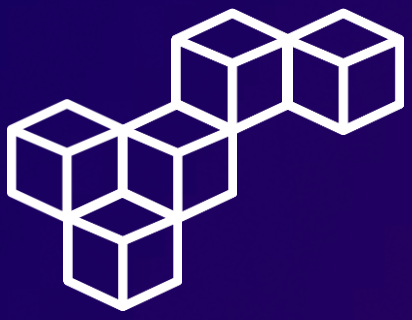
AWS re:Start LAB

Editing Files

WEEK 2

aws re/start



Overview

Editing files in Linux involves manipulating text or code within files using various command-line tools and techniques. This process is fundamental for system administrators, developers, and anyone working extensively in a Linux environment. Whether it's modifying configuration files, scripts, or documents, knowing how to effectively edit files is crucial for managing a Linux system.

One of the primary methods for editing files in Linux is through terminal-based text editors, which offer powerful functionalities for navigating, modifying, and saving changes to files directly from the command line. These editors provide a streamlined workflow for making quick adjustments or extensive revisions to text-based files without relying on graphical user interfaces.

Note: This lab was made using Windows Subsystem for Linux.

Topics covered

- Use the vimtutor executable to complete lessons 1-3
- Edit a text file with the Vim text editor
- Edit a text file with the nano text editor

Task 1

Use SSH to connect to an Amazon Linux EC2 instance

Initial Preparations

In the AWS Management Console, select the EC2 instance and make note of the **Public IPv4 address**.

Download the private key file **labsuser.pem**. Change to the Downloads directory and modify the permissions on the key to be read-only (r--------).

Connect to the instance using SSH

Establish a connection to the EC2 instance using the ssh command, the key and the instance's public IPv4 address.

```
support@HP-Pavilion-Laptop:~/Downloads$ ssh -i labsuser.pem ec2-user@35.94.7.190
The authenticity of host '35.94.7.190 (35.94.7.190)' can't be established.
ED25519 key fingerprint is SHA256:i2iCWrzrBjKL9egiQckE2bmIHApRzu0LRH57qnt4giU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '35.94.7.190' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-10-156 ~]$
```

aws re/start

Task 2

Run the Vim tutorial

Start vimtutor

Enter the vimtutor command to start the vimtutor session. If vimtutor does not work, you may need to install Vim by entering the command sudo yum install vim.

```
[ec2-user@ip-10-0-10-156 ~]$ vimtutor
```

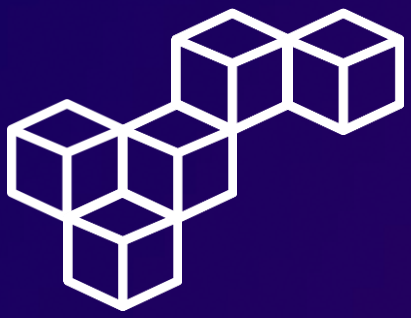
```
===============================================================================
=    W e l c o m e   t o   t h e   V I M   T u t o r    -    Version 1.7     =
===============================================================================

     Vim is a very powerful editor that has many commands, too many to
     explain in a tutor such as this.  This tutor is designed to describe
     enough of the commands that you will be able to easily use Vim as
     an all-purpose editor.
```

Complete the first three lessons in vimtutor.

Lesson 1:

```
                         Lesson 1 SUMMARY


  1. The cursor is moved using either the arrow keys or the hjkl keys.
        h (left)       j (down)       k (up)       l (right)

  2. To start Vim from the shell prompt type:  vim FILENAME <ENTER>

  3. To exit Vim type:     <ESC>   :q!   <ENTER>  to trash all changes.
             OR type:      <ESC>   :wq   <ENTER>  to save the changes.

  4. To delete the character at the cursor type:  x

  5. To insert or append text type:
        i   type inserted text   <ESC>         insert before the cursor
        A   type appended text   <ESC>         append after the line

NOTE: Pressing <ESC> will place you in Normal mode or will cancel
      an unwanted and partially completed command.
```

aws re/start



Task 2

Run the Vim tutorial

Lesson 2:

```
                        Lesson 2 SUMMARY

1. To delete from the cursor up to the next word type:        dw
2. To delete from the cursor up to the end of the word type:  de
3. To delete from the cursor to the end of a line type:       d$
4. To delete a whole line type:                               dd

5. To repeat a motion prepend it with a number:   2w
6. The format for a change command is:
          operator   [number]   motion
   where:
     operator - is what to do, such as  d  for delete
     [number] - is an optional count to repeat the motion
     motion   - moves over the text to operate on, such as  w (word),
                e (end of word),  $ (end of the line), etc.

7. To move to the start of the line use a zero:  0

8. To undo previous actions, type:           u  (lowercase u)
   To undo all the changes on a line, type:  U  (capital U)
   To undo the undos, type:                  CTRL-R
```

Lesson 3:

```
                        Lesson 3 SUMMARY


1. To put back text that has just been deleted, type   p .  This puts the
   deleted text AFTER the cursor (if a line was deleted it will go on the
   line below the cursor).

2. To replace the character under the cursor, type   r   and then the
   character you want to have there.

3. The change operator allows you to change from the cursor to where the
   motion takes you.  eg. Type  ce  to change from the cursor to the end of
   the word,  c$  to change to the end of a line.

4. The format for change is:

       c   [number]   motion
```

