてメッセージ番号を指定すると，指定したメッセージを取得できます．

(6) quit 命令は，ログアウトしてPOP3接続を終了するためのコマンドです．quit 命令を実行すると，POP3 サーバとの接続を切断してプロンプトに戻ります．

IMAP

次に，IMAPのポートに接続します．ユーザ `abe` のメールスプールの内容を取得してみます．図28の下線の部分を入力します．

```
[root@svr01 ~]# telnet svr01.nss00t000.teikyo smtp ※ imapの確認をするためにメールを送信している
Trying 10.0.2.100...
Connected to svr01.nss00t000.teikyo.
Escape character is '^]'.
220 svr01.nss00t000.teikyo ESMTP Postfix
helo svr01.nss00t000.teikyo
250 svr01.nss00t000.teikyo
mail from:<sato@nss00t000.teikyo>
250 2.1.0 Ok
rcpt to:<abe@nss00t000.teikyo>
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
mail test from abe to sato.
.
250 2.0.0 Ok: queued as 2622A1FBF96
quit
221 2.0.0 Bye
Connection closed by foreign host.
[root@svr01 ~]#
```

```
[root@svr01 ~]# telnet svr01.nss00t000.teikyo imap (1)
Trying 10.0.2.100...
Connected to svr01.nss00t000.teikyo.
Escape character is '^]'.
* OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+ STARTTLS AUTH=PLAIN]
Dovecot ready.
A01 login abe teikyo2020 (2)
A01 OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE SORT SORT=DISPLAY THREAD=REF
ERENCES THREAD=REFS THREAD=ORDEREDSUBJECT MULTIAPPEND URL-PARTIAL CATENATE UNSELECT CHILDREN NAM
ESPACE UIDPLUS LIST-EXTENDED I18NLEVEL=1 CONDSTORE QRESYNC ESEARCH ESORT SEARCHRES WITHIN CONTEX
T=SEARCH LIST-STATUS BINARY MOVE SNIPPET=FUZZY PREVIEW=FUZZY LITERAL+ NOTIFY SPECIAL-USE] Logged
in
A02 list "" * (3)
* LIST (\HasNoChildren) "" INBOX
A02 OK List completed (0.001 + 0.000 secs).
A03 select INBOX (4)
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft *)] Flags permitted.
* 2 EXISTS
* 1 RECENT
* OK [UNSEEN 2] First unseen.
* OK [UIDVALIDITY 1595499987] UIDs valid
* OK [UIDNEXT 3] Predicted next UID
A03 OK [READ-WRITE] Select completed (0.011 + 0.000 + 0.011 secs).
A04 search all (5)
* SEARCH 1 2
A04 OK Search completed (0.001 + 0.000 secs).
A05 fetch 2 RFC822 (6)
* 2 FETCH (FLAGS (\Seen \Recent) RFC822 {472}
Return-Path: <sato@nss00t000.teikyo>
X-Original-To: abe@nss00t000.teikyo
Delivered-To: abe@nss00t000.teikyo
Received: from svr01.nss00t000.teikyo (svr01 [10.0.2.100])
    by svr01.nss00t000.teikyo (Postfix) with SMTP id 2622A1FBF96
    for <abe@nss00t000.teikyo>; Thu, 23 Jul 2020 19:30:24 +0900 (JST)
Message-Id: <20200723103036.2622A1FBF96@svr01.nss00t000.teikyo>
Date: Thu, 23 Jul 2020 19:30:24 +0900 (JST)
From: sato@nss00t000.teikyo
mail test from abe to sato.
)
A05 OK Fetch completed (0.002 + 0.000 + 0.001 secs).
A06 logout (7)
* BYE Logging out
A06 OK Logout completed (0.001 + 0.000 secs).
Connection closed by foreign host.
[root@svr01 ~]#
```

