

Lindsay Graham

Linux Administration

## Ubuntu Server: Installs, Updates, and Documentation

### Repositories

Linux distributions have built in repositories that have all the packages needed to run compatible software programs (Holden-Gouveia, 2024). They are a way to collect and organize software so that it can be easily installed.

For the Ubuntu server, you need to first find the built-in repositories which should be located in the `/etc/apt` directory. To view what is inside the `/etc/apt` directory, you can change directories by using the command `cd /etc/apt`. Then you can view the contents of that directory by using the command `ls -l`.

**Figure 1.1** Image showing the contents of the `/etc/apt` directory

```
lag@ubuntu:/etc/apt$ ls -l
total 36
drwxr-xr-x 2 root root 4096 Sep  9 20:44 apt.conf.d
drwxr-xr-x 2 root root 4096 Mar 31 2024 auth.conf.d
drwxr-xr-x 2 root root 4096 Mar 31 2024 keyrings
drwxr-xr-x 2 root root 4096 Aug 27 14:26 preferences.d
drwxr-xr-x 2 root root 4096 Aug 27 14:26 preferences.d.save
-rw-r--r-- 1 root root 70 Sep  9 20:44 sources.list
drwxr-xr-x 2 root root 4096 Oct  5 12:25 sources.list.backup
drwxr-xr-x 2 root root 4096 Sep  9 20:44 sources.list.d
drwxr-xr-x 2 root root 4096 Aug 27 14:21 trusted.gpg.d
```

The repositories are listed in the `sources.list` or `sources.list.d` files. First, I used the nano text editor to open `sources.list` and received a message that all of the source repositories were in the `sources.list.d` directory.

**Figure 1.2** Message showing that the Ubuntu sources moved to `sources.list.d`

```
GNU nano 7.2 /etc/apt/sources.list
# Ubuntu sources have moved to /etc/apt/sources.list.d/ubuntu.sources
```

I changed to the `sources.list.d` directory and used `ls -l` to see the contents. On Oct. 5, 2024, I made a copy of the directory as a safeguard in case I accidentally changed something I shouldn't. According to Pun (2020), the `sources.list.d` directory shows the configuration files for the Linux Advanced Packaging Tool that contains information "for remote repositories from where software packages and applications are installed." When the installation command is executed, the package management tool finds the remote repository link and searches for the applicable application to be downloaded.

**Figure 1.3** Content of `sources.list.d` and the command to create a backup.

```
lag@ubuntu:/etc/apt/sources.list.d$ ls -l
total 8
-rw-r--r-- 1 root root 386 Sep  9 20:39 ubuntu.sources
-rw-r--r-- 1 root root 2552 Aug 27 14:21 ubuntu.sources.curtin.orig
lag@ubuntu:/etc/apt/sources.list.d$ cd ~
lag@ubuntu:~$ sudo cp /etc/apt/sources.list.d /etc/apt/sources.list.backup
cp: -r not specified; omitting directory '/etc/apt/sources.list.d'
lag@ubuntu:~$ sudo cp -r /etc/apt/sources.list.d /etc/apt/sources.list.backup
```

It is important to check that everything copied correctly so I cd to the backup directory and verified the contents.

**Figure 1.4** Verification that October 5, 2024 backup included all files

```
lag@ubuntu:~$ cd /etc/apt/sources.list.backup
lag@ubuntu:/etc/apt/sources.list.backup$ ls -l
total 8
-rw-r--r-- 1 root root 386 Oct  5 12:25 ubuntu.sources
-rw-r--r-- 1 root root 2552 Oct  5 12:25 ubuntu.sources.curtin.orig
lag@ubuntu:/etc/apt/sources.list.backup$ _
```

In the `ubuntu.sources` file the repositories were listed. Elements of the repository were Types, uniform resource identifiers (URIs), Suites, Components, and Signed-By. For the repositories shown below in Figure 1.5, the “deb” Types illustrated that the type of files would be precompiled Debian source files. The URIs are internet links to where the repository is stored. The “Suites” section details the release name of the distribution (Ubuntu, 2015). The “Components” section illustrates the ways that the software is available to end users, and the “Signed-By” section illustrates a chain of key signatures that essentially says that the software is legitimate (Ubuntu, 2024). To add a new repository, you can use the command **sudo add-apt-repository ppa:<repository name>** (Ubuntu, 2015).

**Figure 1.5** Results showing Ubuntu’s built in repositories

```
GNU nano 7.2
Types: deb
URIs: http://us.archive.ubuntu.com/ubuntu/
Suites: noble noble-updates noble-backports
Components: main restricted universe multiverse
Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg

Types: deb
URIs: http://security.ubuntu.com/ubuntu/
Suites: noble-security
Components: main restricted universe multiverse
Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg
```

## Installation

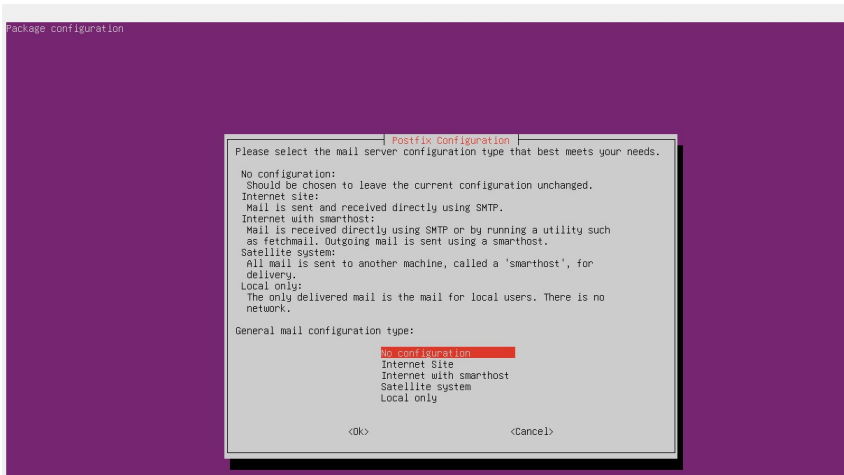
In this tutorial, the following software programs will be installed on the Ubuntu server: `tmux`, `emacs`, `fail2ban`, `cowsay`, `lolcat`, and `vim` (if not already installed). To install the programs, use the command **sudo apt install <program name>**. Before doing that, it is always good to update your system first. This can be done with the command **sudo apt update**.

**Figure 1.6** Image showing the results of **sudo apt update** on October 5, 2024

```
lag@ubuntu:~$ sudo apt update
[sudo] password for lag:
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:4 http://us.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:5 http://us.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [538 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [384 kB]
Fetched 1,047 kB in 1s (746 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
16 packages can be upgraded. Run 'apt list --upgradable' to see them.
lag@ubuntu:~$ sudo apt install tmux
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
tmux is already the newest version (3.4-1ubuntu0.1).
tmux set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 16 not upgraded.
lag@ubuntu:~$
```

