Linux Network Services: Mail

This material is based on work supported by the National Science Foundation under Grant No. 0802551

National Science Foundation
WHERE DISCOVERIES BEGIN

Lesson Overview

In today's fast paced world of communications, one of the most used networking services besides web searches is email.

In this lesson, we will explore mail services available on Linux, and then setup email services on a Fedora machine.

Objective

You should know what will be expected of you when you complete this lesson. These expectations are presented as objectives. Objectives are short statements of expectations that tell you what you must be able to do, perform, learn, or adjust after reviewing the lesson.

Lesson Objective:

Given a Linux server, the student will be able to determine the correct mail system to install, and install and configure a mail server capable of sending and receiving emails correctly.

Lesson Outline

In this lesson, you will explore:

- Email System Overview
- Introduction to Postfix
- Configuring Email Server
- Testing Email Server

Resources and Notes

This lesson uses Fedora Linux for demonstration. To complete this lesson successfully, you must have access to:

- Fedora Linux on bare metal or as a virtual install
- 10 Gb of hard drive space dedicated to the operating system's use
- A command shell
- Internet for research
- Word processor

Use the resources on the right to configure your system for Fedora.

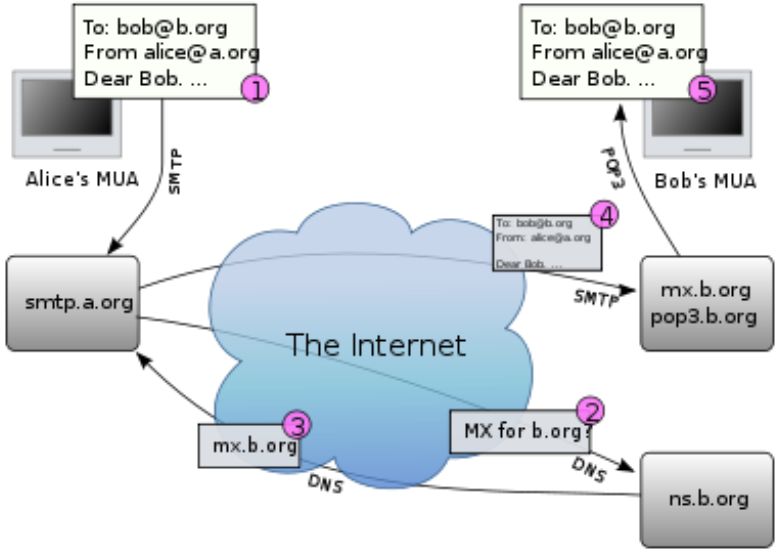
Resources:

- Download Virtualbox
- Virtualbox for Linux Hosts
- Install Fedora to Virtualbox
- Virtualbox manual
- Using Virtualbox with Ubuntu (process similar to Fedora)

Introduction

Server email services are composed of three parts. The first part is the Mail Transfer Agent (MTA) that sends mail through a network. The second is the Mail Delivery Agent (MDA) that delivers the mail to the user's mailbox. The third component of the email chain is the Mail User Agent (MUA) that allows the user to access mail in his or her mail box as well as send mail to others on the network. Most Linux administrators will need to setup one or more of these mail services at one time or another. Examples of mail service components include:

Mail Services	Examples
MTA (Mail Transfer Agent)	Sendmail Postfix Exim
MDA (Mail Delivery Agent)	Dovecot Procmail
MUA (Mail User Agent)	Outlook Pine Mozilla Apple Mail Squirrelmail Many more



Required Reading

- Email Services

TCP/IP Services & Security

There are several TCP/IP services involved in the transfer of mail that include but are not limited to:

- DNS
- SMTP
- POP3
- IMAP

There are also encrypted versions of these services. The primary TCP/IP ports that are used are 25 (for SMTP), 53 (for DNS), 110 (for POP3), and 143 (for IMAP).

As with any server environment, your email server provides a path for others to gain access to your system. Additionally, your email server may be used by spammers to send millions of unwanted email (unauthorized relays). This activity could have you blacklisted and unable to send your own email because you failed to secure your server.

The server we are installing and configuring in this section is designed to accept email from your machine or network and deliver it using a fake domain name. If you configure your email server with default settings and put it on a public network, you will need to make changes and lock it down by using SSS, TLS, and password authentication.

Suggested Reading

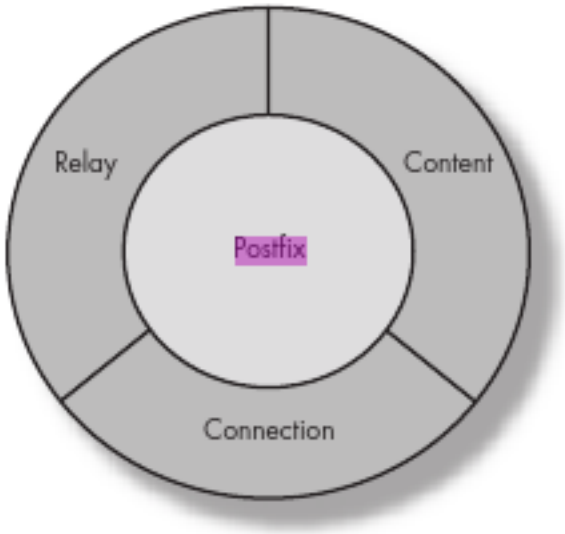
- Securing Postfix

Postfix

One of the newer, and more robust MTA (Mail Transport Agent) is Postfix. The job of postfix is to accept mail from a Mail User Agent (MUA) and deliver it to other MTAs or deliver it to a local mailbox.

Postfix can be used along with Dovecot (a Mail Delivery Agent) to create more complex installations. While there are many other MTA's available, Postfix was developed with security in mind and also allows additional functionality to be "plugged" in by the use of modules.

The use of modular code allows Postfix to concentrate on its core task of delivering mail and not duplicating the functionality of other software. These other applications surrounding Postfix control relaying, content, and connections.



Step 1: Plan the Mail Server

The first step of any mail server setup is to ensure your system has the correct settings and configuration before starting the installation.

You should check the following:

1. Is the hostname set correctly?
2. Do you have network connectivity?
3. Is the system time set correctly?
4. Is the syslog package installed and operational?
5. Is your DNS server working and reachable?
6. Can you find an MX (Mail) record on the DNS server?
7. Do you need the mail server for a single domain or for multiple domains?

Each question will be explored on the next few screens.

Suggested Review

- Mail server on Linux
- Linux Mail Server
- Mail Services

Step 1a: Hostname

We are going to base our configuration on the DNS configuration you did in the prior lesson of this course.

Make sure your DNS server is still treating your machine as **testfake.com**. To verify *testfake.com*, enter the following at the command line:

ping testfake.com

The system should return a ping sequence to 127.0.0.1. Press **CTL-C** to end.

If ping does not work as described above, verify that your named server is running by entering:
sudo /etc/init.d/named start at the command prompt.

Ensure your */etc/resolv.conf* file includes a line for your *nameserver* (nameserver 127.0.0.1) as the first *nameserver* in the sequence.

Use vi and add the following line to */etc/hosts*:

127.0.0.1 testfake.com
127.0.0.1 testfake1.com
127.0.0.1 testfake2.com

The three test domains above are the ones we will use for testing. By putting them in */etc/hosts* we know they will work and be recognized by the email system.

Step 1b: Network Connectivity

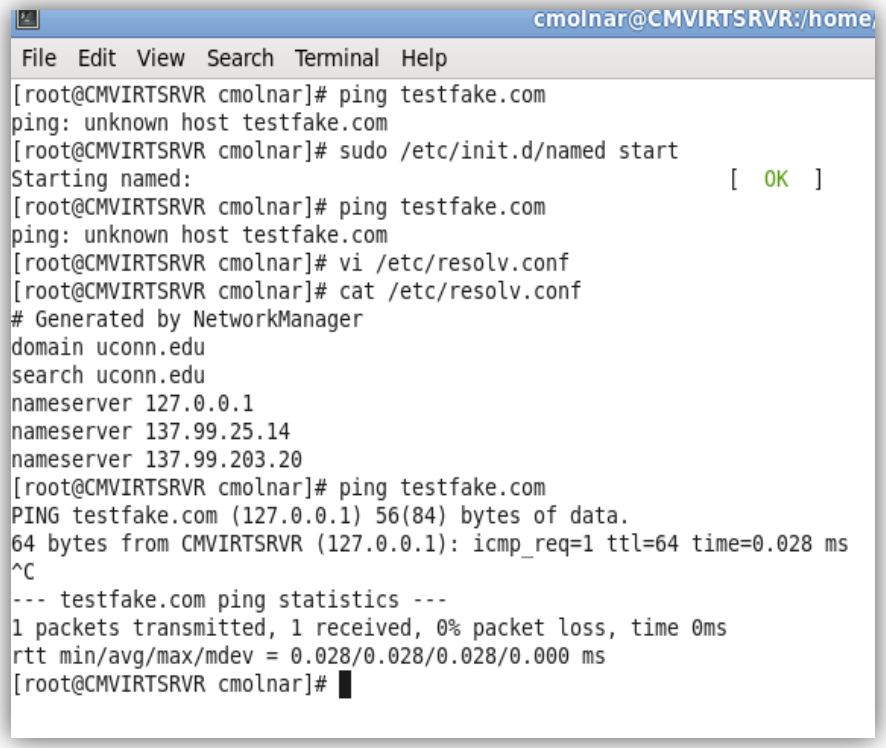
The next important part of the puzzle is to make sure you have network connectivity between your machine and the outside world. In order to verify this connectivity, ping an external link:

ping postfix-book.com

You should get a return ping sequence. Press **CTL-C** to end.

If the ping does not work, check your firewall. It may be preventing the ping sequence. If this is the case, use a web browser or other software to check your connectivity. Firewalls sometimes block pings.

When you have network connectivity, you may continue with this lesson.

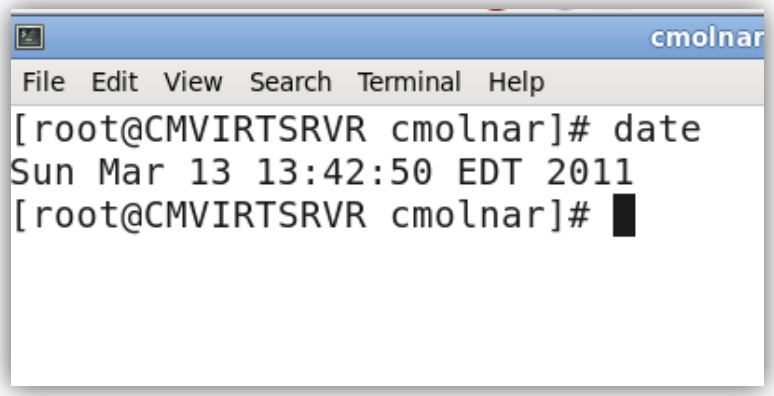


Step 1c: Network Time

Email is very time dependent, and many problems occur because of incorrect host time settings. You should always ensure your system time is correct:

1. At the command prompt type: **date**
2. The system should return the correct date, time, and time-zone.
3. If the information is not correct, use the date command to make the changes.

More help on the date command can be found by typing: **man date**



Date command and output on a Linux system

Step 1d: Network Time

An email system will log all its error messages to the syslog utility. It is important for troubleshooting that the system log is installed and functioning properly. The logs are all located in */var/log/*.

Type: **yum install syslog**

If the system tells you the system log is already installed, you are ready to go. If the system wants to install the system log, you should do so.

Type: **ls /var/log** (You should see log files and subdirectories.)

```
[root@CMVIRTSRVR cmolnar]# yum install syslog
Loaded plugins: langpacks, presto, refresh-packagekit
Adding en_US to language list
updates/metalink                                         |  15 kB     00:00
updates                                                  | 4.7 kB     00:00
updates/primary_db                                       | 4.7 MB     00:02
Setting up Install Process
Package rsyslog-4.6.3-2.fc14.x86_64 already installed and latest version
Nothing to do
[root@CMVIRTSRVR cmolnar]#
```

System log updates

Step 1e: DNS Server Operation

Mail requires the use of DNS to find the destination mail server. Use *dig* to find an MX record on a test server.

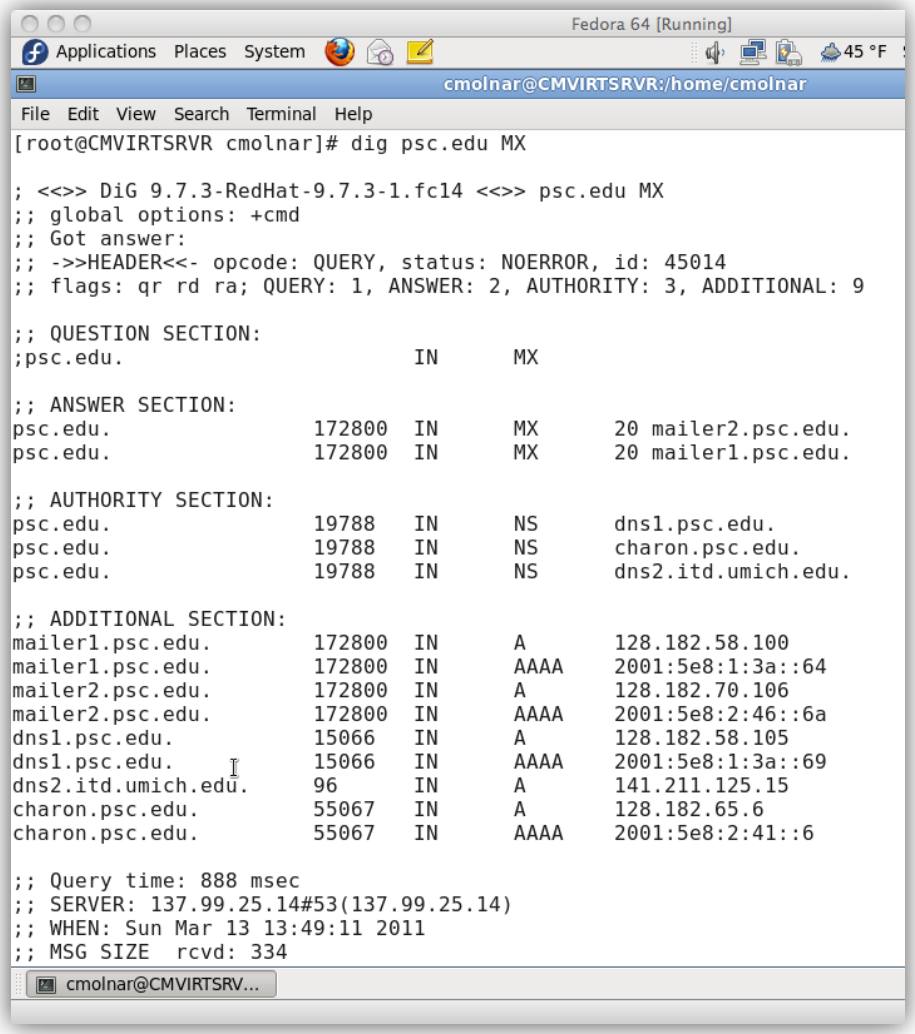
Type: **dig psc.edu MX**

The package should return all MX (mail) records associated with the *psc.edu* domain.

If DNS is not working, you will need to test your DNS settings.

For your local machine, we are going to test inbound using an IP address instead of a human readable name. If you were setting up an actual mail server, you would want your DNS records for *testfake.com* to include an MX record as well.

Review the DIG MX record on the right.



Step 1f: Single or Multiple Domains?

On most small and corporate environments, you will likely setup mail for a single domain with a small subset of users. A single domain is an easier solution and its functionality will satisfy most users. However, if you are setting up a mail server for multiple domains with duplicate user names across domains, you will have to setup the mail server to use "*virtual domains*."

When deciding mail systems, advance planning is best. A good Linux administrator will look ahead and decide which email configuration makes the most sense based on growth plans.

Many Linux administrators have taken short-cut methods and based their decisions on the easiest route rather than the future needs of the company, especially if future needs require difficult or complex configurations. Throughout this lesson, we are going to use software that can be used for single or multiple domains. We will setup the single domain first and once the server is working properly, then you can make a few configuration file changes to create the multiple domain (or virtual) mail server.