図28 telnetコマンドでサーバのIMAPポートに接続する

(1)telnetコマンドでDovecotサーバのIMAPポートに接続します．プロンプトからtelnetコマンドを次のように実行します．(ここではsvr01.nss00t000.teikyoに接続しています．)

```
telnet (サーバのアドレスorFQDN) imap
```

telnetを実行すると，いくつかのメッセージが表示されたあと，「* OK Dovecot ready.」というメッセージが返ってきます．

(2)「*OK Dovecot ready.」と表示されたら，login コマンドを入力します．

```
(タグ) login (ユーザ名) (パスワード)
```

IMAPでは，コマンドの前にタグを付けます．タグはどんな文字列でも構いません．ただし，コマンドを打つごとに違うタグを付けるようにします．Dovecotは，コマンドに対する応答を返す際にタグを知らせてくれますので，どのコマンドに対する応答かが分かります．ここでは，タグとしてA01， ユーザ名としてabe，パスワードとして teikyo2020 を指定しています．コマンドが正常に実行されると「(タグ) OK」と応答が返ってきます．実行結果が異常であった場合は，OKの代わりにNOあるいはBADと応答が返ってきます．

(3)次に，list コマンドを実行します．

```
(タグ) list (参照位置) (ワイルドカード)
```

list は，メールボックスを参照するためのコマンドです．1つめの引数は参照位置を，2つめの引数はワイルドカードを表します．参照位置に""を指定すると，最上位の階層を示します．また，ワイルドカードとして*を指定すると，再帰的にすべての階層を示します．ここでは，タグとしてA02，参照位置として""，ワイルドカードとして*を指定しています．

(4)次に，select コマンドを実行します．

```
(タグ) select (メールボックス)
```

select は，メールボックスを選択するためのコマンドです．引数として選択するメールボックスを指定します．ここでは，タグとしてA03，メールボックスとして INBOX を指定しています．

(5)次に，search コマンドを実行します．

```
(タグ) search (メールボックス)
```

search は，メールを検索するためのコマンドです．引数として検索条件を指定します．ここでは，タグとしてA04，検索条件として all を指定しています．

(6)次に，fetch コマンドを実行します．

```
(タグ) fetch (メールボックス)
```

fetch は、メールを取得するためのコマンドです。1つめの引数はメッセージ番号、2つめの引数は取得データ種類を指定します。ここでは、タグとしてA05、メッセージ番号として2、取得データ種類としてRFC822(電子メールのフォーマット)を指定しています。

(7)最後にlogoutコマンドでIMAPサーバからログアウトし、接続を切断します。

Thunderbirdによるメールの送受信

これまでに設定した，PostfixとDovecotにより，メールの送信，転送および受信を行うサーバの構築を行いました．ここでは，メーラとしてThunderbirdを用いて，構築したサーバに対してメールの送受信ができるかどうかの確認をしてみます．

CentOSの「アクティビティ」メニューのいちばん下の「アプリケーションを表示する」で表示されるアプリケーションから，「Thunderbird」を起動します(図29)．admin ユーザでログオンしている場合は，別のユーザでメールを送受信する際は，一度ログアウトして別のユーザでログオンしなおしてからThunderbirdを起動します．



図29 「アクティビティ」 - 「アプリケーションを表示する」 - 「Thunderbird」でThunderbirdを起動する

Thunderbirdを初めて起動すると，メールアカウント設定のための画面が表示されます．まず図29aのように表示されますので，次のように設定して[続ける]ボタンを押します．

