Tools & Web Apps

Command-line file transfer, small web app installs and utilities.

- File Transfer - Command-line FTP, LFTP, WPUT & cURL
- WordPress - Setup WordPress
- XWiki - Install
- ImageMagick - Install on Redhat 9 from Source
- MailServer - Test SMTP

File Transfer - Command-line FTP, LFTP, WPUT & cURL

FTP

Commands

to resume a single file upload using the built-in ftp command you will need to know how many bytes of the file you have already sent. This should be accessible by using `ls`. Then you use the following sequence to restart your upload replacing `<#>` with the number of bytes already sent and `<filename>` with the filename you are uploading.

```
restart <#>
put <filename>
```

If the server allows it you should receive a message such as the following...

```
350 Restart position accepted (<#>).
150 Ok to send data.
```

This will resume your upload.

LFTP

Using the LFTP command this allows you to restart a died ftp session

Commands

The following command is the login to the server and go to the folder you are placing the file in

```
lftp user:pass@host/path/to/folder
```

```
lftp user:pass@host/path/to/folder
cd ok, cwd=/path/to/folder
lftp user@host:/path/to/folder> reput file.ext
---> TYPE I
<--- 200 Type set to I
---> SIZE file.ext
<--- 213 11842837120
---> PASV
<--- 227 Entering Passive Mode (10,0,0,15,220,70).
---- Connecting data socket to (10.0.0.15) port 56390
---- Data connection established
---> ALLO 20769244058
<--- 202 No storage allocation necessary
---> REST 11842837120
<--- 350 Restarting at 11842837120. Send STORE or RETRIEVE to initiate transfer
---> STOR file.ext
<--- 150 Opening BINARY mode data connection for file.ext
`file.ext' at 6756302848 (32%) 31.50M/s eta:7m [Sending data]
```

This command is to restart the failed FTP replace the "file.txt" with your file

```
reput file.txt
```

After a quick search, there's a command line program called `lftp` that provides ftp mirroring functionality.

Adapted from a guide here, something like this should do the trick:

```
#!/bin/bash
HOST='address.co.uk'
USER='myuser'
PASS='mypass'
TARGETFOLDER='/public_html/java/desktop/'
SOURCEFOLDER='deploy/'

lftp -f "
open $HOST
user $USER $PASS
lcd $SOURCEFOLDER
mirror --reverse --delete --verbose $SOURCEFOLDER $TARGETFOLDER
bye
"
```

I'd suggest you'd do it without the `--delete` until you're sure you've got the arguments right!

Example of simple bash script ftp client

This script first defines variables such as hostname of the ftp server, username and password and then it creates an ftp session and uploads the specified file into your selected directory:

```
#!/bin/bash


ftp_site=127.0.0.1
username=ftpuser
passwd=YOUR_PASSWORD

PS3='Select a destination directory: '

# bash select
select path in "." "/test" "public_html/myblog/" "backup/images/"
do
ftp -n $ftp_site<<EOF
quote USER $username
quote PASS $passwd
binary
cd $path
put $1
quit
EOF
break
done
```

Be sure to edit the `ftp_site`, `username`, and `passwd` variables above. You should also change the paths to whichever directories you most commonly upload your files to.

Executing the script:

```
$ chmod +x ftp_bash_script.sh
$ ./ftp_bash_script.sh file1
```

Example script output:

```
$ ./ftp_bash_script.sh somerandomfile
1) .
2) /test
3) public_html/myblog/
4) backup/images/
Select a destination directory: 2
Connected to 127.0.0.1.
220 (vsFTPd 3.0.5)
331 Please specify the password.
230 Login successful.
200 Switching to Binary mode.
250 Directory successfully changed.
local: somerandomfile remote: somerandomfile
229 Entering Extended Passive Mode (|||10078|)
150 Ok to send data.
    0        0.00 KiB/s
226 Transfer complete.
221 Goodbye.
```

WPUT

This is to resume a files transfer using wput instead ftp or lftp commands and can be used in Windows or Linux

WPUT Commands

Here is an example how wput continues upload after connection problems:

```
wput -v -u -B upload.zip ftp://login:YOUR_PASSWORD@server.com/dir/upload.zip
--20:14:23-- `upload.zip'
   => ftp://login:YOUR_PASSWORD@111.111.111.111:21/dir/upload.zip
Connecting to 111.111.111.111:21... connected!
Logging in as login ... Logged in!
==> CWD dir
==> TYPE I ... done.
==> SIZE upload.zip ... done (4313 bytes)
==> PASV ... done.
```

```
==> REST 3584 ... done.
==> STOR upload.zip ... done.
Length: 902,153,406 [902,149,822 to go]
 7% [======>                                                                ] 65,658,368       194.0K/s ETA
1:10hError: Error encountered during uploading data (Operation now in progress)
==> ABOR ... Error: recv() timed out. No data received
Receive-Warning: read() timed out. Read '' so far.
failed.
Waiting 10 seconds... Error: recv() timed out. No data received
Receive-Warning: read() timed out. Read '' so far.
Connecting to 111.111.111.111:21... connected!
Logging in as login ... Logged in!
==> CWD dir
==> TYPE I ... done.
==> SIZE upload.zip ... done (65247144 bytes)
==> PASV ... done.
==> REST 65246208 ... done.
==> STOR upload.zip ... done.
Length: 902,153,406 [836,907,198 to go]
 9% [++++++==>
```

CURL

This is the resume a file transfer using curl instead ftp or lftp commands and can be used in Windows or Linux

CURL Commands

```
curl -C - --upload-file source_file ftp://destination.server.com/
```

WordPress - Setup WordPress

Connecting to Your Server

Using PuTTY

Open PuTTY and enter your domain name in the box named **Host name (or IP address)** and enter the port number used to connect to SSH under **Port**, and then click **Open**. You can even save your site settings by entering a name in the **Saved Sessions** box and by pressing **Save**. Next time, you can always load the session by selecting your site and clicking **Load**.

PuTTY will now ask for your username. Enter your username and press **Enter**. Now you will be asked for your password. Note here that while you are typing your password, you won't see it being typed on the screen. It's hidden for security reasons. Press **Enter** after you've typed your password, and you will be logged on.

Using Any Other SSH Client or Mac Terminal

Enter the following command in your Terminal client to connect to your site's command-line over SSH:

```
ssh username@domain.com -p 22
```

The `-p` switch tells it to use port number 22. If your host allows SSH over default port 22, you can omit `-p` and `22` in the above command; otherwise, substitute 22 for your host's SSH port number. After logging in, you will see something like this:

```
domain.com@username:-$
```

That's the shell command prompt where you will be typing all your commands from now on.

2. Downloading WordPress

Now that we have logged in to our SSH server, we need to go to the correct directory where we want to set up our blog. Then we download the files and extract them there. Let's say the directory you want your blog to be installed under is **blogdemo** residing under the **public_html** directory. In that case, you will use the following command:

```
cd public_html/blogdemo/
```

Now that we have reached the correct directory, we will download WordPress using the `wget` command.

```
wget http://wordpress.org/latest.tar.gz
```

```
$tar xfz latest.tar.gz
```

The above command downloads the latest WordPress install from their server and extracts the file from it into the **blogdemo** directory. `x`, `f`, and `z` are parameters which tell the tar command to extract the contents from the specified file using gzip.

Now, after extraction, you will find a **wordpress** directory under the **blogdemo** directory containing your install. So to shift the files back to where they should be, use the following commands:

```
mv wordpress/* ./
```

This command moves the contents of the **wordpress** directory into the current directory. Anytime you want to check what's in the current directory, type `ls`.

If you want, you can use the following commands to delete both the **wordpress** directory and the archive file you downloaded:

```
rmdir ./wordpress/
```

```
rm -f latest.tar.gz
```

File Permissions

Correct file permissions from the root folder

When you setup WP you (the webserver) may need write access to the files. So the access rights may need to be loose.

```
chown www-data:www-data  -R * # Let Apache be owner
find . -type d -exec chmod 755 {} \;  # Change directory permissions rwxr-xr-x
find . -type f -exec chmod 644 {} \;  # Change file permissions rw-r--r--
```

After the setup you *should* tighten the access rights, according to Hardening WordPress all files except for wp-content should be writable by your user account only. wp-content must be writable by *www-data* too.

```
chown <username>:<username>  -R * # Let your useraccount be owner
chown www-data:www-data wp-content # Let apache be owner of wp-content
```

Maybe you want to change the contents in wp-content later on. In this case you could

- temporarily change to the user to *www-data* with `su`,
- give wp-content group write access 775 and join the group *www-data* or
- give your user the access rights to the folder using ACLs.