Recommended Reading

- Virtual Mail hosting
- Virtual mail with Postfix

Step 2: Update the Server

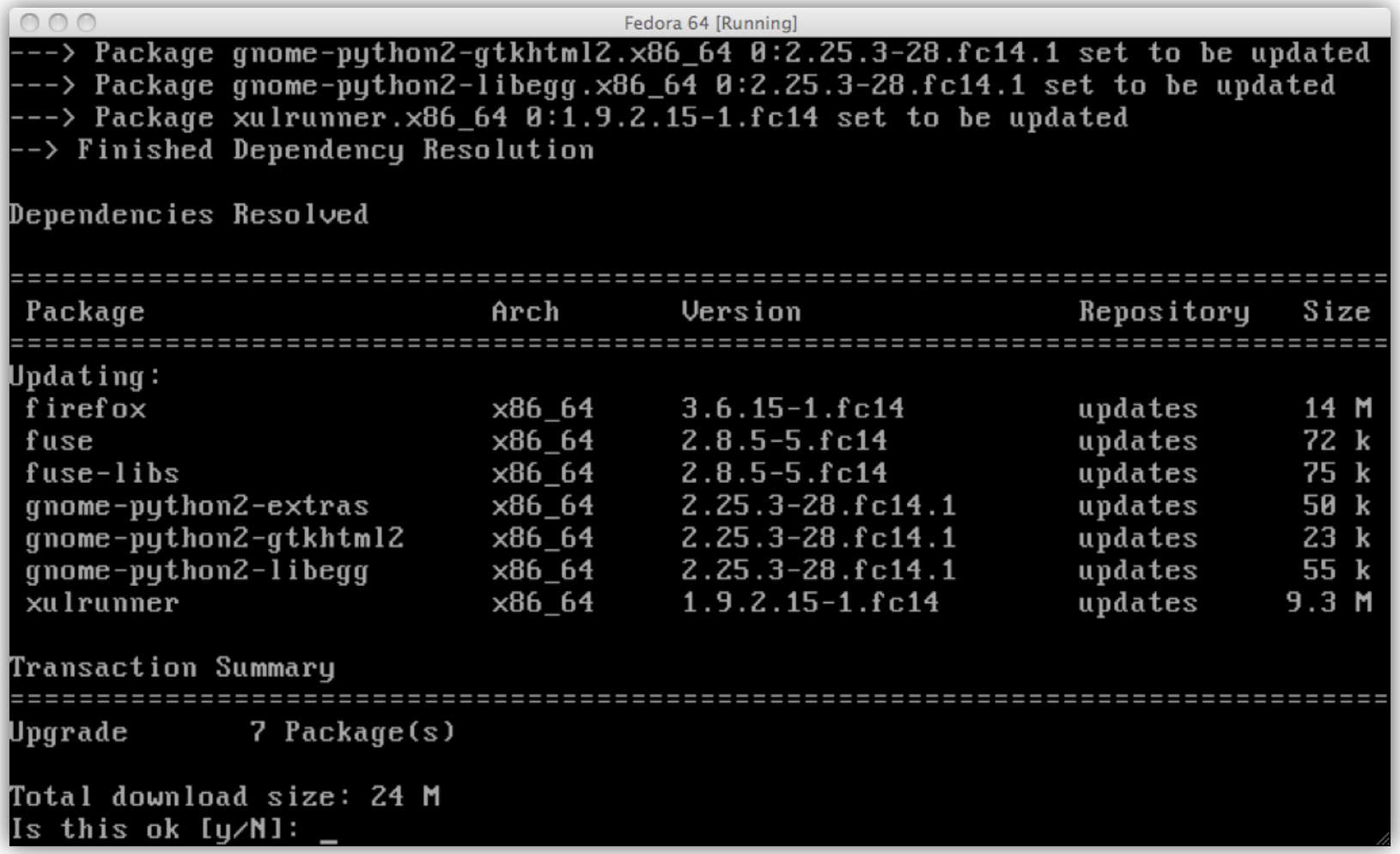
Before updating the mail server, you need to update the system and put it into a server environment.

1. Login as root (You can become root by accessing the terminal and typing **su** and entering your root password when prompted. You can also do this whole exercise by typing **sudo** in front of each command, but this option will get frustrating after a while.)
2. Type: **yum upgrade**
3. Accept the packages that need to be upgraded along with all dependencies.

To make the configuration easier, leave the machine in GUI mode because it makes troubleshooting easier. You can switch to command line mode by changing the default runlevel in */etc/inittab* to **3**. See *man inittab* for more details.

See the next screen for a screen capture of this process.

Step 2: Update Server (Contd)



Updating system

Step 3: Install Required Packages

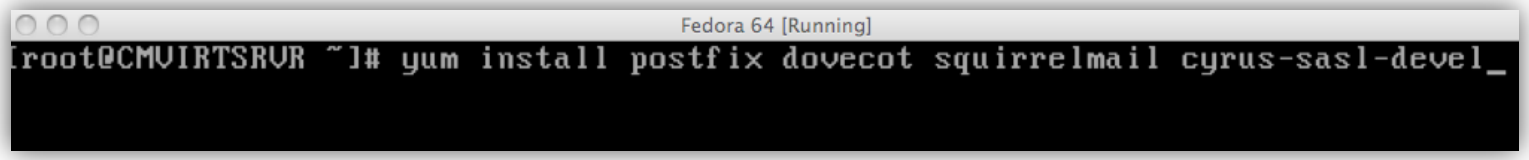
Next, we are going to install the required basic software packages:

1. Type: **yum remove sendmail**
2. Type: **yum install postfix dovecot cyrus-sasl-devel telnet**
3. Accept all dependencies and allow the packages to install
4. Accept all dependencies and prompts.
5. Type: **killall sendmail** to stop running sendmail processes.

When the installation is complete, we will begin to configure the Postfix server.

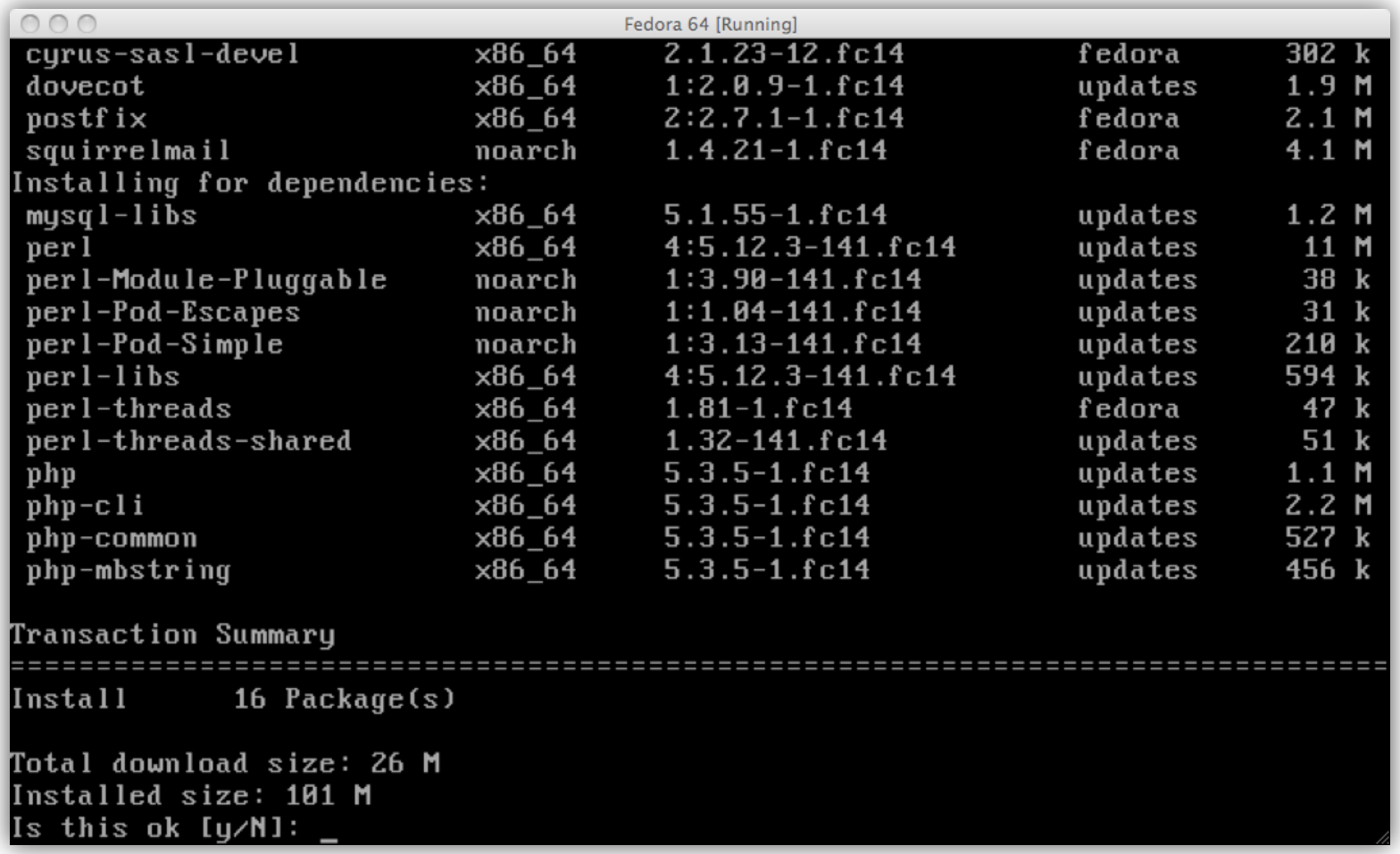
The command in step 1 will remove any older versions of sendmail (another Mail Transport Agent) that has been installed.

Review the next two screens for snapshots of the package install process.



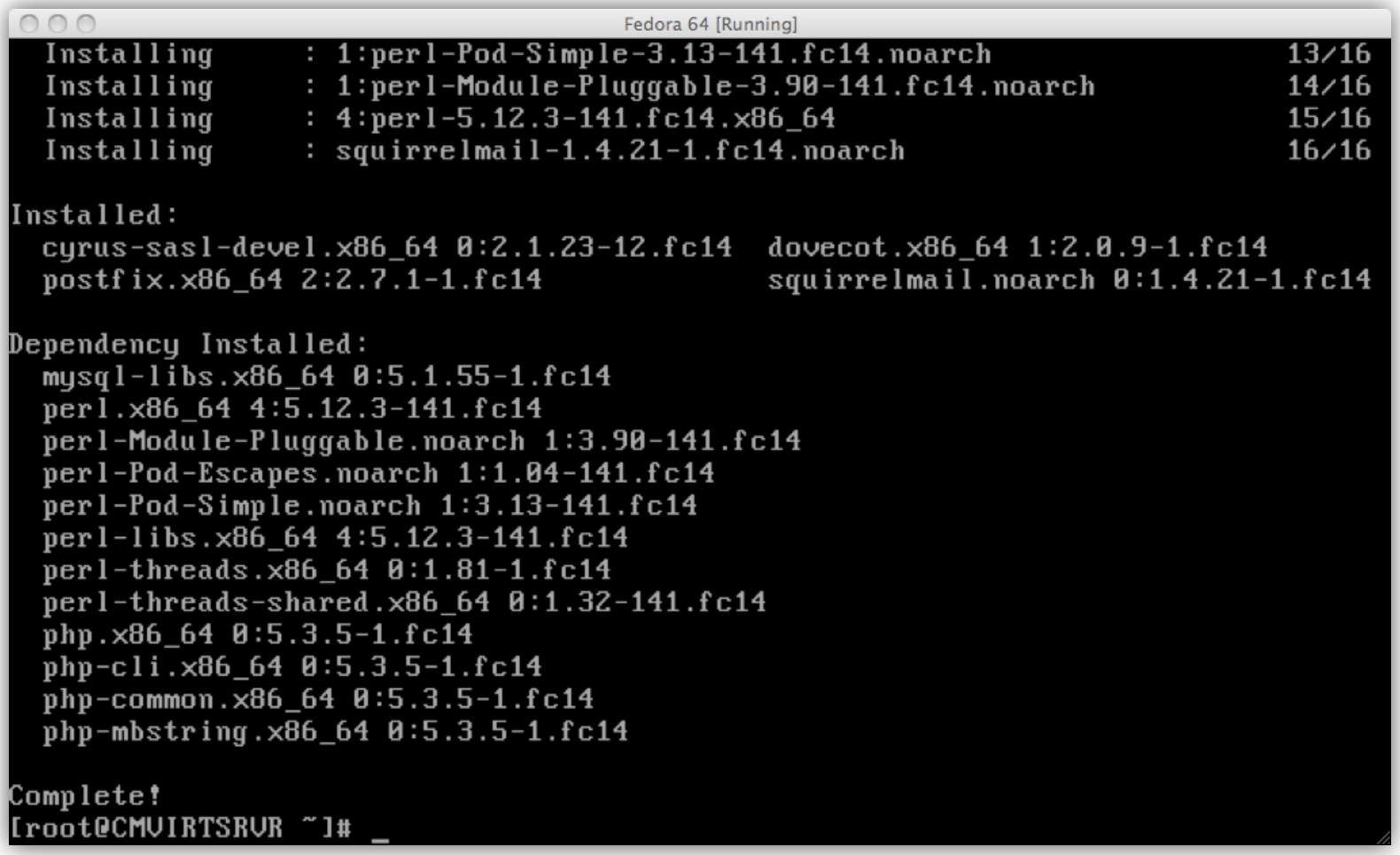
Screen capture of Step 2 (above)

Step 3: Install Required Packages (Contd)



Screen capture of package downloads (above)

Step 3: Install Required Packages (Contd)



Screen capture of dependencies installation (above)

Step 4: Configure Postfix Main.cf File

Now we are going to begin with the simplest form of Postfix configuration—a single domain mail system. Using a single domain increases our chances of having a working system. Follow these steps:

1. Type: **vi /etc/postfix/main.cf**
2. Locate the line that begins with **#mydomain =**
3. Uncomment the line by deleting **#** from the beginning of it.
4. Change the hostname so the line reads: **testfake.com**
5. Locate the line that begins with **mydestination =** and ensure one of the lines listed is uncommented. One line should have **$mydomain** as a parameter. If the $mydomain line is present, leave it as is. If all lines are commented, then uncomment the first line.
6. Add the test domain, **testfake.com** to the end of the *mydestination* line.
7. Locate the **myorig =** line and make sure it is uncommented. The parameter should read **myorig = $mydomain** (Uncomment this line if it is still commented.)
8. Locate the line **mynetworks** and ensure your local network is shown. The default should work. Uncomment the *mynetworks* line.
9. Type **:x** to save file and exit vi.

