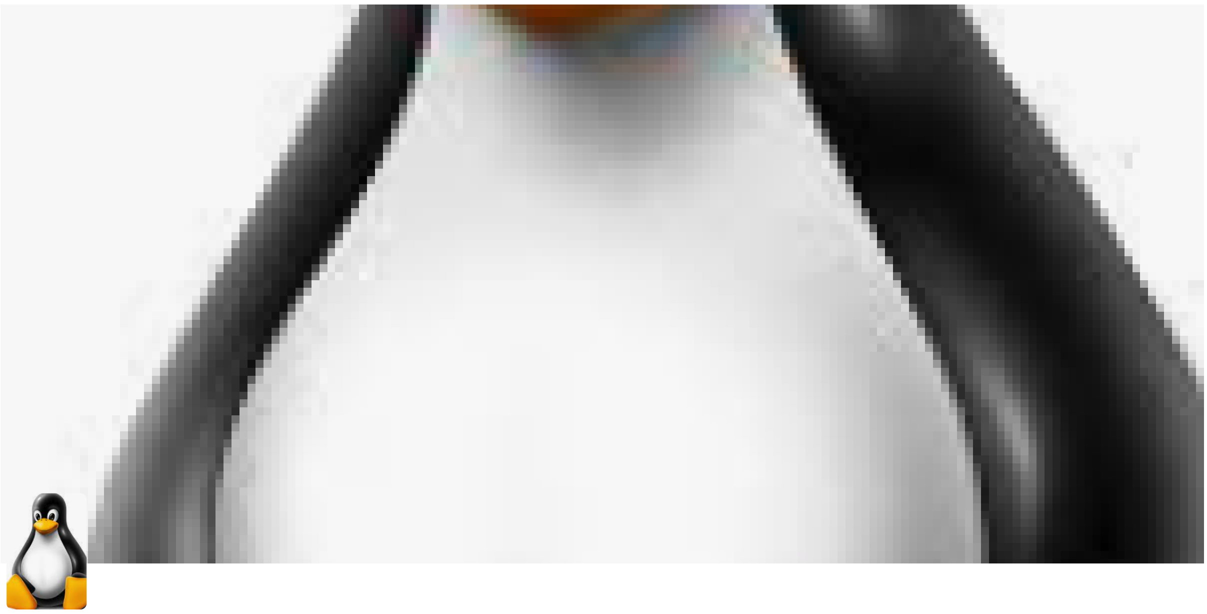


Linux Commands

(https://github.com/fatihtepe)

echo "Today is $(date)"
Today is Thu 13 May 2021 11:15:21 EDT

echo "Path is $PATH"
Path is /opt/homebrew/bin:/op............

ping host-name/IP

SSH

- `ssh` stands for “Secure Shell”.
- It is a protocol used to securely connect to a remote server/system.
- `ssh` is secure in the sense that it transfers the data in an encrypted form between the host and the client

```
ssh user@host(IP/Domain_name)
```

> Example

```
ssh -i cert.pem ec2-user@54.93.34.220
```

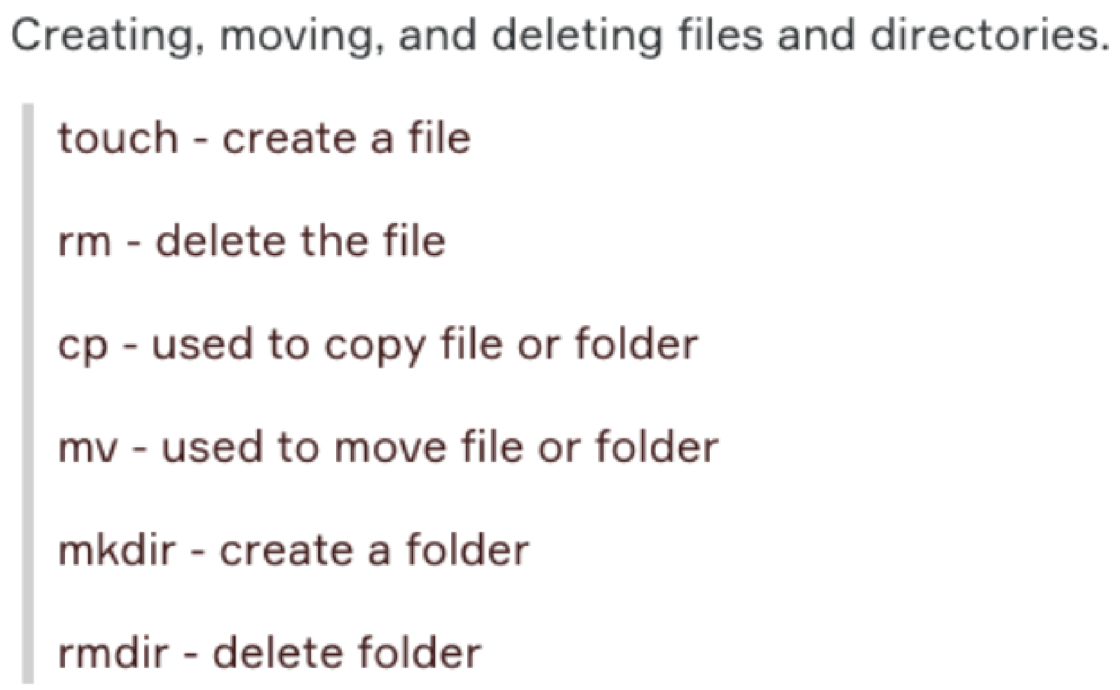
The info page for a particular command is invoked by preceding the command with `info`.