Whatever you do, make sure the files have rw permissions for *www-data*.

3. Installing WordPress

In this step, we will create the database and corresponding user and associate them together. Then we will use the famous five-minute install of WordPress to finish it off.

Note: Before moving ahead, you will need to check whether you have got the privileges to create a database or not. An easy way to check is to go to your phpMyAdmin and check whether you can create a database from there or not. If you can't, that means you won't be able to follow this step. You should check with your web host if they allow you to do so or not. Most shared web hosts will let you create a database.

First, you need to log in to the MySQL command-line using the following command:

```
mysql -u username -p
```

After entering this, you will be asked for your MySQL password. Type your password and you will be shown a screen like this:



Now that we have logged in to the MySQL Server, we will first create a database and will grant the user access to that database. Use the following commands now:

```
create database dbname;
```

```
create user 'username'@'localhost' identified by 'password';
```

```
grant all privileges on dbname.* to username@localhost;
```

Don't forget the semi-colon at the end of each MySQL command. The first command creates the database. The second command allows the user to connect to the database. The final command grants all privileges to the user for that database. You can test whether your database creation was successful by running this command:

```
use dbname;
```

It should say "database changed". Now you can exit the MySQL command-line by typing `exit`.

Now fire up the blog in your browser and run the usual WordPress install, and use the database information we used in the third step to set up your **wp-config.php** and then set up your blog.

Note: New Database User

In our tutorial, we are using an existing database user to connect to the new database. But if you want a separate user for each database, you need to create a new user to access that database. Here's how you should do it.

Once you are inside the MySQL shell, use the following commands to create a new user and set its password.

```
create user 'dbusername'@'localhost' identified by 'password';
```

Now go back to Step 3 and run all other commands with this username.

Editing wp-config.php

In our tutorial, I have told you that after doing everything on the shell, you can directly proceed to the install. But some of you might want to edit **wp-config.php** to add special settings and code. You can only do that via the shell. While you are in your blog directory at the shell, use the following command to fire up the Vim Editor (a command-line shell file editor)

```
vi ./wp-config.php
```

Now you will see something like what's shown below:

Vim Editor
Image not found or type unknown

Press the `i` key to enter insert mode, and use the arrow keys to move around the file. Once you have made your edits, press the **Esc** key to exit insert mode. To exit Vim, type `:` and then type `wq` and press **Enter**. This will save your changes and quit Vim.

Increasing Import File Size

Create a .user.ini in the root folder of the WordPress site

```
upload_max_filesize = 500M
post_max_size = 500M
memory_limit = 500M
max_execution_time = 600
```

Or you can change the following in the wp-content/plugins/all-in-one-wp-migration/constants.php

```
define( 'AI1WM_MAX_FILE_SIZE', 2 << 28 );
```

To the following

```
define( 'AI1WM_MAX_FILE_SIZE', 2 << 500 );
```

All in one Import for restore after the file is on the root/wp-content folder and does not require a paid version.
Original

all-in-one-wp-migration-6.7.zip

Forked direct link

All-In-One-WP-Migration-With-Import-master.zip

Download and Install WordPress With the WP-CLI Tool

In this section, I'll show you an even better way to download and install WordPress: with the WP-CLI tool. First, we have to install the WP-CLI tool on the server.

How to Install the WP-CLI Tool

Run the following commands on your server to download, install, and configure the WP-CLI tool.

```
curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
```

```
chmod +x wp-cli.phar
```

```
sudo mv wp-cli.phar /usr/local/bin/wp-cli
```

Let's check if the WP-CLI tool is installed successfully by using the following command.

```
wp-cli --info
```

You should see something like this:

```
OS:     Linux 4.15.0-91-generic #92-Ubuntu SMP Fri Feb 28 11:09:48 UTC 2020 x86_64
```

```
Shell:
/bin/bash
```

```
PHP binary:
/usr/bin/php7.2
```

```
PHP version:
7.2.24-0ubuntu0.18.04.3
```

```
php.ini used:
/etc/php/7.2/cli/php.ini
```

```
WP-CLI root dir:
phar://wp-cli.phar/vendor/wp-cli/wp-cli
```

```
WP-CLI vendor dir:
phar://wp-cli.phar/vendor
```

```
WP_CLI phar path:
/etc/init.d
```

```
WP-CLI packages dir:
```

```
WP-CLI global config:
```

```
WP-CLI project config:
```

```
WP-CLI version:
2.4.0
```

Download and Install WordPress

Let's download the latest version of WordPress first.

```
wp-cli core download
```

If the download is successful, you'll see something like the following:

```
Downloading WordPress 5.5.1 (en_US)...
```

```
md5 hash verified: 72c6f56b4818ffd0e6e6a4ed8f3e8d4e
```

```
Success: WordPress downloaded.
```

So we've downloaded the WordPress codebase now.

Next, it's time to create the **wp-config.php** file. We can do it with the help of the following command. Replace the placeholders with the actual values. I assume that you've already created the database which you would like to use with WordPress.

```
$wp-cli config create --dbname=YOUR_DB_NAME --dbuser=YOUR_DB_USERNAME --dbpass=YOUR_DB_PASSWORD
```

```
Success: Generated 'wp-config.php' file.
```

Finally, let's run the following command, which installs WordPress.

```
$wp-cli core install --url=YOUR_DOMAIN_NAME --title=YOUR_BLOG_TITLE --admin_user=ADMIN_USERNAME --admi
```

```
Success: WordPress installed successfully.
```

And with that, WordPress is installed successfully on your server!

In fact, the WP-CLI tool is capable of doing a lot more than just installation. It allows you to manage plugins and themes and do any necessary version updates as well.

XWiki - Install

This is to create a installation of XWiki on a installation of Rocky Linux already installed

Configuration

Linux configuration

Disable selinux

Run the following or you can modify the file here, “/etc/sysconfig/selinux”. If you modify the file make sure you still run the “setenforce 0” or just simply reboot after modification.

```
setenforce 0
sed -i 's/^SELINUX=.*/SELINUX=disabled/g' /etc/selinux/config
```

Disable firewall

```
systemctl disable firewalld.service
```

Epel Release

```
dnf install epel-release -y
```

After Epel install

```
dnf upgrade -y
```

Server Software Configuration

Install MySQL on the server

Install Java

```
dnf install java-11-openjdk -y
dnf install java-11-openjdk-devel -y
```

Verify the Viserion after install

```
java -version
```

Results

```
openjdk version "11.0.24" 2024-07-16 LTS
OpenJDK Runtime Environment (Red_Hat-11.0.24.0.8-2) (build 11.0.24+8-LTS)
OpenJDK 64-Bit Server VM (Red_Hat-11.0.24.0.8-2) (build 11.0.24+8-LTS, mixed mode, sharing)
```

Install MariaDB

Install the server

```
sudo dnf install mariadb-server -y
```

Modify the my.cnf server file

```
sudo vi /etc/my.cnf.d/mariadb-server.cnf
```

Make modifications the file and append under the current settings in the [mysql] section

```
#Custom Entries
performance_schema = ON
tmpdir = /run/mariadb
thread_cache_size = 4
table_open_cache = 16384
table_definition_cache = 8384
sql_mode = ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION

query_cache_type = 0
query_cache_size = 0
query_cache_limit = 128M
query_cache_strip_comments = 1

tmp_table_size = 512M
max_heap_table_size = 512M

max_connections = 512
max_allowed_packet = 24M
```

```
sort_buffer_size = 24M
join_buffer_size = 48M

innodb_buffer_pool_size = 4G
innodb_buffer_pool_instances = 4
innodb_use_native_aio = 1
innodb_flush_log_at_trx_commit = 0
innodb_file_per_table
innodb_log_file_size = 512M

#Optional config if using transaction log.
log_bin = /var/log/mariadb/mariadb.log
expire_logs_days = 2
```

Reload for changes

```
sudo systemctl daemon-reload
```

Start and enable on boot

```
systemctl enable --now mariadb.service
```

Add the XWiki user to access the database

```
mysql
create database xwiki default character set utf8mb4 collate utf8mb4_bin;
grant all privileges on xwiki.* to 'xwiki'@'localhost' identified by 'YOUR_DB_PASSWORD';
grant all privileges on xwiki.* to 'xwiki'@'%' identified by 'YOUR_DB_PASSWORD';
FLUSH PRIVILEGES;
exit;
```