Enter :q! and press Enter to exit vimtutor.

aws re/start

Task 3

Edit a file in Vim

Step 1: Open the Vim text editor

Use vim to create and edit a file called **helloworld**. You will enter Vim Command Mode by default.

```
[ec2-user@ip-10-0-10-156 ~]$ vim helloworld
```

```
~
~
~
"helloworld" [New]                                0,0-1         All
```

Step 2: Insert a few lines

Press i to use Vim Insert Mode and write a few of lines of text. Then, press ESC to exit Insert Mode.

```
Hello World!
This is my first file in Linux and I am aditing it in Vim!
~
~
-- INSERT --                                      2,59          All
```

Step 3: Save your changes

In the Command Mode, enter the command :wq to save your changes to the file and quit.

```
Hello World!
This is my first file in Linux and I am aditing it in Vim!
~
~
:wq
```

aws re/start

Task 3

Edit a file in Vim

Step 4: Check the saved changes

Reopen the file in Vim using the command vim helloword to verify the saved changes.

```
[ec2-user@ip-10-0-10-156 ~]$ vim helloworld
[ec2-user@ip-10-0-10-156 ~]$ vim helloworld
```

```
Hello World!
This is my first file in Linux and I am aditing it in Vim!
~
~
"helloworld" 2L, 72B                          1,1           All
```

Step 5: Add more lines

Press i to enter Vim Insert Mode again and add another line to the editor. Then, press ESC to exit Insert Mode.

```
Hello World!
This is my first file in Linux and I am aditing it in Vim!
I learned how to create a file, edit and save them too!
~
-- INSERT --                                  3,56          All
```

Step 6: Quit without saving

In the Command Mode, enter the command :q! to exit the editor without saving changes.

```
Hello World!
This is my first file in Linux and I am aditing it in Vim!
I learned how to create a file, edit and save them too!
~
:q!
```

Task 3

Edit a file in Vim

Step 7: Check the unsaved changes

Reopen the file in Vim using the command vim helloword to verify the unsaved changes.

```
[ec2-user@ip-10-0-10-156 ~]$ vim helloworld
[ec2-user@ip-10-0-10-156 ~]$ vim helloworld
[ec2-user@ip-10-0-10-156 ~]$ vim helloworld
```

```
Hello World!
This is my first file in Linux and I am aditing it in Vim!
~
~
"helloworld" 2L, 72B                          1,1           All
```

Additional challenge

Try additional useful commands.

- The dd command to delete the entire line.
- The u command to undo the last command.
- The :w command to save changes without quitting.

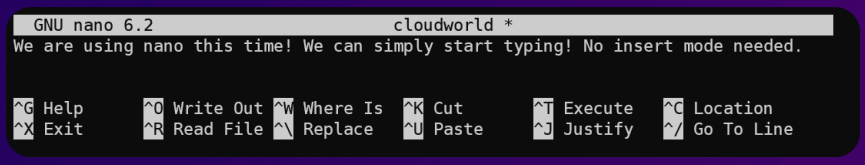
Task 4

Edit a file in nano

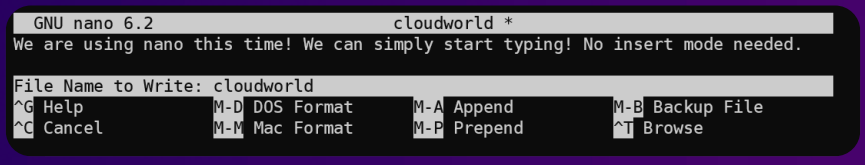
Use nano to create and edit a file called **cloudworld**.

```
[ec2-user@ip-10-0-10-156 ~]$ nano cloudworld
```

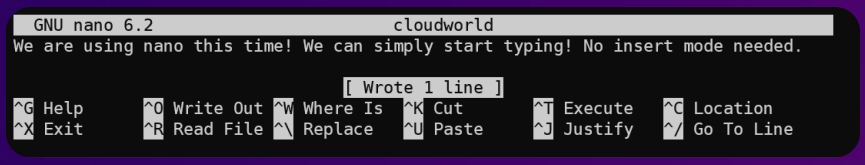
Unlike vim, in the nano editor you can start typing right way.



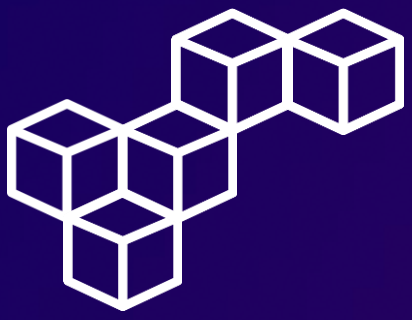
Press ^O to save your changes to the file. Then, press Enter to confirm the file name once you save it.



Now that you have saved the file, press ^X to exit the editor.



aws re/start



Conclusions

Text editors

Text editors like Vim and nano play pivotal roles in the Linux ecosystem, offering efficient and customizable tools for editing files directly from the command line.

Vim (Vi IMproved)

Vim stands out for its modal editing system, which allows users to navigate, edit, and manipulate text with speed and precision once they master its commands and shortcuts.

GNU nano

nano provides a user-friendly interface and simpler commands, making it an accessible choice for beginners or those who prefer a straightforward text editing experience in Linux.

aws re/start

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru