

5°

Lab - AWS re/Start

Edición de archivos





Tutorial de Vim

A continuación, se tratarán los siguientes temas:

- Utilizar el archivo ejecutable vimtutor para llevar a cabo las tareas 1 a 4
- Copiar el contenido del archivo `/var/log/secure` y editarlo con nano

Notar que Vim es un editor de texto de Linux

```
ec2-user@ip-10-0-10-189:~  
===== Welcome to the VIM Tutor - 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.  
  
The approximate time required to complete the tutor is 30 minutes,  
depending upon how much time is spent with experimentation.  
  
ATTENTION:  
The commands in the lessons will modify the text. Make a copy of this  
file to practice on (if you started "vimtutor" this is already a copy).  
  
It is important to remember that this tutor is set up to teach by  
use. That means that you need to execute the commands to learn them  
properly. If you only read the text, you will forget the commands!  
  
Now, make sure that your Caps-Lock key is NOT depressed and press  
the j key enough times to move the cursor so that lesson 1.1  
completely fills the screen.  
"/tmp/tutorJiLGbv" 972 lines, 33583 bytes
```

Tarea 01



Movimiento del cursor:

j (abajo), k (arriba), h (izquierda), l (derecha)

```
** To move the cursor, press the h,j,k,l keys as indicated. **  
      ^  
      k  
    < h      l >      Hint:  The h key is at the left and moves left.  
      j              The l key is at the right and moves right.  
      v              The j key looks like a down arrow.  
1. Move the cursor around the screen until you are comfortable.
```

Para salir del tutorial del vim, usamos `:q!` y para iniciar de nuevo el tutorial usamos el comando `vimtutor`. Y para borrar un caracter donde se encuentra el cursor es con la tecla x. Para insertar texto, usamos `i` o `a` y empezamos a agregar los caracteres deseados

```
** Press i to insert text. **  
  
1. Move the cursor to the first line below marked --->.  
  
2. To make the first line the same as the second, move the cursor on top  
   of the character BEFORE which the text is to be inserted.  
  
3. Press i and type in the necessary additions.  
  
4. As each error is fixed press <ESC> to return to Normal mode.  
   Repeat steps 2 through 4 to correct the sentence.  
  
---> There is text missing this .  
---> There is some text missing from this line.  
  
5. When you are comfortable inserting text move to lesson 1.5.
```

Tarea 01



Usamos `:wq` con el fin de guardar las que hemos hecho en el archivo y salir.

```
                ** Use :wq to save a file and exit. **

!! NOTE: Before executing any of the steps below, read this entire lesson!!

1.  If you have access to another terminal, do the following there.
    Otherwise, exit this tutor as you did in lesson 1.2:  :q!

2.  At the shell prompt type this command:  vim file.txt <ENTER>
    'vim' is the command to start the Vim editor, 'file.txt' is the name of
    the file you wish to edit.  Use the name of a file that you can change.

3.  Insert and delete text as you learned in the previous lessons.

4.  Save the file with changes and exit Vim with:  :wq <ENTER>
```

Resumen de la primera lección del tutorial

```
~~~~~
                        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.
```

Tarea 01



Resumen de la lección 2 del tutorial

```
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
```

Resumen de la lección 3 del tutorial

```
~~~~~
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

Now go on to the next lesson.
```

Tarea 01



Resumen de la lección 4 del tutorial

```
~~~~~
Lesson 4 SUMMARY

1. CTRL-G displays your location in the file and the file status.
   G moves to the end of the file.
   number G moves to that line number.
   gg moves to the first line.

2. Typing / followed by a phrase searches FORWARD for the phrase.
   Typing ? followed by a phrase searches BACKWARD for the phrase.
   After a search type n to find the next occurrence in the same direction
   or N to search in the opposite direction.
   CTRL-O takes you back to older positions, CTRL-I to newer positions.

3. Typing % while the cursor is on a (,),[,],{, or } goes to its match.

4. To substitute new for the first old in a line type      :s/old/new
   To substitute new for all 'old's on a line type         :s/old/new/g
   To substitute phrases between two line #'s type        :#,#s/old/new/g
   To substitute all occurrences in the file type          :%s/old/new/g
   To ask for confirmation each time add 'c'              :%s/old/new/gc
~~~~~
```

Resumen de la lección 5 del tutorial

```
~~~~~
Lesson 5 SUMMARY

1. :!command executes an external command.

   Some useful examples are:
   (Windows)      (Unix)
   :!dir           :!ls           - shows a directory listing.
   :!del FILENAME :!rm FILENAME  - removes file FILENAME.

2. :w FILENAME writes the current Vim file to disk with name FILENAME.

3. v motion :w FILENAME saves the Visually selected lines in file
   FILENAME.

4. :r FILENAME retrieves disk file FILENAME and puts it below the
   cursor position.

5. :r !dir reads the output of the dir command and puts it below the
   cursor position.
~~~~~
```


Tarea 01



Resumen de la lección 6 del tutorial

```
~~~~~
Lesson 6 SUMMARY

1. Type o to open a line BELOW the cursor and start Insert mode.
   Type O to open a line ABOVE the cursor.

2. Type a to insert text AFTER the cursor.
   Type A to insert text after the end of the line.

3. The e command moves to the end of a word.

4. The y operator yanks (copies) text, p puts (pastes) it.

5. Typing a capital R enters Replace mode until <ESC> is pressed.

6. Typing ":set xxx" sets the option "xxx". Some options are:
   'ic' 'ignorecase'      ignore upper/lower case when searching
   'is' 'incsearch'      show partial matches for a search phrase
   'hls' 'hlsearch'      highlight all matching phrases
   You can either use the long or the short option name.

7. Prepend "no" to switch an option off:  :set noic
```

Resumen de la lección 7 del tutorial

```
~~~~~
Lesson 7 SUMMARY

. Type :help or press <F1> or <HELP> to open a help window.

. Type :help cmd to find help on cmd .

. Type CTRL-W CTRL-W to jump to another window.

. Type :q to close the help window.

. Create a vimrc startup script to keep your preferred settings.

. When typing a : command, press CTRL-D to see possible completions.
  Press <TAB> to use one completion.
```

Tarea 02



Ahora procedemos a editar un archivo usando el comando *vim*. En este caso en un archivo llamado helloworld creamos el siguiente texto, con *dd* se eliminó la línea, con *u* es algo similar a CTRL + Z (anula el efecto del último comando), y *:w* para simplemente guardar, pero sin salir del archivo

A screenshot of a terminal window titled "ec2-user@ip-10-0-10-106:~". The terminal shows the output of a command sequence: "Hello World!", "This is my first file in Linux and I a editing it in Vim!", and "I learned how to create a file, edit and save them too!". Below the text, there are several tilde (~) symbols representing blank lines. At the bottom of the terminal, a status bar displays '"hellorworld" 3L, 127B written' on the left, '3,1' in the center, and 'All' on the right. The terminal has a black background with white text.

Tarea 03



Ahora procedemos a editar un archivo usando el comando *nano*. En este caso en un archivo llamado *cloudworld* creamos el siguiente texto, donde no necesitamos presionar la tecla *i* para insertar texto. Y se guarda utilizando *CTRL+O+ENTER* y salimos del editor con *CTRL+X*

```
ec2-user@ip-10-0-10-106:~
GNU nano 2.9.8 cloudworld Modified
We are using nano this time! We can simply start typing! No insert mode needed.
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```