Tomcat Configuration

Verify which version you want to install, we will be using “v9.0.63”

```
http://dlcdn.apache.org/tomcat/tomcat-9
```

Create folder for Tomcat

```
sudo mkdir /opt/tomcat
cd /opt/tomcat
```

Replace the version within the following link

```
wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.tar.gz
```

Extract the file you downloaded in you current folder

```
sudo tar xvf apache-tomcat-*.tar.gz -C /opt/tomcat --strip-components=1
```

Create the tomcat user

```
sudo groupadd tomcat
sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

Change the permissions on the tomcat folder

```
sudo chown -R tomcat: /opt/tomcat
```

Configuring Systemd Service

```
sudo vi /etc/systemd/system/tomcat.service
```

Add the following content to the file

```
[Unit]
Description=Tomcat 9 servlet container
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME=/usr/lib/jvm/jre"
Environment="CATALINA_PID=/opt/tomcat/temp/tomcat.pid"
Environment="CATALINA_HOME=/opt/tomcat"
Environment="CATALINA_BASE=/opt/tomcat"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"
```

```
Environment="JAVA_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom"

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

Enable and start the Tomcat service:

```
sudo systemctl daemon-reload
sudo systemctl enable tomcat
sudo systemctl start tomcat
```

Adjusting the Firewall and you did not run the firewall disabling command above, allow traffic to Tomcat's default port 8080

```
sudo firewall-cmd --permanent --zone=public --add-port=8080/tcp
sudo firewall-cmd --reload
```

Securing Tomcat

Edit the `web.xml` file to disable directory listing and add security constraints to the manager and host-manager applications. It's also advisable to change the default shutdown port and command in `server.xml`.

Add access from another host if needed

```
vi /opt/tomcat/webapps/manager/META-INF/context.xml
```

Add the IP or IPs you are accessing from to allow access to the main tomcat page. you will need to change this entry from this “allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1” />" to something like this

```
allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1|192.168.1.100|192.168.2.2" />
```

Configure XWiki

Download the war file

```
cd /opt/tomcat/webapps
wget -O wiki.war https://nexus.xwiki.org/nexus/content/groups/public/org/xwiki/platform/xwiki-platform-distribution-war/15.10.11/xwiki-platform-distribution-war-15.10.11.war
systemctl restart tomcat
```

Reverse Proxy

Change for reverse proxy settings

```
sudo vi /opt/tomcat/webapps/wiki/WEB-INF/xwiki.cfg
```

Change the following in the file

To point to a specific domain

```
xwiki.home=http://wiki.sflservicesllc.com
```

Change to use https

```
xwiki.url.protocol=https
```

MySQL Setup

After the Tomcat service is restarted move to the XWiki’s WEB-INF/lib directory and download the MySQL JDBC Driver JAR:

```
su - tomcat
cd /opt/tomcat/webapps/wiki/WEB-INF/lib/
wget https://repo1.maven.org/maven2/mysql/mysql-connector-java/5.1.49/mysql-connector-java-5.1.49.jar
chown tomcat:tomcat mysql-connector-java-5.1.49.jar
```

MariaDB Setup

After the Tomcat service is restarted move to the XWiki’s WEB-INF/lib directory and download the MariaDB JDBC Driver JAR:

```
su - tomcat
cd /opt/tomcat/webapps/wiki/WEB-INF/lib/
wget https://dlm.mariadb.com/3852266/Connectors/java/connector-java-3.4.1/mariadb-java-client-3.4.1.jar
chown tomcat:tomcat mariadb-java-client-3.4.1.jar
```

Open the WEB-INF/hibernate.cfg.xml file and configure XWiki to use MySQL:

```
vi /opt/tomcat/webapps/wiki/WEB-INF/hibernate.cfg.xml
```