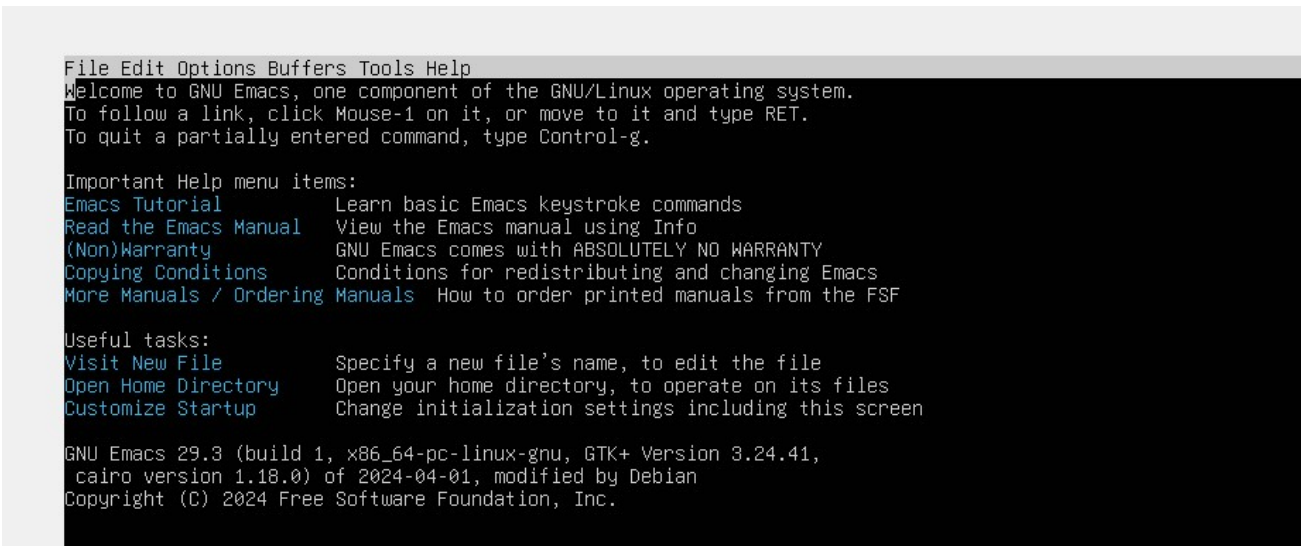


**Figure 1.11** Image of Postfix configuration options



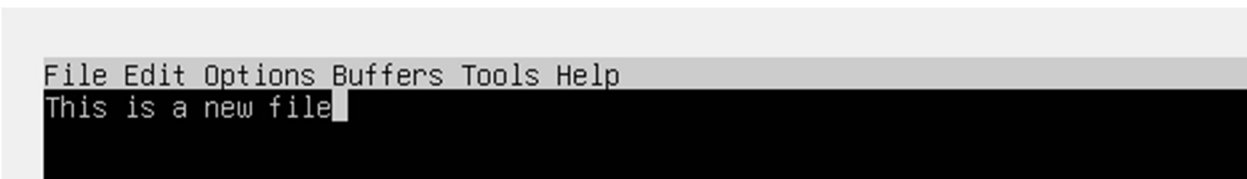
To verify that emacs installed correctly, type **emacs**. Next, the emacs menu will appear on the screen.

**Figure 1.12** Image showing the emacs help menu



To learn how to work in emacs you can read the tutorial, or select some of the links in the main menu. Below is an example of creating a new file from selecting the Visit New File option. To exit emacs, you have to select **Ctrl + X + C**.

**Figure 1.13** Image showing a new file created in emacs



To find where emacs is stored on the server, type the command **whereis emacs**.

**Figure 1.14** Image showing the results of **whereis emacs**



## Fail2ban

Fail2ban is a program that is used to secure a Linux system against malicious logins (Prakash, 2019). Fail2ban was installed by typing the command **sudo apt install fail2ban**.

**Figure 1.15** Image showing the installation of fail2ban

```
lag@ubuntu:~$ sudo apt install fail2ban
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-pyasyncore python3-pyinotify whois
Suggested packages:
  monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyasyncore python3-pyinotify whois
0 upgraded, 4 newly installed, 0 to remove and 16 not upgraded.
Need to get 496 kB of archives.
After this operation, 2,572 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

To Start fail2ban, use the command **systemctl start fail2ban**. As fail2ban starts, the program will require you to authenticate several times.

**Figure 1.16** Image showing fail2ban starting and the authentication requests.

```
lag@ubuntu:~$ systemctl start fail2ban
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'fail2ban.service'.
Authenticating as: lindsay (lag)
Password:
==== AUTHENTICATION COMPLETE ====
lag@ubuntu:~$ systemctl enable fail2ban
Synchronizing state of fail2ban.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable fail2ban
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: lindsay (lag)
Password:
==== AUTHENTICATION COMPLETE ====
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: lindsay (lag)
Password:
==== AUTHENTICATION COMPLETE ====
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ====
Authentication is required to manage system service or unit files.
Authenticating as: lindsay (lag)
Password:
==== AUTHENTICATION COMPLETE ====
lag@ubuntu:~$ _
```

Fail2ban has two primary configuration files – fail2ban.conf and jail.conf. The fail2ban.conf file is the file that manages the operational settings of the fail2ban daemon. The jail.conf is where a user can customize configurations like setting allowlists and denylists by typing directly into the file.



**Figure 1.17** Image showing the files in the fail2ban directory

```
lag@ubuntu:/etc/fail2ban$ ls -l
total 64
drwxr-xr-x 2 root root 4096 Oct 6 11:28 action.d
-rw-r--r-- 1 root root 3017 Nov 9 2022 fail2ban.conf
drwxr-xr-x 2 root root 4096 Jun 10 21:27 fail2ban.d
drwxr-xr-x 3 root root 4096 Oct 6 11:28 filter.d
-rw-r--r-- 1 root root 25607 Nov 9 2022 jail.conf
drwxr-xr-x 2 root root 4096 Oct 6 11:28 jail.d
-rw-r--r-- 1 root root 645 Nov 9 2022 paths-arch.conf
-rw-r--r-- 1 root root 2728 Nov 9 2022 paths-common.conf
-rw-r--r-- 1 root root 627 Nov 9 2022 paths-debian.conf
-rw-r--r-- 1 root root 738 Nov 9 2022 paths-opensuse.conf
```

**Figure 1.18** Image showing a snippet of the fail2ban.conf file

```
# Notes.: Set the socket file. This is used to communicate with the daemon. Do
#         not remove this file when Fail2ban runs. It will not be possible to
#         communicate with the server afterwards.
# Values: [ FILE ]   Default: /var/run/fail2ban/fail2ban.sock
#
socket = /var/run/fail2ban/fail2ban.sock

# Option: pidfile
# Notes.: Set the PID file. This is used to store the process ID of the
#         fail2ban server.
# Values: [ FILE ]   Default: /var/run/fail2ban/fail2ban.pid
#
pidfile = /var/run/fail2ban/fail2ban.pid
```

**Figure 1.19** Image showing a snippet of the jail.conf file

```
%(default/action_)s[name=%(__name__)s-udp, protocol="udp"]
logpath = /var/log/mumble-server/mumble-server.log

[screensharingd]
# For Mac OS Screen Sharing Service (VNC)
logpath = /var/log/system.log
logencoding = utf-8

[haproxy-http-auth]
# HAProxy by default doesn't log to file you'll need to set it up to forward
# logs to a syslog server which would then write them to disk.
# See "haproxy-http-auth" filter for a brief cautionary note when setting
# maxretry and findtime.
logpath = /var/log/haproxy.log
```

To see where fail2ban is on your linux system, use the command **whereis fail2ban**.

**Figure 1.20** Image showing the results of the whereis fail2ban program.