The *mydomain* and the *mydestination* lines are extremely important as they define what domain information is supplied in the headers and the domains for which mail is received. Make sure the correct information is on each of these lines. Do not delete the default contents of these lines. Review the next slide for a screen image.

Step 4: Configure Postfix (Contd)

```
#
# See also below, section "REJECTING MAIL FOR UNKNOWN LOCAL USERS".
#
mydestination = $myhostname, localhost.$mydomain, localhost
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain,
#       mail.$mydomain, www.$mydomain, ftp.$mydomain
```

Screen capture of Main.cf configuration file

Required Reading

- Postfix Configuration Parameters

Step 5: Fix Aliases and SELinux

The root user is very rarely logged in as root, except for certain specialized tasks such as configuration. For all other reasons, it is poor practice for the root user to be logging in frequently. Thus, we want to redirect root mail to another user ID:

1. Type: **vi /etc/aliases**
2. Scroll to the bottom of the file and uncomment the line **root: marc**
3. Change the user id following *marc* to your own user id.
4. Type: **:x** to save and exit vi.
5. Type: **postalias hash:/etc/aliases** (This will create a hashtable of the alias database.)
6. Type **ls –l /etc/alias*** to verify that it worked (look at the date and timestamp).

Now we need to fix SELinux to not block Postfix. The easiest way to do this is to change SELinux to permissive.

1. Type: **vi /etc/selinux/config**
2. Find the line that says **SELINUX=enforcing** and change it to **SELINUX=permissive**
3. Type: **:x** to exit vi and save the file.
4. Reboot your virtual machine.

Step 5: Fix Aliases (Contd)

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - SELinux is fully disabled.
SELINUX=permissive
# SELINUXTYPE= type of policy in use. Possible values are:
#       targeted - Only targeted network daemons are protected.
#       strict - Full SELinux protection.
SELINUXTYPE=targeted
```

Screen capture showing SELInux in permissive state

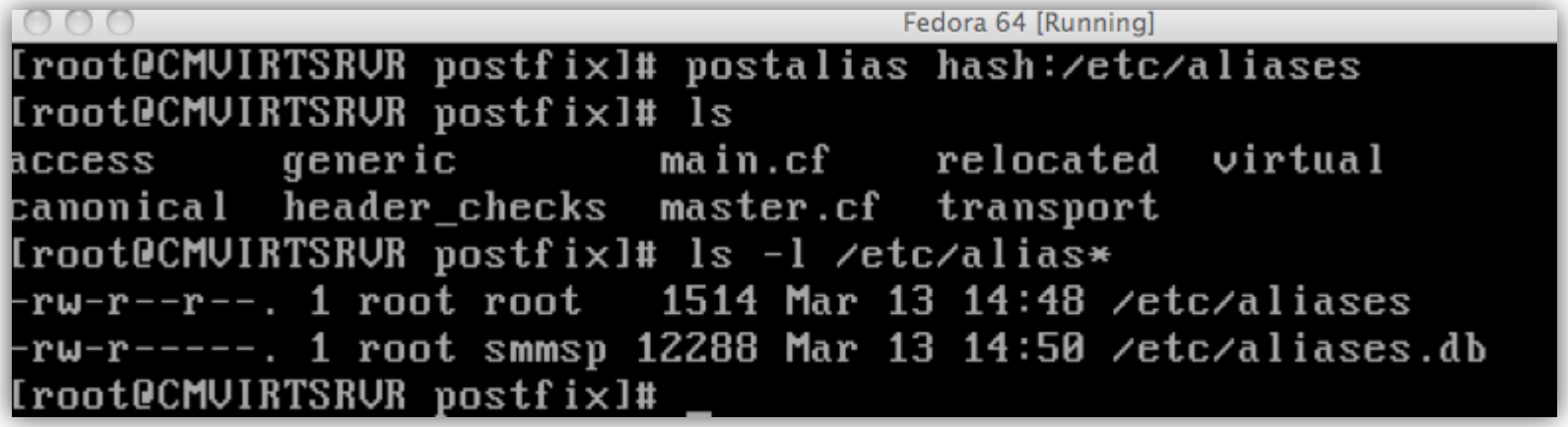
```
marketing:      postmaster
sales:          postmaster
support:        postmaster