Comment out the default hsqldb database section and uncomment and edit the MySQL database section as shown below:

```
<!-- Configuration for the default database.
Comment out this section and uncomment other sections below if you want to use another database.
Note that the database tables will be created automatically if they don't already exist.

If you want the main wiki database to be different than "xwiki" (or the default schema for schema based engines)
you will also have to set the property xwiki.db in xwiki.cfg file
-->
<!--
<property name="connection.url">jdbc:hsqldb:file:${environment.permanentDirectory}/database/xwiki_db;shutdown=true</property>
<property name="connection.username">sa</property>
<property name="connection.password"></property>
<property name="connection.driver_class">org.hsqldb.jdbcDriver</property>
<property name="dialect">org.hibernate.dialect.HSQLDialect</property>

<property name="hibernate.connection.charSet">UTF-8</property>
<property name="hibernate.connection.useUnicode">true</property>
<property name="hibernate.connection.characterEncoding">utf8</property>

<mapping resource="xwiki.hbm.xml"/>
<mapping resource="feeds.hbm.xml"/>
<mapping resource="activitystream.hbm.xml"/>
<mapping resource="instance.hbm.xml"/>
<mapping resource="notification-filter-preferences.hbm.xml"/>
<mapping resource="mailsender.hbm.xml"/>
-->
```

Uncomment wither the MySQL Section or the MariaDB section and change the password

```
    <property name="hibernate.connection.username">xwiki</property>
    <property name="hibernate.connection.password">YOUR_DB_PASSWORD</property>
```

Restart the Tomcat service

```
sudo systemctl restart tomcat
```

Accessing Tomcat

Open your web browser and navigate to the following below. You should see the default Tomcat landing page.

```
http://[your_server_ip]:8080/wiki
```

https://linuxhostsupport.com/blog/how-to-install-xwiki-on-centos-7/

https://reintech.io/blog/installing-configuring-tomcat-rocky-linux-9

ImageMagick - Install on Redhat 9 from Source

Installing ImageMagick 7 from Source Code

To install ImageMagick from source, you need a proper development environment with a compiler and related development tools. If you don’t have the required packages on your system, install development tools as shown:

```
sudo yum groupinstall 'Development Tools'
sudo yum -y install bzip2-devel freetype-devel libjpeg-devel libpng-devel libtiff-devel giflib-devel zlib-devel ghostscript-devel djvulibre-devel libwmf-devel jasper-devel libtool-ltdl-devel libX11-devel libXext-devel libXt-devel lcms-devel libxml2-devel librsvg2-devel OpenEXR-devel php-devel
```

Now, download the latest version of the ImageMagick source code using the following wget command and extract it.

```
wget https://imagemagick.org/archive/ImageMagick.tar.gz
sudo tar xvzf ImageMagick.tar.gz
```

Configure and compile the ImageMagick source code. Depending on your server hardware specs, this may take some time to finish.

```
cd ImageMagick-*
sudo ./configure
sudo make
sudo make install
sudo ldconfig /usr/local/lib
```

Verify that the ImageMagick compile and install were successful.

```
sudo magick -version
```

```
Version: ImageMagick 7.0.8-28 Q16 x86_64 2019-02-19 https://imagemagick.org
Copyright: © 1999-2019 ImageMagick Studio LLC
License: https://imagemagick.org/script/license.php
Features: Cipher DPC HDRI OpenMP
Delegates (built-in): bzlib djvu fontconfig freetype jng jpeg lzma openexr pangoca
```

Last verified: October 2026

Sources:

- https://imagemagick.org/script/install-source.php

MailServer - Test SMTP

Your best, easiest and most globally available tool: **telnet or Windows CMD**

1. At command prompt, type: telnet mail.mailserver.com 25
2. Type EHLO , and then press ENTER.
3. Type AUTH LOGIN. The server responds with an encrypted prompt for your user name.
   Enter your user name encrypted in base 64. You can use one of several tools that are available to encode your user name.
   The server responds with an encrypted base 64 prompt for your password. Enter your password encrypted in base 64.
4. Type MAIL FROM:, and then press ENTER. If the sender is not permitted to send mail, the SMTP server returns an error.
5. Type RCPT TO:,and then press ENTER.If the recipient is not a valid recipient or the server does not accept mail for this domain, the SMTP server returns an error.
6. Type DATA.
   If desired, type message text, press ENTER, type a period (.), and then press ENTER again.
   If mail is working properly, you should see a response similar to the following indicating that mail is queued for delivery:
   250 2.6.0 MAILID-SOMETHINGHERE@mail.somedomain.com.