```
lag@ubuntu:~$ whereis fail2ban
fail2ban: /etc/fail2ban /usr/share/man/man1/fail2ban.1.gz
```

## Cowsay

Cowsay is a fun tool that can display a cow in the terminal along with a message (Sreenath, 2023). Cowsay was installed by typing the command **sudo apt install cowsay**.

**Figure 1.21** Image showing the installation of cowsay

```
lag@ubuntu:~$ sudo apt install cowsay
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  filters cowsay-off
The following NEW packages will be installed:
  cowsay
0 upgraded, 1 newly installed, 0 to remove and 16 not upgraded.
Need to get 18.6 kB of archives.
After this operation, 93.2 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/noble/universe amd64 cowsay all 3.03+dfsg2-8 [18.6 kB]
Fetched 18.6 kB in 0s (48.7 kB/s)
Selecting previously unselected package cowsay.
(Reading database ... 142789 files and directories currently installed.)
Preparing to unpack .../cowsay_3.03+dfsg2-8_all.deb ...
Unpacking cowsay (3.03+dfsg2-8) ...
Setting up cowsay (3.03+dfsg2-8) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
lag@ubuntu:~$
```

To use cowsay, you type the command **cowsay** followed by text. To test that cowsay was installed correctly, I typed **cowsay Good Morning!**

**Figure 1.22** Image showing the results of **cowsay Good Morning!**

```
lag@ubuntu:~$ cowsay Good Morning!
-----
< Good Morning! >
-----
      \   ^__^
       (oo)\_______
          (_____)  )\/\
              ||----w |
              ||     ||

lag@ubuntu:~$
```

You can also use options with cowsay. For instance, the -g option shows a “greedy” cow with dollar signs for eyes.

**Figure 1.23** Image showing the results of cowsay with the -g option saying “I want all the food!”

```
lag@ubuntu:~$ cowsay -g I want all the food!
-----
< I want all the food! >
-----
      \   ^__^
       ($$)\_______
          (_____)  )\/\
              ||----w |
              ||     ||

lag@ubuntu:~$ _
```

To see where cowsay is located, use the **whereis** command.

**Figure 1.24** Results of **whereis cowsay** command

```
lag@ubuntu:~$ whereis cowsay
cowsay: /usr/games/cowsay /usr/share/cowsay /usr/share/man/man6/cowsay.6.gz
```

**Lolcat**

Lolcat is another fun program that can add a rainbow of colors to the Linux terminal (Saive, 2023). Lolcat was installed by typing the command **sudo apt install lolcat**.

**Figure 1.25** Image showing the installation of lolcat

```
lag@ubuntu:~$ sudo apt install lolcat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fonts-lato javascript-common libjs-jquery libruby libruby3.2 rake ruby ruby-net-telnet ruby-optimist ruby-paint ruby-rubygems ruby-sdbm ruby-webrick
  ruby-xmlrpc ruby3.2 rubygems-integration unzip zip
Suggested packages:
  apache2 | lighttpd | httpd ri ruby-dev bundler
The following NEW packages will be installed:
  fonts-lato javascript-common libjs-jquery libruby libruby3.2 lolcat rake ruby ruby-net-telnet ruby-optimist ruby-paint ruby-rubygems ruby-sdbm ruby-webrick
  ruby-xmlrpc ruby3.2 rubygems-integration unzip zip
0 upgraded, 19 newly installed, 0 to remove and 16 not upgraded.
Need to get 9,311 kB of archives.
After this operation, 42.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

To see if lolcat installed correctly, I typed the command **lolcat -h** to view the help menu.

**Figure 1.26** Image showing the lolcat help menu

```
lag@ubuntu:~$ lolcat -h
Usage: lolcat [OPTION]... [FILE]...

Concatenate FILE(s), or standard input, to standard output.
With no FILE, or when FILE is -, read standard input.

-p, --spread=<f>      Rainbow spread (default: 3.0)
-F, --freq=<f>        Rainbow frequency (default: 0.1)
-S, --seed=<i>        Rainbow seed, 0 = random (default: 0)
-a, --animate         Enable psychedelics
-d, --duration=<i>    Animation duration (default: 12)
-s, --speed=<f>       Animation speed (default: 20.0)
-i, --invert          Invert fg and bg
-t, --truecolor       24-bit (truecolor)
-f, --force           Force color even when stdout is not a tty
-v, --version         Print version and exit
-h, --help            Show this message

Examples:
lolcat f - g          Output f's contents, then stdin, then g's contents.
lolcat                Copy standard input to standard output.
fortune | lolcat      Display a rainbow cookie.

Report lolcat bugs to <https://github.com/buysiooq/lolcat/issues>
lolcat home page: <https://github.com/buysiooq/lolcat/>
Report lolcat translation bugs to <http://speaklolcat.com/>

lag@ubuntu:~$ _
```

Lolcat can be used to read files like cat, or can be used in the terminal with other commands like cowsay. Below is an example of cowsay used with lolcat.

**Figure 1.27** Image of cow and colored text saying “Can this cow change colors?”

```
lag@ubuntu:~$ cowsay Can this cow change colors? | lolcat
< Can this cow change colors? >
  \      ^__^
   (oo)\_______
      (__)\       )\/\
         ||----w |
         ||     ||
```



To see where lolcat is located, use the **whereis** command.

**Figure 1.28** Results of **whereis lolcat** command

```
lag@ubuntu:~$ whereis lolcat
lolcat: /usr/games/lolcat /usr/share/man/man6/lolcat.6.gz
lag@ubuntu:~$ _
```

### Vim (Vi Improved)

Vim is a powerful text editor that was already installed on the Ubuntu server. Some of vim's top features are that it can support hundreds of programming languages and file formats, can be used for search and replace, and can integrate with other tools (Vim, 2024). Type **vim** to open vim in the terminal.

**Figure 1.29** Image showing the results of typing vim in the terminal

```

      Vim - Vi Improved
      version 9.1.697
      by Bram Moolenaar et al.
      Modified by team@vimtrucker.debian.org
      Vim is open source and freely distributable

      Help poor children in Uganda!
      type :help iccf<Enter>      for information

      type :q<Enter>              to exit
      type :help<Enter> or :h<Enter> for on-line help
      type :help version9<Enter> for version info

lag@ubuntu:~$
```

To create a file you can type **:i** to go to insert mode.

**Figure 1.30** Example of text typed into a vim file using insert mode.

```
jjojo
~
~
~
```

To exit vim, type **:q**.

To see where the vim files are located, use the **whereis vim** command.

**Figure 1.31** Results of **whereis vim** command

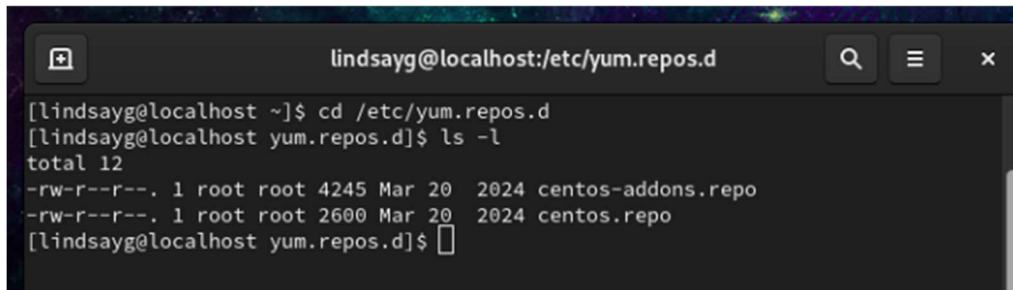
```
lag@ubuntu:~$ whereis vim
vim: /usr/bin/vim /etc/vim /usr/share/vim /usr/share/man/man1/vim.1.gz
lag@ubuntu:~$ _
```

## Part 2: CentOS Server: Installs, Updates, and Documentation

### Repositories

Linux distributions have built in repositories that have all the packages needed to run compatible software programs (Holden-Gouveia, 2024). They are a way to collect and organize software so that it can be easily installed. For the CentOS server, the built-in repositories are located in the `/etc/yum.repos.d` directory (Naver Cloud, 2024). To view what is inside `/etc/yum.repos.d` directory, change to that directory and view the contents of that directory by using the command `ls -l`.