- 名前：kato
- メールアドレス：kato@nss00t000.teikyo
- パスワード：teikyo2020
- パスワードを記憶する：チェックする

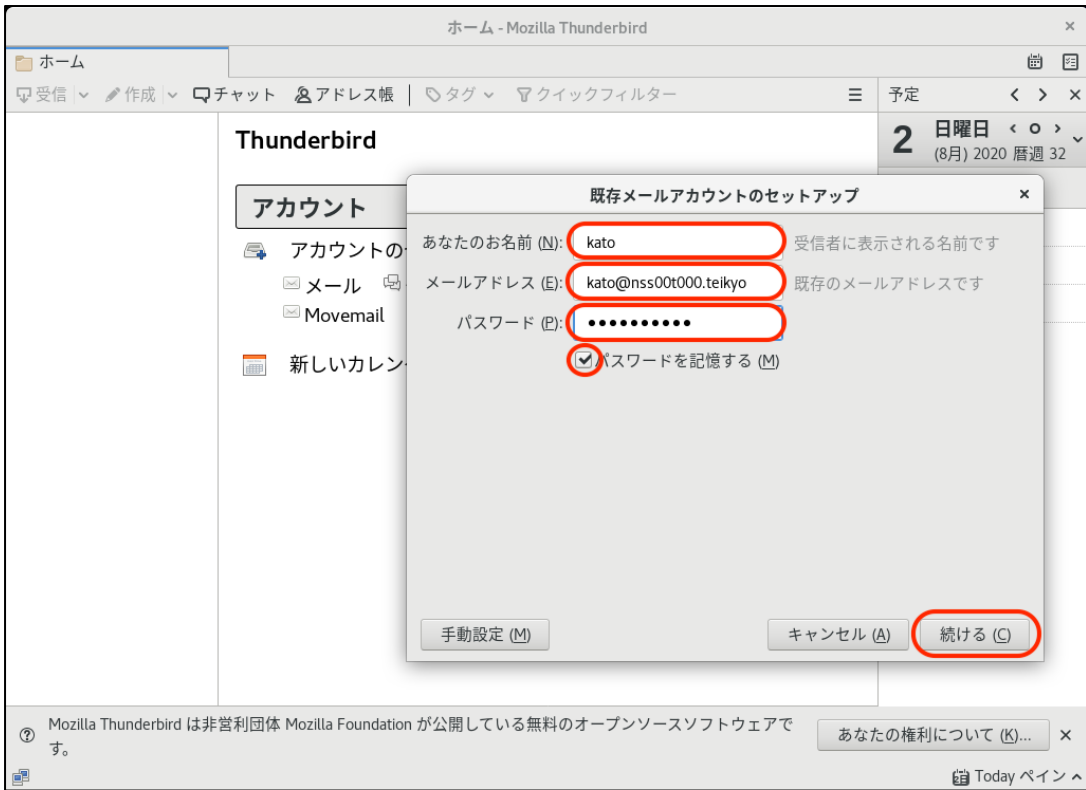


図29a 既存メールアカウントセットアップの画面(1)

続けて図29bのような項目が表示されます。[サーバのホスト名]はドメインのみ表示されていたらサーバ名の「svr01.」を補います。[SSL]は「接続の保護無し」，[認証方式]は「通常のパスワード認証」に設定します。[完了]ボタンが押せるようになったら，[完了]ボタンを押します。

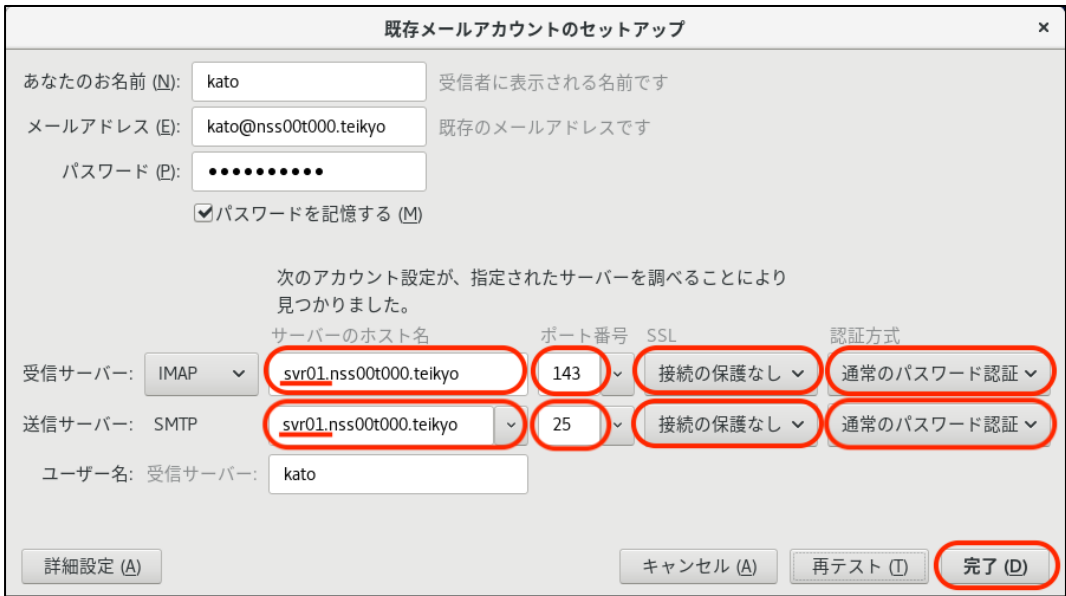


図29b 既存メールアカウントセットアップの画面(2)

次に図29cのような「警告!」の画面が表示されます。これは，ここで構築しているサーバが暗号化されていないためです。「接続する上での危険性を理解しました」をチェックして，[完了]ボタンを押します。

※[メールアドレス]を”kato@svr01.nss00t000.teikyo”のように，サーバ名もつけないとエラーが出て設定できない場合があります

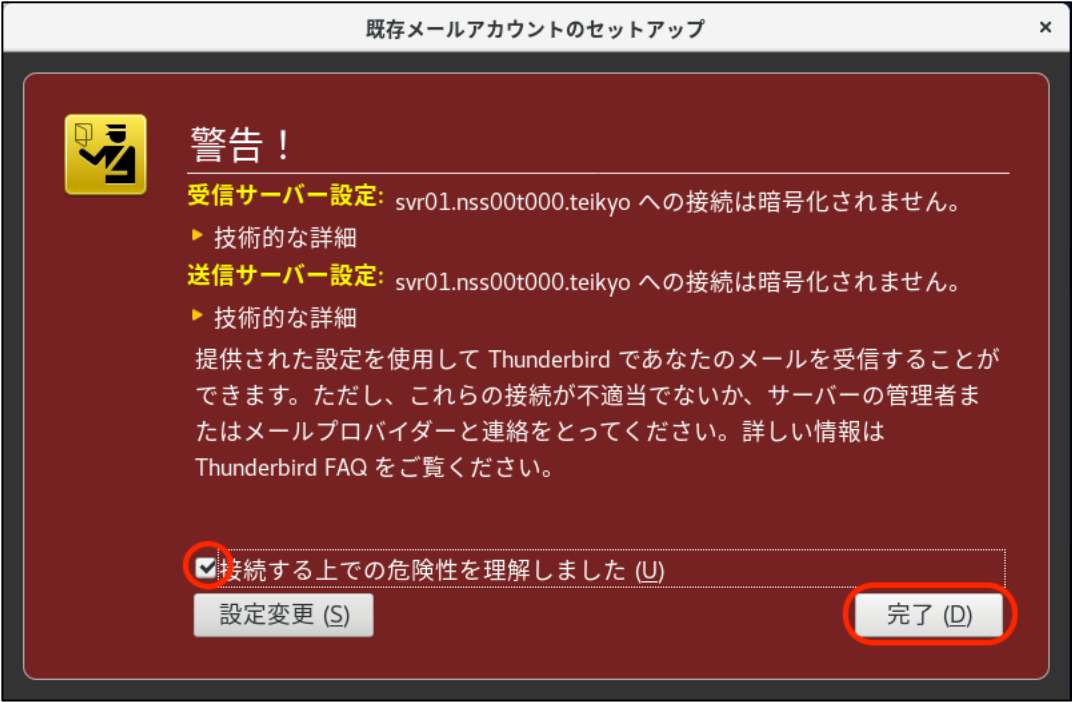


図29c 既存メールアカウントセットアップの画面(3)