> info <command>

info echo

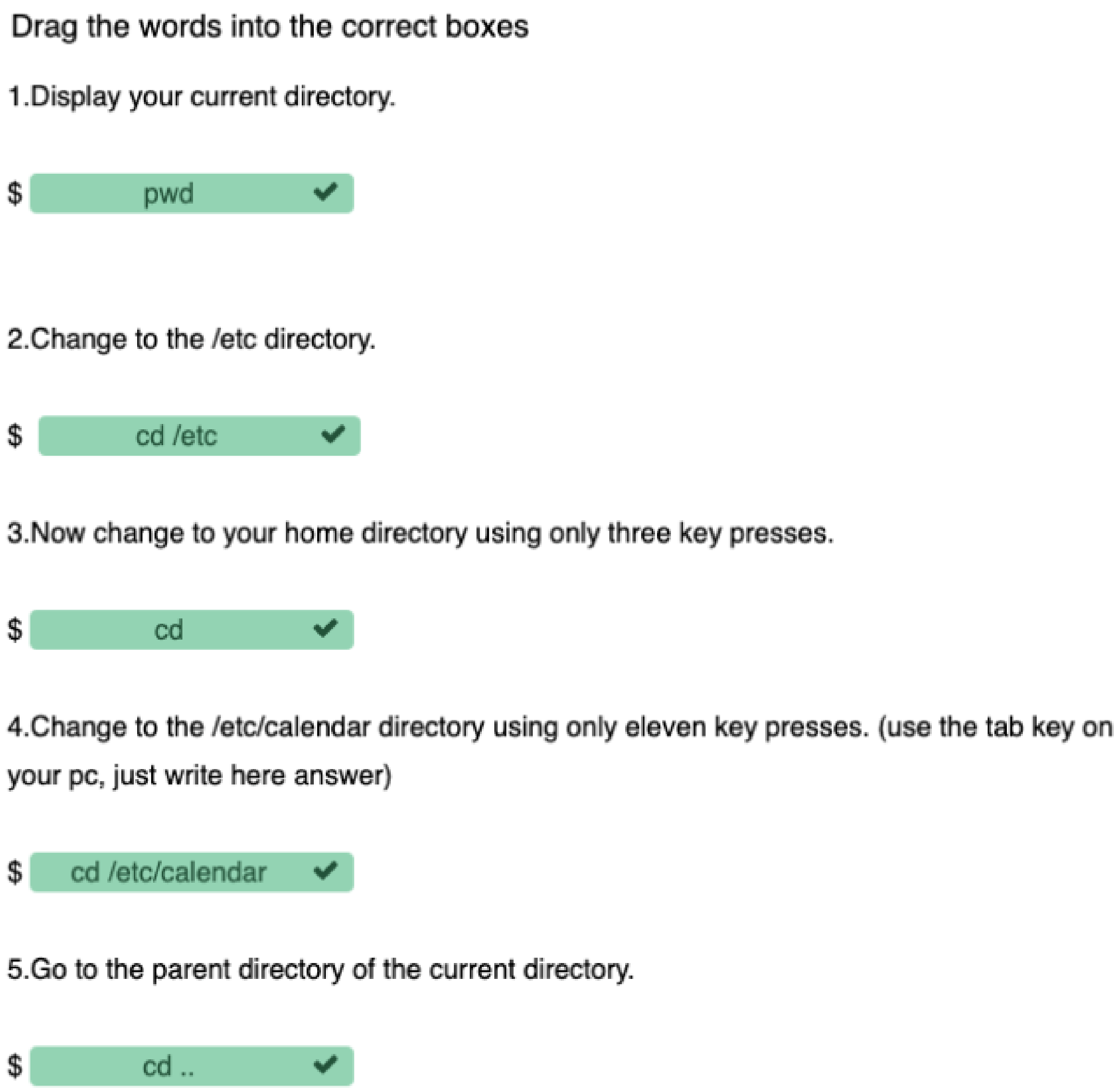
man ls

The main difference between Man and Info is the amount of content that they have; Info contains a whole lot more than Man does.



Creating, moving, and deleting files and directories.

> touch - create a file
>
> rm - delete the file
>
> cp - used to copy file or folder
>
> mv - used to move file or folder
>
> mkdir - create a folder
>
> rmdir - delete folder



Drag the words into the correct boxes

1.Display your current directory.

$ pwd ✓

2.Change to the /etc directory.

$ cd /etc ✓

3.Now change to your home directory using only three key presses.

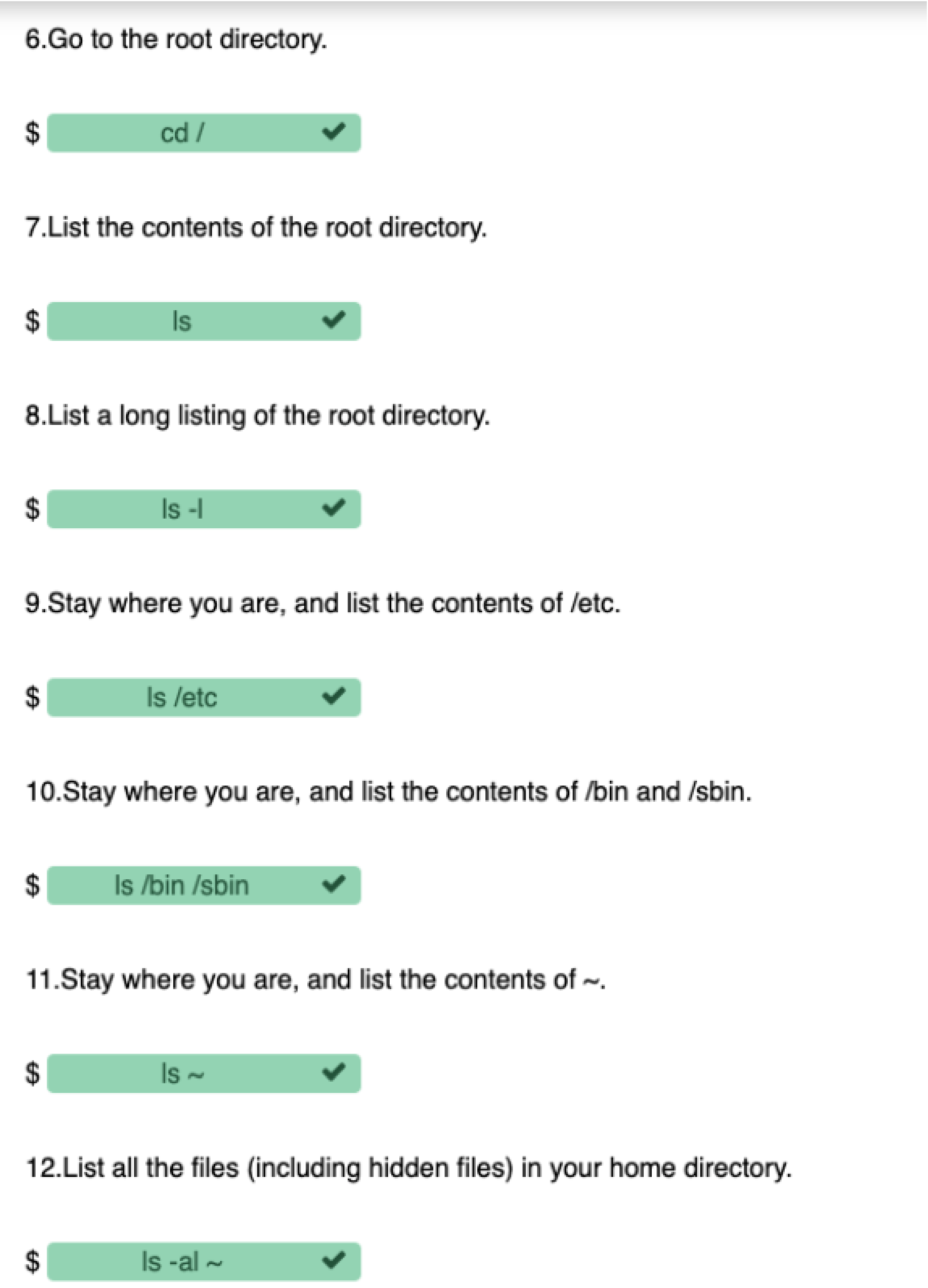
$ cd ✓

4.Change to the /etc/calendar directory using only eleven key presses. (use the tab key on your pc, just write here answer)

$ cd /etc/calendar ✓

5.Go to the parent directory of the current directory.

$ cd .. ✓



6.Go to the root directory.

```
$ cd / ✓
```

7.List the contents of the root directory.

```
$ ls ✓
```

8.List a long listing of the root directory.

```
$ ls -l ✓
```

9.Stay where you are, and list the contents of /etc.

```
$ ls /etc ✓
```

10.Stay where you are, and list the contents of /bin and /sbin.

```
$ ls /bin /sbin ✓
```

11.Stay where you are, and list the contents of ~.

```
$ ls ~ ✓
```

12.List all the files (including hidden files) in your home directory.

```
$ ls -al ~ ✓
```

13.List the files in /boot in a human readable format.

$ `ls -lh /boot` ✓

14.Create a directory testdir in your home directory.

$ `mkdir ~/testdir` ✓

15.Change to the /etc directory, stay here and create a directory newdir in your home directory.

$ `cd /etc` ✓

$ `mkdir ~/newdir` ✓

16.Create in one command the directories ~/dir1/dir2/dir3 (dir3 is a subdirectory from dir2, and dir2 is a subdirectory from dir1).

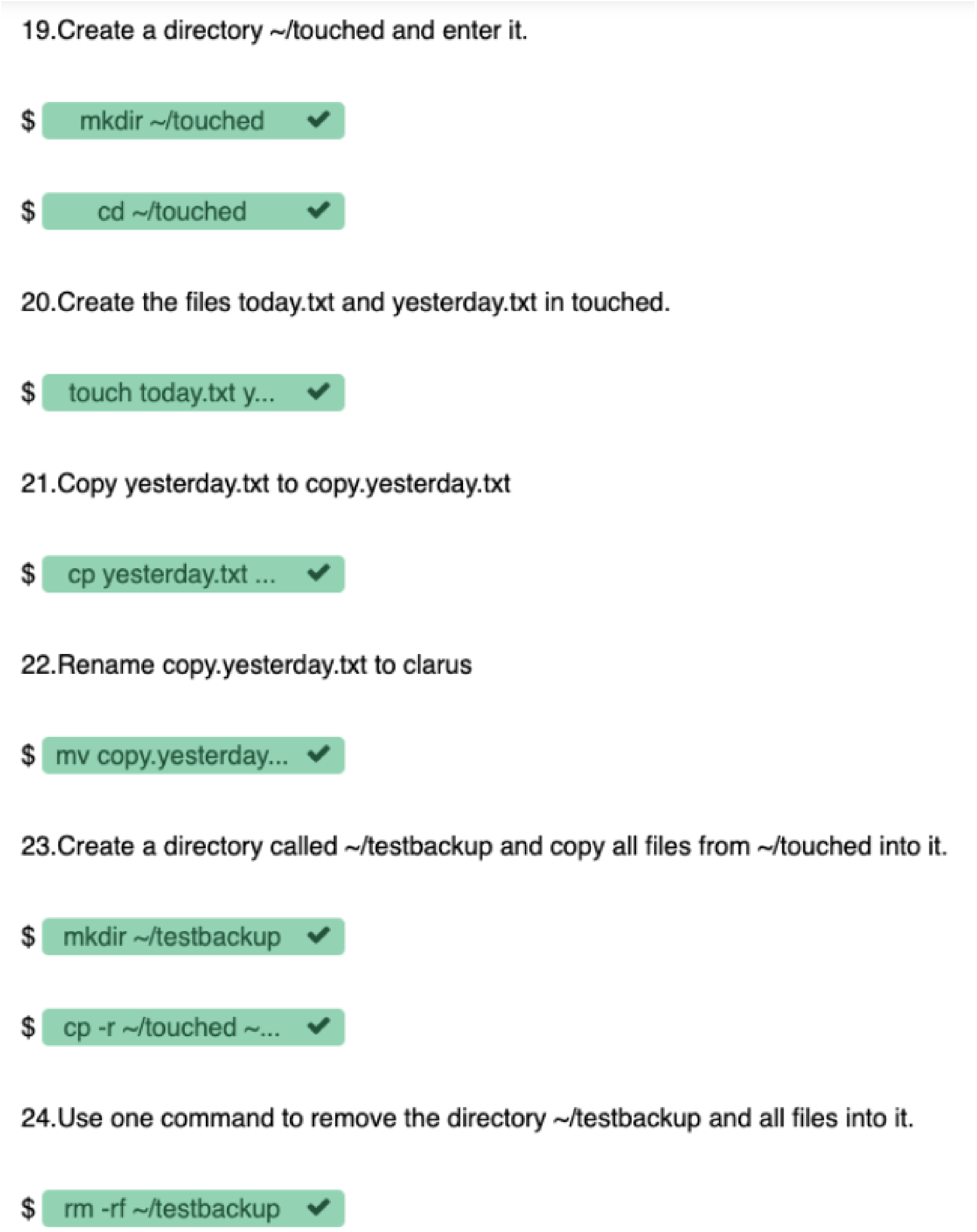
$ `mkdir -p ~/dir1/d...` ✓

17.Remove the directory testdir.

$ `rmdir testdir` ✓

18.List the files in the /bin directory

$ `ls /bin` ✓



19.Create a directory ~/touched and enter it.

```
$ mkdir ~/touched ✓
```

```
$ cd ~/touched ✓
```

20.Create the files today.txt and yesterday.txt in touched.

```
$ touch today.txt y... ✓
```

21.Copy yesterday.txt to copy.yesterday.txt