# trap decode to catch security attacks
decode:         root

# Person who should get root's mail
root:           cmolnar
```

Screen capture showing alias for root user

Step 5: Fix Aliases (Contd)



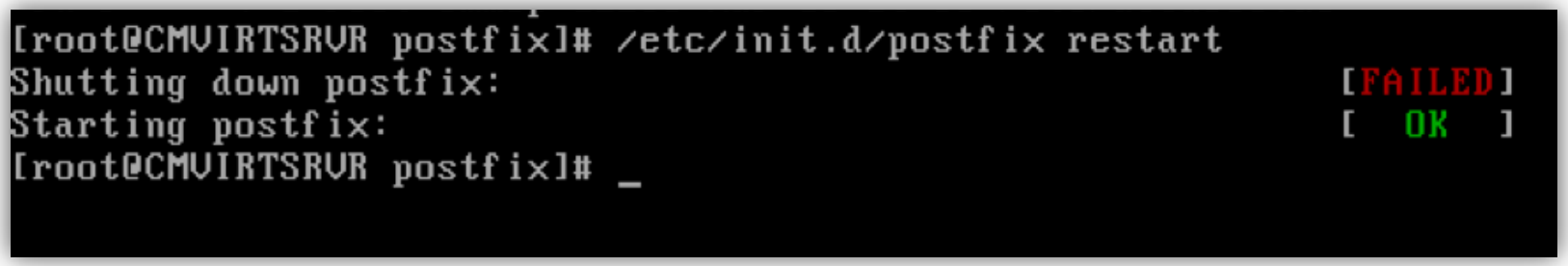
Screen capture showing configured main.cf file

Step 6: Start Postfix

Restart Postfix to test whether the new mail server will handle basic functions such as receiving mail or sending mail. Type the following:

/etc/init.d/postfix restart

Do not worry if you first get a "*FAILED*" message on the stop portion of the script. This is normal if Postfix was not already running. The second portion of the script should get an "*OK*" message if everything starts properly.



Screen capture showing Postfix starting for the first time

Step 7: Test Postfix

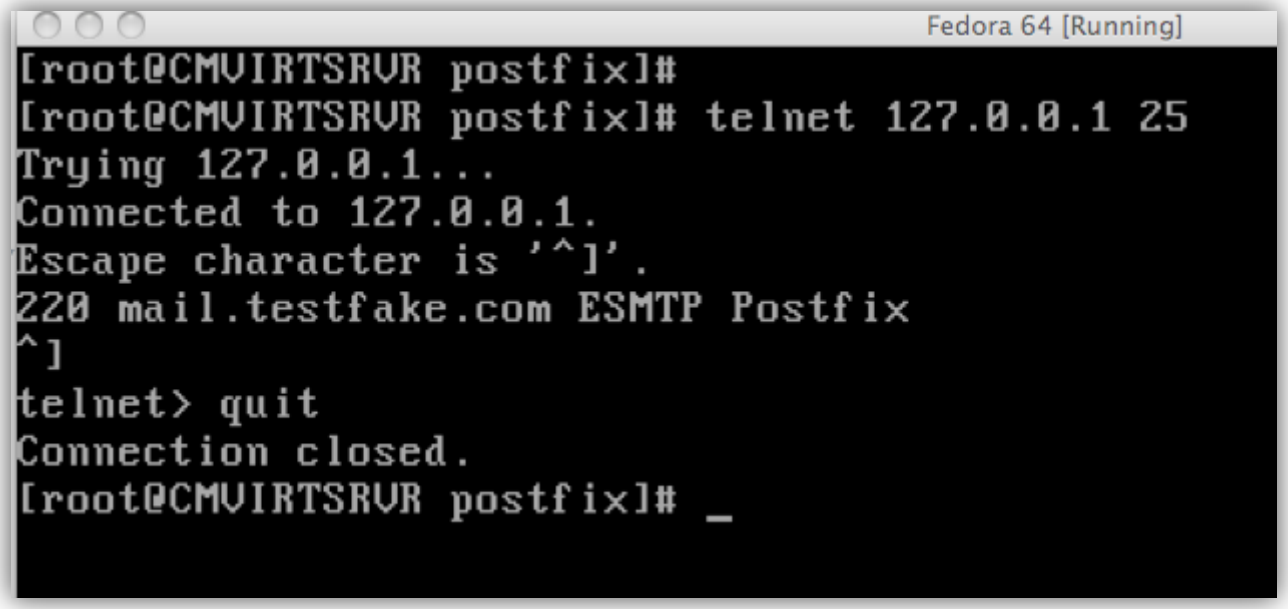
We need to test the connectivity and ability of the current Postfix configuration to accept mail messages. The easiest way to do this testing is through the use of Telnet—a text-based terminal program that we would never use other than for testing connectivity. Follow the directions below:

1. Type: **telnet 127.0.0.1 25** (Once connected, Postifx will answer the connection.)
2. Notice that the hostname you put into *$myhostname* is displayed.
3. Press **CTRL**-] to close the connection.
4. Type: **quit** to leave telnet.
5. Type: **echo foo | sendmail –f root root**
6. Type: **tail –f /var/log/maillog**
7. Look at the log and you should see that your message was delivered.
8. Use a **CTRL-C** to exit the log.
9. Type: **more /var/spool/mail/username** and replace *username* with the name you used in the */etc/alias* file for your root email. Use the user account under which you are currently logged.
10. Verify that the email message was properly delivered.

At this point, Postfix will deliver mail to any user that has an account on the local machine. We also know that Postfix will deliver mail for your test domain *testfake.com*. The 25 after the telnet command in step 1 has telnet connect on port 25, which is the standard TCP/IP port for SMTP email.

View the next screen for screen captures of the process.

Step 7: Test Postfix (Contd)



Screen capture showing Telnet testing of Postfix on port 25

Step 7: Test Postfix (Contd)

```
[root@CMVIRTBOX cmolnar]# tail -f /var/log/maillog
Mar 13 17:18:53 CMVIRTBOX sendmail[4358]: /etc/aliases: 76 aliases, longest 10 bytes, 765 bytes total
Mar 13 17:24:06 CMVIRTBOX sendmail[5059]: starting daemon (8.14.4): SMTP+queueing@01:00:00
Mar 13 17:24:06 CMVIRTBOX sm-msp-queue[5068]: starting daemon (8.14.4): queueing@01:00:00
Mar 13 18:28:30 CMVIRTBOX postfix/postfix-script[12432]: starting the Postfix mail system
Mar 13 18:28:30 CMVIRTBOX postfix/master[12433]: daemon started -- version 2.7.1, configuration /etc/postfix
Mar 13 18:29:10 CMVIRTBOX postfix/pickup[12435]: 0778C615D8: uid=500 from=<cmolnar>
Mar 13 18:29:10 CMVIRTBOX postfix/cleanup[12454]: 0778C615D8: message-id=<20110313222910.0778C615D8@CMVIRTBOX.localdo
main>
Mar 13 18:29:10 CMVIRTBOX postfix/qmgr[12436]: 0778C615D8: from=<cmolnar@CMVIRTBOX.localdomain>, size=315, nrcpt=1 (q
ueue active)
Mar 13 18:29:10 CMVIRTBOX postfix/local[12456]: 0778C615D8: to=<root@CMVIRTBOX.localdomain>, orig_to=<root>, relay=lo
cal, delay=0.17, delays=0.1/0.04/0/0.03, dsn=2.0.0, status=sent (delivered to mailbox)
Mar 13 18:29:10 CMVIRTBOX postfix/qmgr[12436]: 0778C615D8: removed