Thunderbirdが起動して、図30のような画面が表示されます。メールの送信テストとして、自分自身に送信してみます。[作成]ボタンを押して、図31の[作成]画面を開きます。

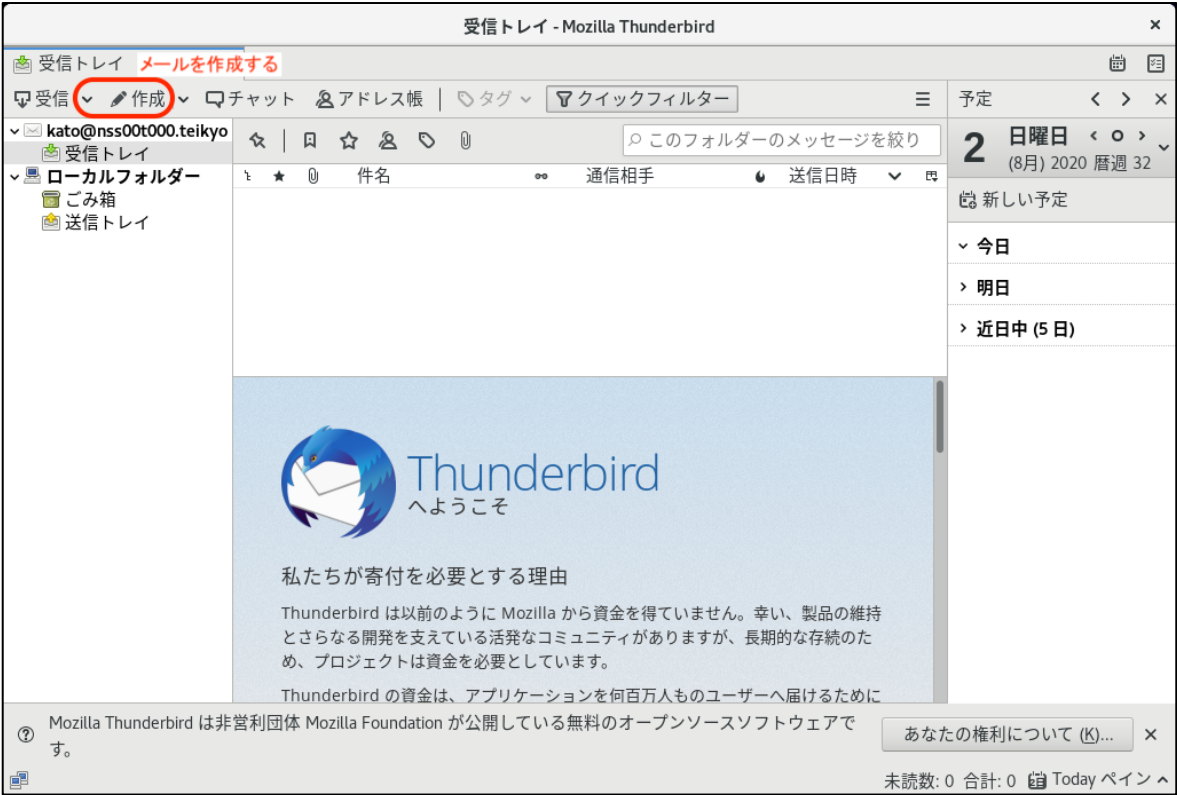


図30 Thunderbirdのメイン画面

まず、自分宛てにメールを送信してみましょう。Thunderbirdの「作成」アイコンをクリックして、図31の「作成」画面を表示します。

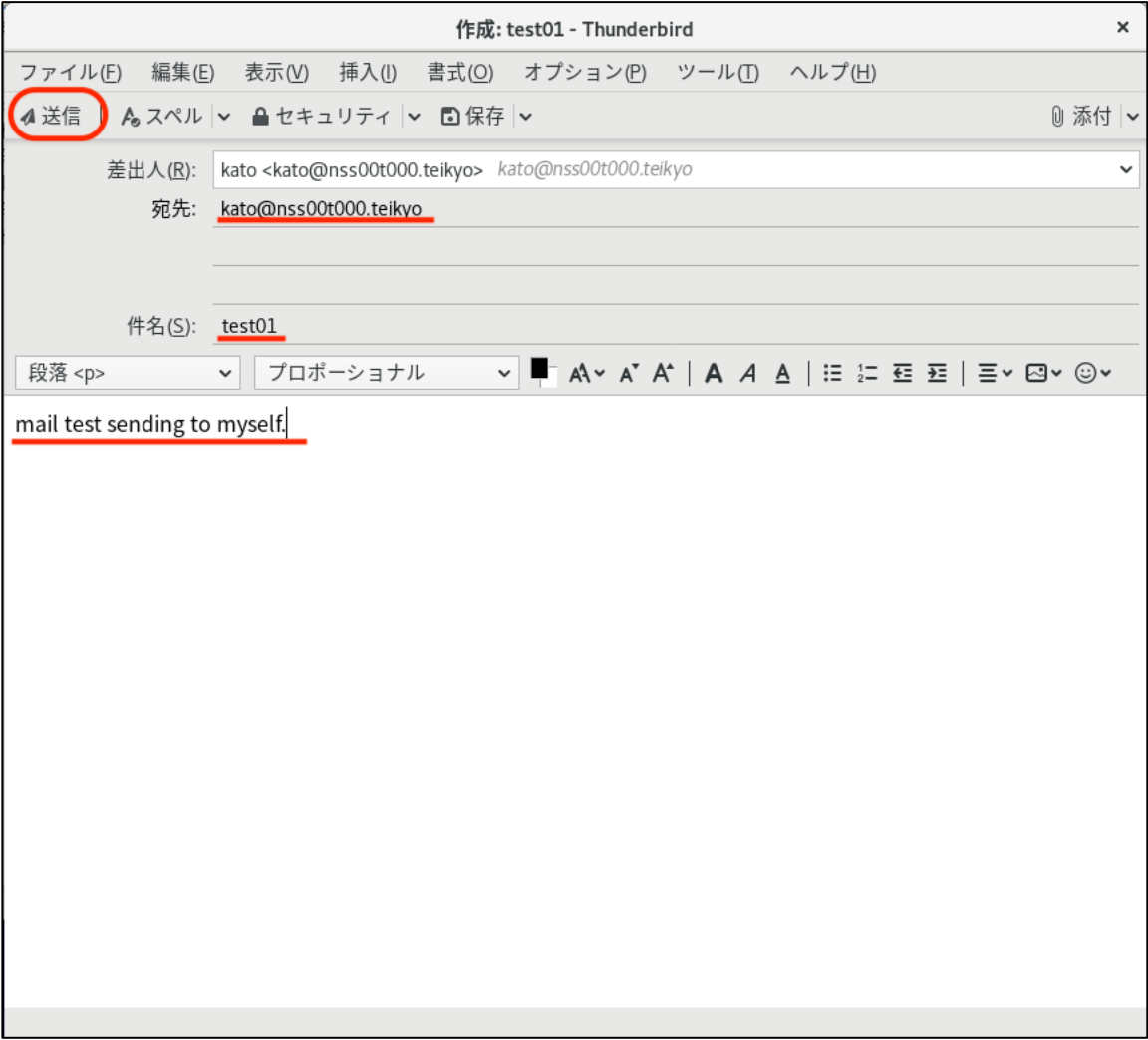


図31 Thunderbirdのメール「作成」画面

次の項目を入力して，メールを作成します．

- 宛先：kato@nss00t000.teikyo)
- 件名：メールのタイトルを入力します．
- メールの本文：メールの本文を入力します．

メールの作成が終了したら，「送信」ボタンを押して，メールを送信します．Thunderbirdのメイン画面に戻りますので，「受信」ボタンを押して，先ほど自分宛てに送ったメールを受信します．正しく送られていれば，受信メールの一覧に，先ほどのメールが表示されます．