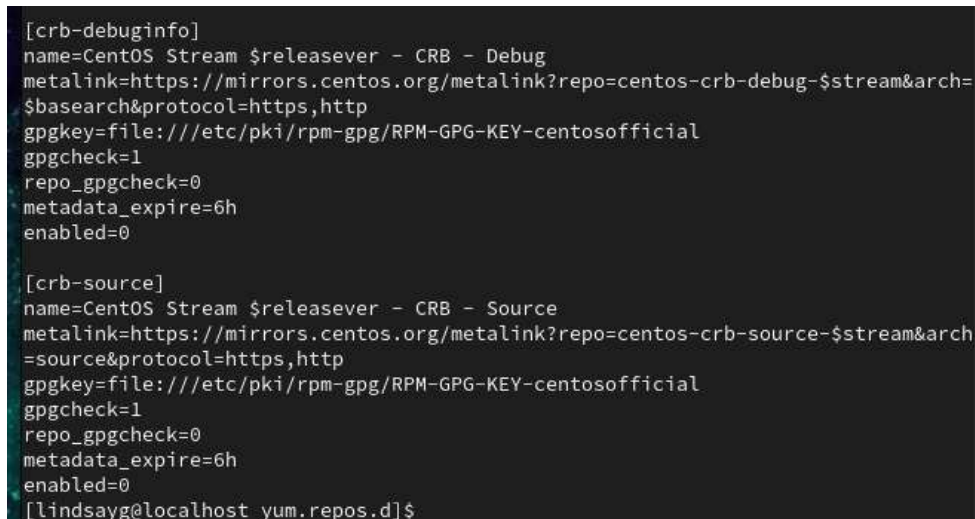
**Figure 2.1** Image showing the contents of the `/etc/yum.repos.d`



```
lindsayg@localhost:~$ cd /etc/yum.repos.d
lindsayg@localhost yum.repos.d$ ls -l
total 12
-rw-r--r--. 1 root root 4245 Mar 20 2024 centos-addons.repo
-rw-r--r--. 1 root root 2600 Mar 20 2024 centos.repo
lindsayg@localhost yum.repos.d$
```

The CentOS working repositories are in the `centos.repo` file. The additional CentOS repositories are in the `centos-addons.repo` file.

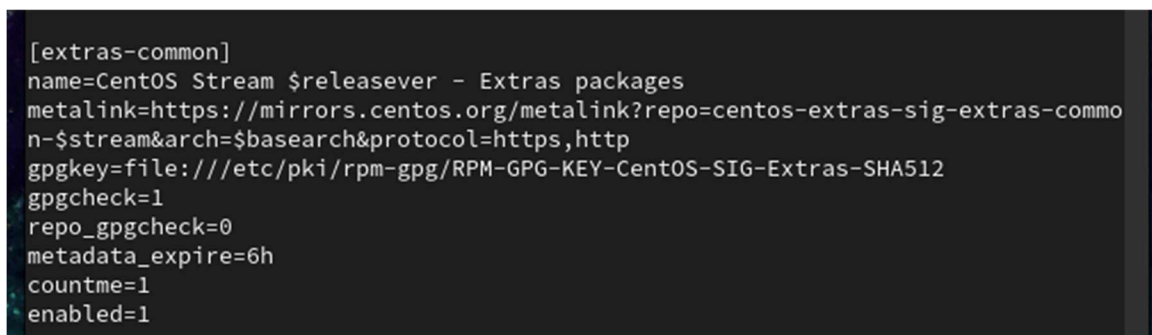
**Figure 2.2** A snippet of the results of `cat centos.repo`



```
[crb-debuginfo]
name=CentOS Stream $releasever - CRB - Debug
metalink=https://mirrors.centos.org/metalink?repo=centos-crb-debug-$stream&arch=$basearch&protocol=https,http
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
gpgcheck=1
repo_gpgcheck=0
metadata_expire=6h
enabled=0

[crb-source]
name=CentOS Stream $releasever - CRB - Source
metalink=https://mirrors.centos.org/metalink?repo=centos-crb-source-$stream&arch=source&protocol=https,http
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
gpgcheck=1
repo_gpgcheck=0
metadata_expire=6h
enabled=0
lindsayg@localhost yum.repos.d$
```

**Figure 2.3** A snippet of the results of `cat centos-addons.repo`



```
[extras-common]
name=CentOS Stream $releasever - Extras packages
metalink=https://mirrors.centos.org/metalink?repo=centos-extras-sig-extras-common-$stream&arch=$basearch&protocol=https,http
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-SIG-Extras-SHA512
gpgcheck=1
repo_gpgcheck=0
metadata_expire=6h
countme=1
enabled=1
```

The structure of the repo files includes a header, or repo id shown in brackets at the top (Reselman, 2022). Next is the name, in this case CentOS Stream. The next line is a metalink link that points to an XML document that lists the locations where the repository data can be found. After that is a gpgkey which is used to verify the authenticity of the software. Next is the gpgcheck which indicates that the system will verify the digital signatures of the package before installing them. The next setting is repo\_gpgcheck which indicates whether GPG checking is set for the repository. The default for repo\_gpgcheck is zero, or false. The metadata\_expire indicates how often the repository metadata should be refreshed. This document also includes information that can be used to verify the accuracy and integrity of the data in those repositories. The enabled indicator defines whether the repository is enabled for use. The countme line setting indicates the number of times a repository has been used. To create a new repository you can use the command **yum install <repository URL>** (Marrich, 2020).

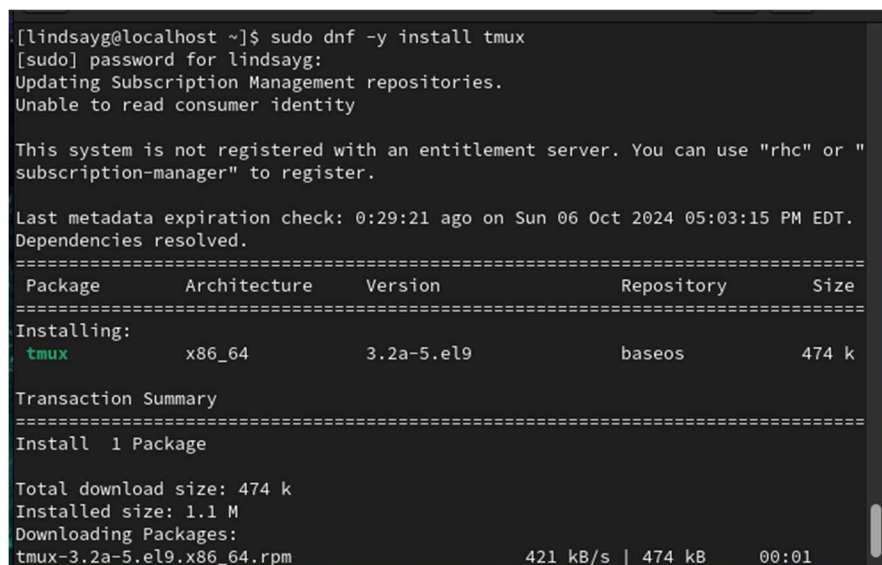
## Installation

In this tutorial, the following software programs will be installed on the CentOS server: tmux, emacs, fail2ban, cowsay, lolcat, and vim (if not already installed). To install the programs, you can use the command **sudo dnf -y install <program name>** (Marijan, 2022). The dnf is a newer package management tool that aims to replace yum in new CentOS distributions (Marijan, 2023). The -y option automatically adds yes to all prompts. Before doing that, it is always good to update your system first. This can be done with the command **sudo yum update**. The CentOS system was updated on October 6, 2024.