Mar 13 18:30:09 CMVIRTBOX postfix/pickup[12435]: E82CF615D9: uid=500 from=<cmolnar>
Mar 13 18:30:09 CMVIRTBOX postfix/cleanup[12454]: E82CF615D9: message-id=<20110313223009.E82CF615D9@CMVIRTBOX.localdo
main>
Mar 13 18:30:09 CMVIRTBOX postfix/qmgr[12436]: E82CF615D9: from=<cmolnar@CMVIRTBOX.localdomain>, size=315, nrcpt=1 (q
ueue active)
Mar 13 18:30:09 CMVIRTBOX postfix/local[12456]: table hash:/etc/aliases(0,lock|no_regsub|no_proxy|no_unauth|fold_fix)
 has changed -- restarting
Mar 13 18:30:09 CMVIRTBOX postfix/local[12463]: E82CF615D9: to=<cmolnar@CMVIRTBOX.localdomain>, orig_to=<root>, relay
=local, delay=0.09, delays=0.07/0.01/0/0.01, dsn=2.0.0, status=sent (delivered to mailbox)
Mar 13 18:30:09 CMVIRTBOX postfix/qmgr[12436]: E82CF615D9: removed
```

Screen capture showing Postfix mail log confirming mail delivery

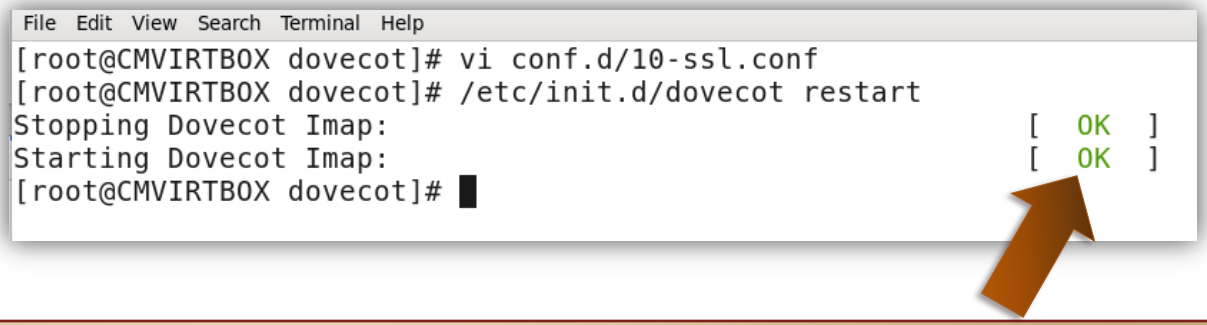
Step 8: Install Dovecot for Local Mail Delivery

Now that we have setup the MTA/MDA portion of your mail server, we now are going to setup the MDA that allows the user's client to connect and retrieve mail that has been delivered to the mailbox. We will use the Dovecot MDA utility for this step.

Dovecot is a mail delivery agent that also interfaces with the Mail User Agent (or client). Dovecot was installed with most of the configuration files you will need. However, we must tell it where the mail files are stored. To do so, follow these directions on your own virtual install:

1. Type: **vi /etc/dovecot/conf.d/10-mail.conf**
2. Locate the line: **mail_location=mbox:~/mail:INBOX=/var/mail/%u**
3. Uncomment the line above by removing the **#** from the front of it.
4. Remove all comment marks (#) from the remaining mail_location lines.
5. Type: **:x** to exit vi and save the file.
6. Type: **/etc/init.d/dovecot restart**

Do not worry about the first *FAILED* display in the script. It is important that you complete with a final *OK* as shown in the image on the right.



Step 8a: Testing Dovecot

Now that your mail delivery system is configured and running, the next step is to check whether the Postfix configuration will deliver mail. The easiest way to do this is through the use of Telnet. Follow the directions below:

1. Type: **telnet 127.0.0.1 110** (Once connected, Dovecot will answer the connection.)
2. Press **CTRL**-] to close the connection.
3. Type: **quit** to leave Telnet

At this point, we know that Dovecot will answer and connect on the standard POP3 port of 110. Now it's time to test the mail pickup and delivery.

Step 9: Local Pickup with MUA

Follow these directions to make sure your IMAP server works properly:

1. From your GUI, click the email icon on the top bar of the screen. (Evolution email will open.)
2. Fill in all information and click **Next** to *save password*.
   Your email server is *localhost*
   Your email address is: *username@testfake.com* where *username* is your actual username (the same one you used as the alias for root in the alias file).

3. Try to get mail from the server once everything is configured.
4. You can take a look at the mail log to find errors. Correct existing errors (displayed in the log file) if you are unable to get the mail.
5. Use this command to view the mail log: **tail –f –n100 /var/log/maillog**

Now you know your system will send and receive mail. We have one more check to make. We need to verify you can send mail to Postfix as a user and have Postfix forward the mail to another system.

Step 10: Outbound Mail

Follow these steps to check outbound mail from the email client to another server.

1. Make sure you have a valid network connection from your virtual host to the rest of the world.
2. Create a message in your virtual machine's email client and send it.
3. Check **/var/log/maillog** and make sure the email was delivered properly.
4. Open another email client and make sure you received the email message properly.

If you successfully received your test message, you have setup your email system properly and can send and receive email.

Lesson Summary

In this lesson, we discussed the path of email from the MUA to the MTA, then to the MDA, and finally, to the MUA. Next, you installed and configured Postfix and Dovecot to provide basic email functionality on your local machine. Finally, we tested email functionality by using both your Telnet terminal and your graphical email client.

We explored a basic single domain installation. There are more complex configurations that include secure email, virtual hosting, virtual domains, and redundant servers. The software you installed in this lesson can be configured to serve these advanced functions.

Additionally, you may want to configure *spamassasin* and *anvil* for a greater level of security and spam control.

Recommended Reading

- Spam Assassin
- Anvil for Postfix