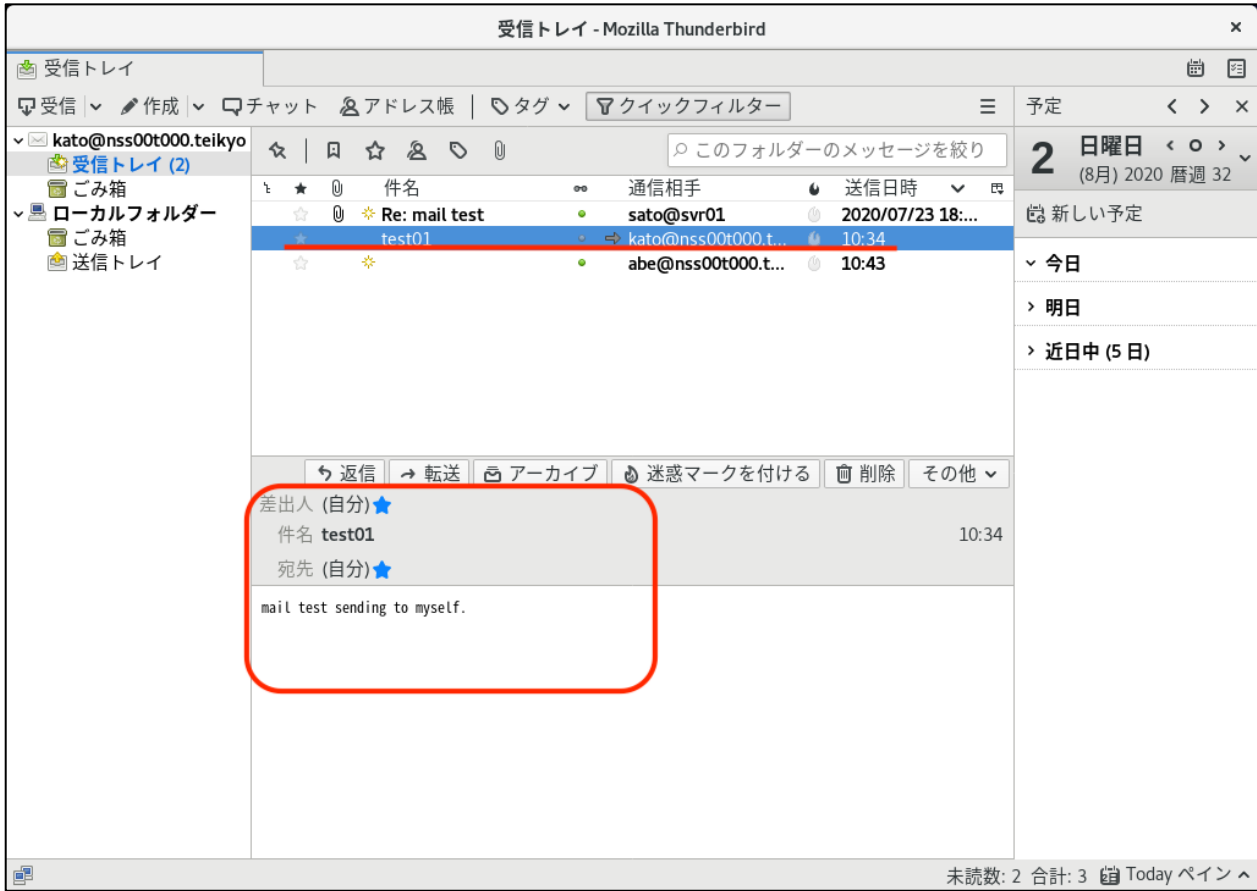


図31a 自分自身に送信したメールが届いている