## Tmux

Tmux is a terminal multiplexer, which means that it can allow a user to create multiple “pseudo terminals” from a single terminal window (Gerardi, 2022). Tmux sessions can be detached from the terminal and run in the background which can be helpful for long running processes. Tmux was installed by typing the command **sudo dnf -y install tmux**.

**Figure 2.4** Image showing the tmux installation



```
[lindsayg@localhost ~]$ sudo dnf -y install tmux
[sudo] password for lindsayg:
Updating Subscription Management repositories.
Unable to read consumer identity

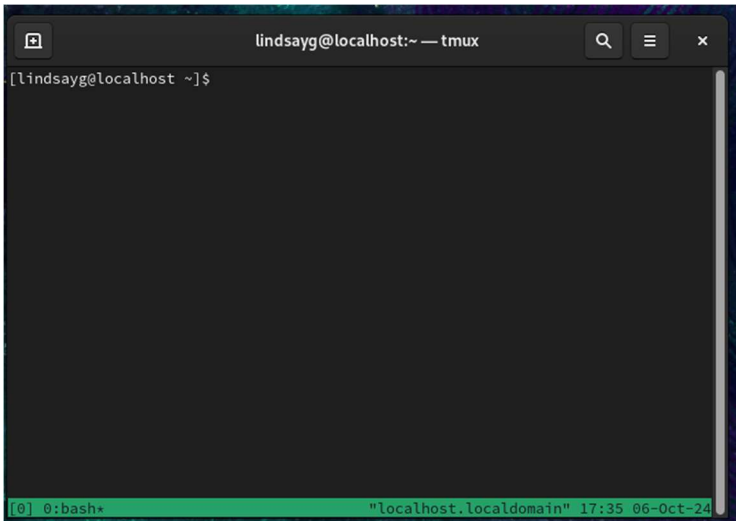
This system is not registered with an entitlement server. You can use "rhc" or "
subscription-manager" to register.

Last metadata expiration check: 0:29:21 ago on Sun 06 Oct 2024 05:03:15 PM EDT.
Dependencies resolved.
=====
Package           Architecture    Version           Repository        Size
=====
Installing:
tmux              x86_64         3.2a-5.el9        baseos            474 k
Transaction Summary
=====
Install 1 Package

Total download size: 474 k
Installed size: 1.1 M
Downloading Packages:
tmux-3.2a-5.el9.x86_64.rpm                421 kB/s | 474 kB    00:01
```

To verify that tmux installed correctly, you can type tmux in the command line to open the program. That will connect to the tmux server and open a new session (0) in the window.

**Figure 2.5** Snippet showing the tmux screen



To detach from the session select **Ctrl+B** then **D**.

**Figure 2.6** Image showing the screen after detaching from the session



To find out where the tmux files are located, use the command **whereis tmux**. The image below shows where the tmux files are located.

**Figure 2.7** Image showing location of tmux files

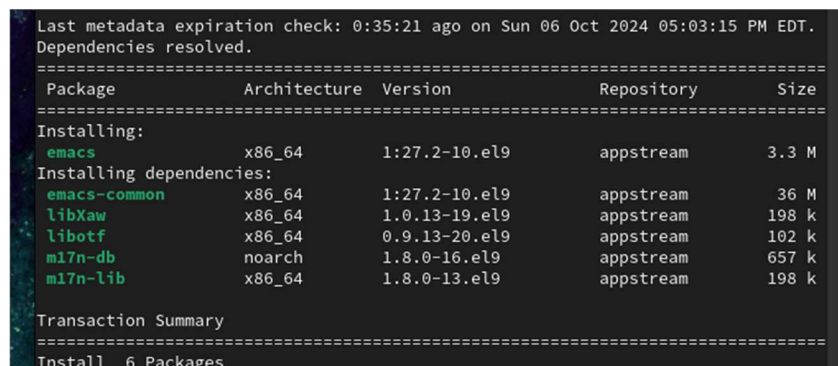


## Emacs

Emacs is a popular, text editor that can also function as an integrated development environment (IDE) for different programming languages (Hussain, 2023). Emacs was installed by typing the command **sudo dnf install emacs**.

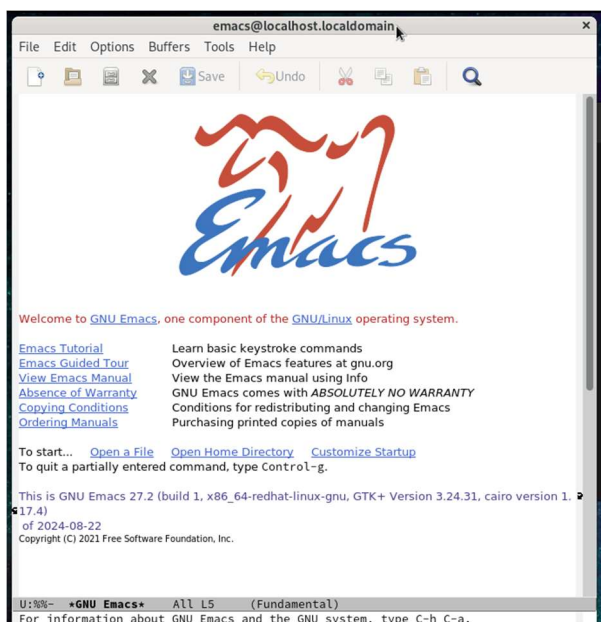
The -y option did not work for this program, as there was an error caused by a plugin that interfered.

**Figure 2.8** Snippet of emacs installation



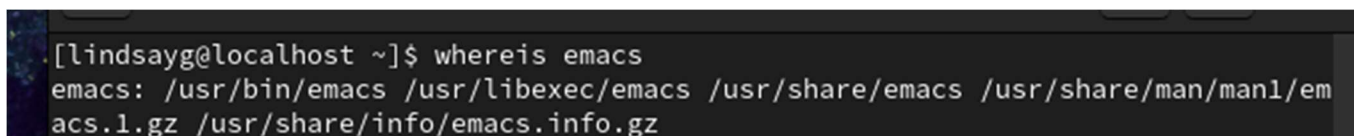
To verify that emacs installed correctly, type **emacs**. On CentOS, the emacs graphical user interface (GUI) will appear on the screen.

**Figure 2.9** Image showing the emacs GUI



Because emacs has the GUI, it is very easy to open a file, start a new file, and exit. There was not an option for a command line only install. To find where emacs is stored on the server, type the command **whereis emacs**. Below is an image showing where the various emacs files are stored on the server.

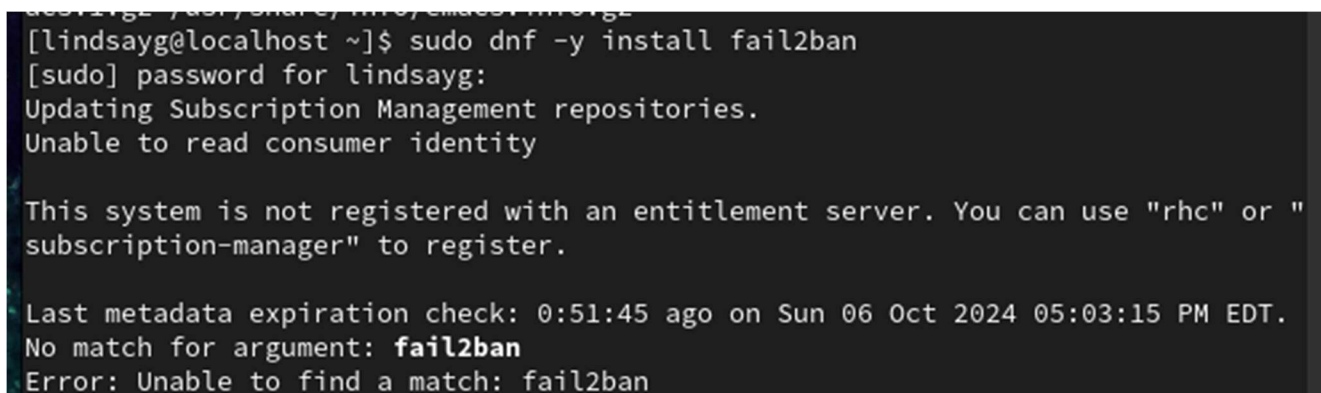
**Figure 2.10** Image showing the results of **whereis emacs**



## Fail2ban

Fail2ban is a program that is used to secure a Linux system against malicious logins (Prakash, 2019). I attempted to install Fail2ban by typing the command **sudo dnf -y install fail2ban**. I received an error that said there was no match for fail2ban.

**Figure 2.11** Image showing the failed installation of fail2ban





To install fail2ban on CentOS, you have to first install the Extra Packages for Enterprise Linux (EPEL) (UpCloud, 2024). This can be done through the command **sudo yum install epel-release**.

**Figure 2.12** Snippet of the successful installation of EPEL

```
Installed:
  epel-next-release-9-7.el9.noarch      epel-release-9-7.el9.noarch

Complete!
[lindsayg@localhost ~]$
```

Next, install fail2ban by typing the command **sudo yum install fail2ban**.

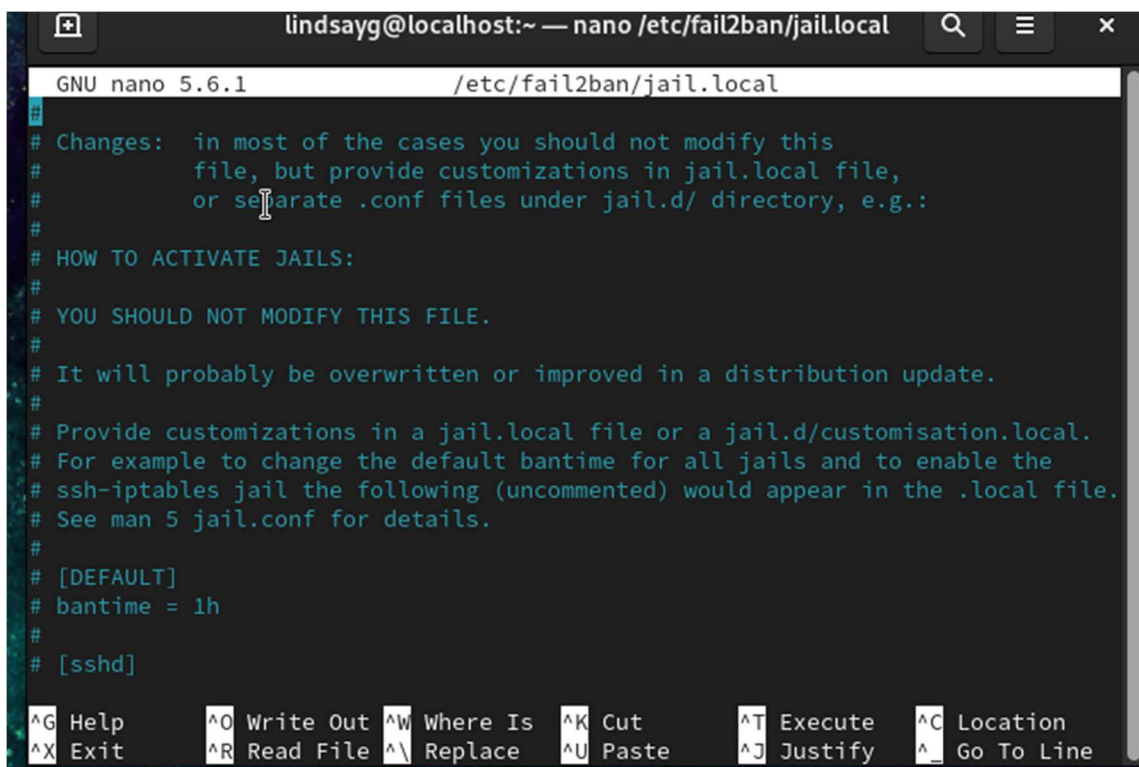
**Figure 2.13** Snippet of the successful installation of fail2ban

```
Installed:
  esmtp-1.2-19.el9.x86_64      fail2ban-1.0.2-12.el9.noarch
  fail2ban-firewalld-1.0.2-12.el9.noarch  fail2ban-selinux-1.0.2-12.el9.noarch
  fail2ban-sendmail-1.0.2-12.el9.noarch  fail2ban-server-1.0.2-12.el9.noarch
  libesmtp-1.0.6-24.el9.x86_64  liblockfile-1.14-10.el9.x86_64

Complete!
[lindsayg@localhost ~]$
```

After installation, make a copy of the jail.conf primary configuration file and save locally. This can be done with the command **sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local**. It is recommended to customize the local configuration file as the primary configuration file will get overwritten or improved as the distribution is updated. To view and edit the local configuration file, open it up using a text editor.

**Figure 2.14** An image showing the jail.local file opened in nano



```
lindsayg@localhost:~ — nano /etc/fail2ban/jail.local
GNU nano 5.6.1 /etc/fail2ban/jail.local
# Changes:  in most of the cases you should not modify this
#           file, but provide customizations in jail.local file,
#           or separate .conf files under jail.d/ directory, e.g.:
#
# HOW TO ACTIVATE JAILS:
#
# YOU SHOULD NOT MODIFY THIS FILE.
#
# It will probably be overwritten or improved in a distribution update.
#
# Provide customizations in a jail.local file or a jail.d/customisation.local.
# For example to change the default bantime for all jails and to enable the
# ssh-iptables jail the following (uncommented) would appear in the .local file.
# See man 5 jail.conf for details.
#
# [DEFAULT]
# bantime = 1h
#
# [sshd]
```

<sup>^</sup>G Help    <sup>^</sup>O Write Out    <sup>^</sup>W Where Is    <sup>^</sup>K Cut    <sup>^</sup>T Execute    <sup>^</sup>C Location  
<sup>^</sup>X Exit    <sup>^</sup>R Read File    <sup>^</sup>\ Replace    <sup>^</sup>U Paste    <sup>^</sup>J Justify    <sup>^</sup>\_ Go To Line



To start fail2ban, use the command **sudo systemctl start fail2ban**. To see the status of fail2ban type **systemctl status fail2ban**. To navigate away from this screen use **Ctrl + C**. To

**Figure 2.15** The image below shows the status of fail2ban running on the CentOS system.

```
[lindsayg@localhost ~]$ systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
   Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; disabled; preset>
   Active: active (running) since Sun 2024-10-06 18:16:56 EDT; 3min 24s ago
     Docs: man:fail2ban(1)
   Process: 107346 ExecStartPre=/bin/mkdir -p /run/fail2ban (code=exited, stat>
  Main PID: 107347 (fail2ban-server)
    Tasks: 3 (limit: 10949)
   Memory: 16.5M
      CPU: 106ms
   CGroup: /system.slice/fail2ban.service
           └─107347 /usr/bin/python3 -s /usr/bin/fail2ban-server -xf start

Oct 06 18:16:56 localhost.localdomain systemd[1]: Starting Fail2Ban Service...
Oct 06 18:16:56 localhost.localdomain systemd[1]: Started Fail2Ban Service.
Oct 06 18:16:56 localhost.localdomain fail2ban-server[107347]: Server ready
lines 1-15/15 (END)
```

To see where fail2ban is on your linux system, use the command **whereis fail2ban**.

**Figure 2.16** Image showing the results of the whereis fail2ban program.

```
[lindsayg@localhost ~]$ whereis fail2ban
fail2ban: /etc/fail2ban /usr/share/man/man1/fail2ban.1.gz
```

To stop the service use the command **sudo systemctl stop fail2ban**. To disable it from automatically starting up use **sudo systemctl disable fail2ban**.

### Cowsay

Cowsay is a fun tool that can display a cow in the terminal along with a message (Sreenath, 2023). Cowsay was installed by typing the command **sudo dnf -y install cowsay**.

**Figure 2.17** Image showing a snippet from the installation of cowsay

```
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing     : cowsay-3.7.0-10.el9.noarch     1/1
  Running scriptlet: cowsay-3.7.0-10.el9.noarch   1/1
  Verifying      : cowsay-3.7.0-10.el9.noarch     1/1
Installed products updated.

Installed:
  cowsay-3.7.0-10.el9.noarch

Complete!
[lindsayg@localhost ~]$
```

To use cowsay, you type the command **cowsay** followed by text. To test that cowsay was installed correctly, I typed **cowsay Will this work the same?**

**Figure 2.18** Image showing the results of **cowsay Will this work the same?**



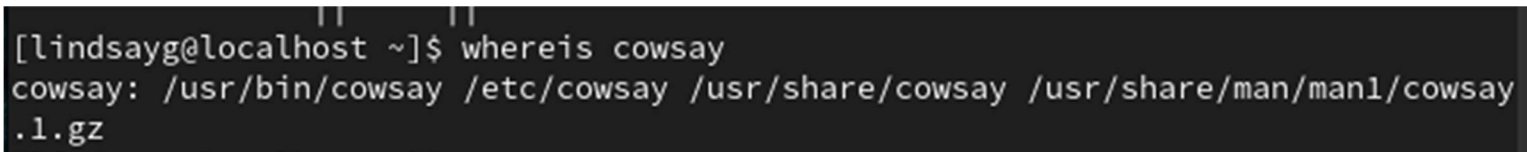
```

[~] lindsayg@localhost:~
[~] lindsayg@localhost ~]$ cowsay Will this work the same?
-----
< Will this work the same? >
-----
      \   ^__^
       (oo)\_______
          (__)\       )\/\
              ||----w |
              ||     ||
[~] lindsayg@localhost ~]$

```

To see where cowsay is located, use the **whereis** command.

**Figure 2.19** Results of **whereis cowsay** command



```

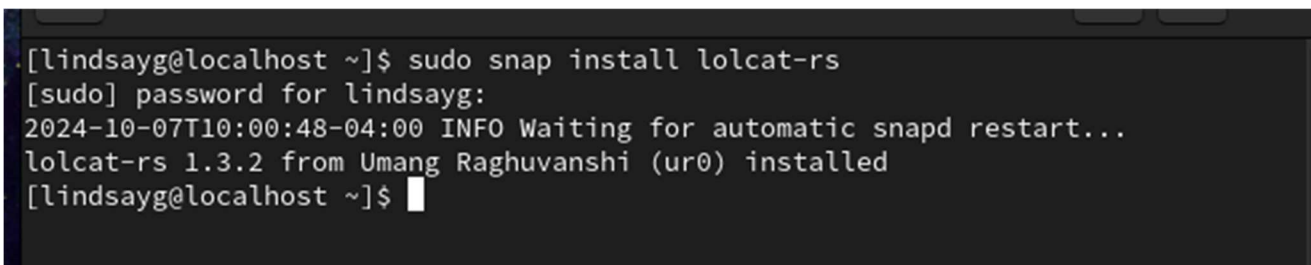
[~] lindsayg@localhost ~]$ whereis cowsay
cowsay: /usr/bin/cowsay /etc/cowsay /usr/share/cowsay /usr/share/man/man1/cowsay
        .1.gz

```

## Lolcat

Lolcat is another fun program that can add a rainbow of colors to the Linux terminal (Saive, 2023). To install lolcat on the CentOS, a few steps needed to happen first (Raghuvanshi, 2024). The first step was to install the epel-release repository, but that was already done in a prior installation on October 6, 2024. The next step was to install the Snap app store that contains Linux applications such as lolcat. The commands to install snap and enable communication were **sudo yum install snapd**, **sudo systemctl enable --now snapd.socket**, and **sudo ln -s /var/lib/snapd/snap /snap**. To ensure that the paths update correctly, it is recommended to restart your system. After the restart lolcat can be installed with the command **sudo snap install lolcat-rs**. The -rs at the end represents that this is the Rust re-implementation of the original lolcat which is newer and has no dependencies.

**Figure 2.20** Image showing the installation of lolcat



```

[~] lindsayg@localhost ~]$ sudo snap install lolcat-rs
[sudo] password for lindsayg:
2024-10-07T10:00:48-04:00 INFO Waiting for automatic snapd restart...
lolcat-rs 1.3.2 from Umang Raghuvanshi (ur0) installed
[~] lindsayg@localhost ~]$

```

Lolcat can be used to read files like the **cat** command, or can be used in the terminal with other commands like **cowsay**. Below is an example of cowsay and echo being used with lolcat-rs.

**Figure 2.21** Image of echo command and “Hello!” with a cow and colored text saying “It works!”

```
[lindsayg@localhost ~]$ echo Hello! | lolcat-rs
Hello!
[lindsayg@localhost ~]$ cowsay It works! | lolcat-rs
-----
< It works! >
-----
      ^__^
      (oo)\_______
      (--)\/       )\/\
      ||----w |
      ||     ||
```

To see where lolcat is located, use the **whereis** command.

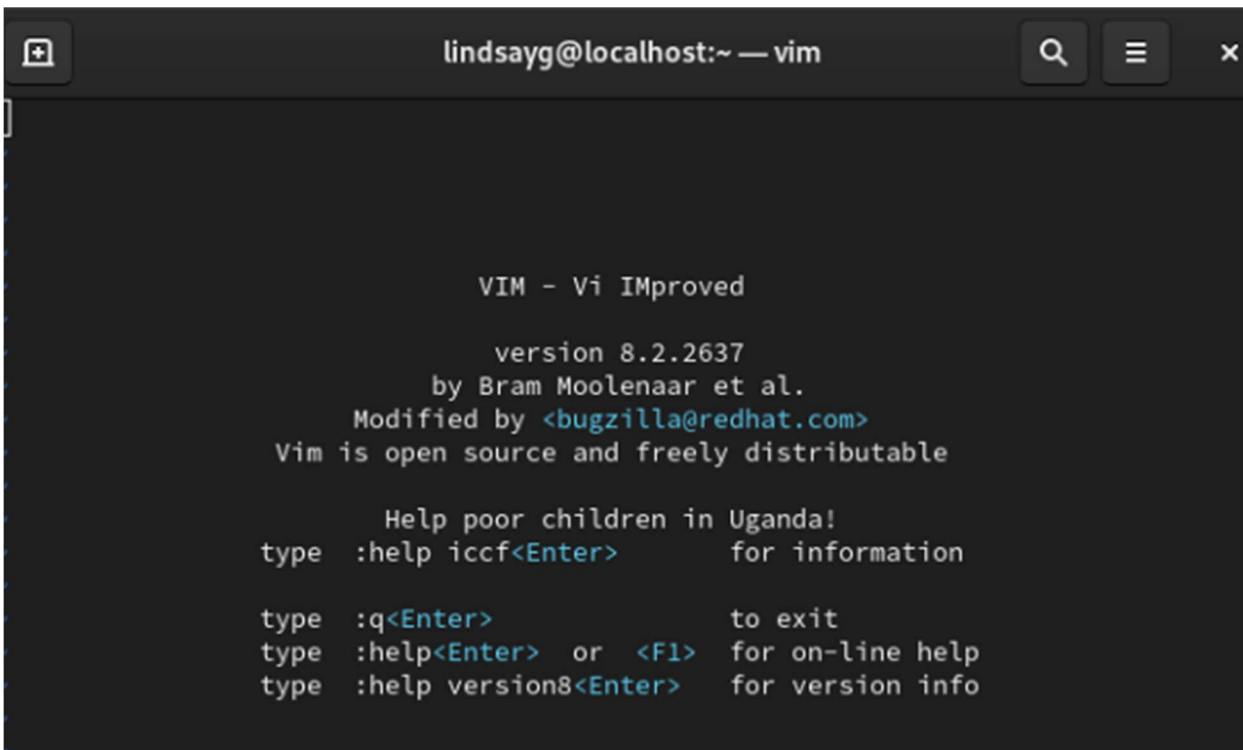
**Figure 2.22** Results of **whereis lolcat-rs** command

```
[lindsayg@localhost ~]$ whereis lolcat-rs
lolcat-rs: /var/lib/snapd/snap/bin/lolcat-rs
[lindsayg@localhost ~]$
```

### Vim (Vi Improved)

Vim is a powerful text editor that was already installed on the CentOS server. Some of vim’s top features are that it can support hundreds of programming languages and file formats, can be used for search and replace, and can integrate with other tools (Vim, 2024). Type **vim** to open vim in the terminal.

**Figure 2.23** Image showing the results of typing vim in the terminal



```
lindsayg@localhost:~ — vim

VIM - Vi IMproved

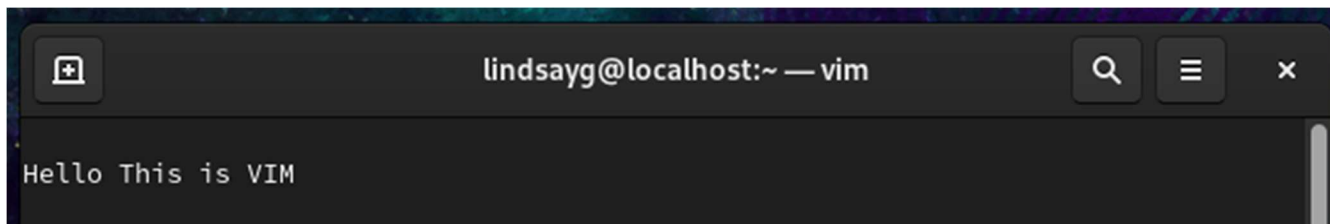
version 8.2.2637
by Bram Moolenaar et al.
Modified by <bugzilla@redhat.com>
Vim is open source and freely distributable

Help poor children in Uganda!
type  :help iccf<Enter>      for information

type  :q<Enter>              to exit
type  :help<Enter> or <F1>   for on-line help
type  :help version8<Enter> for version info
```

To create a file you can type **:i** to go to insert mode.

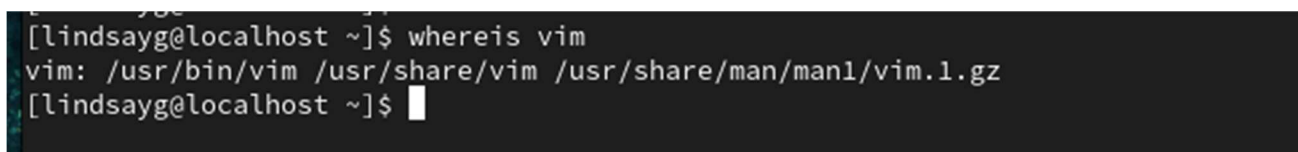
**Figure 2.24** Example of text typed in a vim file using insert mode.



To exit vim, type **:q!**.

To see where the vim files are located, use the **whereis vim** command.

**Figure 2.25** Results of **whereis vim** command



## References

- Drake, M., & Camisso, J. (2022, April 5). *How To Install and Configure Postfix on Ubuntu 22.04* | DigitalOcean. [Www.digitalocean.com. https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-postfix-on-ubuntu-22-04](https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-postfix-on-ubuntu-22-04)
- Gerardi, R. (2022, September 13). *A beginner's guide to tmux*. Enable Sysadmin. <https://www.redhat.com/sysadmin/introduction-tmux-linux>
- Holden-Gouveia, A. (2024). *Installs and updates*. Aholdengouveia.name. <https://www.aholdengouveia.name/LinuxAdmin/installs.html>
- Hussain, S. (2023, January 4). *How To Use the Emacs Editor in Linux*. DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-use-the-emacs-editor-in-linux>
- Marijan, B. (2022, November 24). *YUM vs. APT: What's the Difference?* | phoenixNAP KB. Knowledge Base by PhoenixNAP. <https://phoenixnap.com/kb/yum-vs-apt>
- Marijan, B. (2023, November 16). *DNF vs. YUM: Learn the Differences {Side-by-Side Comparison}*. Knowledge Base by PhoenixNAP. <https://phoenixnap.com/kb/dnf-vs-yum>
- Marrich, A. (2020, February 5). *How to add a Yum repository*. Enable Sysadmin. <https://www.redhat.com/sysadmin/add-yum-repository>
- Naver Cloud. (2024). *Check Linux OS repository settings*. Ncloud-Docs.com. <https://guide.ncloud-docs.com/docs/en/linux-os-repository-check>
- Prakash, A. (2019, March 26). *How to Install and Configure Fail2Ban to Secure Linux Server*. Linux Handbook. <https://linuxhandbook.com/fail2ban-basic/>
- Pun, B. K. (2020, April 22). *What's inside /etc/apt/sources.list*. DEV Community. <https://dev.to/bijaykumarpun/what-s-inside-etc-apt-sources-list-2h1k>
- Raghuvanshi, U. (2024, February 6). *lolcat-rs*. Snapcraft. <https://snapcraft.io/install/lolcat-rs/centos>
- Reselman, B. (2022, October 7). *What's inside an RPM .repo file?* | Red Hat Developer. Red Hat Developer. [https://developers.redhat.com/articles/2022/10/07/whats-inside-rpm-repo-file#anatomy\\_of\\_a\\_repo\\_file](https://developers.redhat.com/articles/2022/10/07/whats-inside-rpm-repo-file#anatomy_of_a_repo_file)
- Saive, R. (2023, June 22). *Lolcat - Make Your Linux Terminal More Colorful and Fun*. Lolcat – Display Text in Rainbow Colors in Linux Terminal. <https://www.tecmint.com/lolcat-color-output-linux-terminal/>
- Sreenath. (2023, July 5). *Using Cowsay Linux Command Like a Pro*. It's FOSS; It's FOSS. <https://itsfoss.com/cowsay/>
- Ubuntu. (2015, July 27). *Repositories/CommandLine - Community Help Wiki*. Ubuntu.com. [https://help.ubuntu.com/community/Repositories/CommandLine#Explanation\\_of\\_the\\_Repository\\_Format](https://help.ubuntu.com/community/Repositories/CommandLine#Explanation_of_the_Repository_Format)
- Ubuntu. (2024). *SecureApt - Community Help Wiki*. Ubuntu.com. <https://help.ubuntu.com/community/SecureApt>
- UpCloud. (2024, September 17). *How to install Fail2ban on CentOS*. UpCloud. <https://upcloud.com/resources/tutorials/install-fail2ban-centos>

Vim. (2024). *welcome home : vim online*. Www.vim.org. <https://www.vim.org/>