```
$ cp yesterday.txt ... ✓
```

22.Rename copy.yesterday.txt to clarus

```
$ mv copy.yesterday... ✓
```

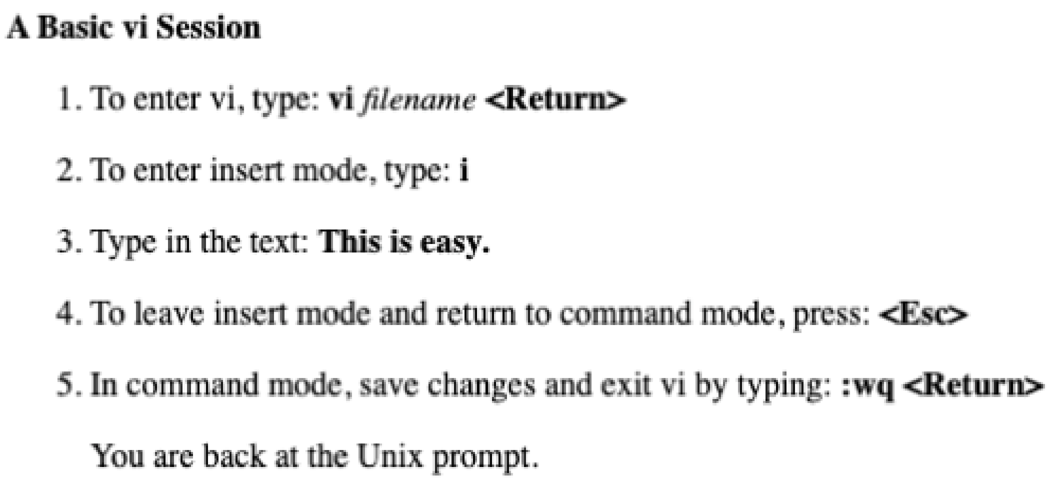
23.Create a directory called ~/testbackup and copy all files from ~/touched into it.

```
$ mkdir ~/testbackup ✓
```

```
$ cp -r ~/touched ~... ✓
```

24.Use one command to remove the directory ~/testbackup and all files into it.

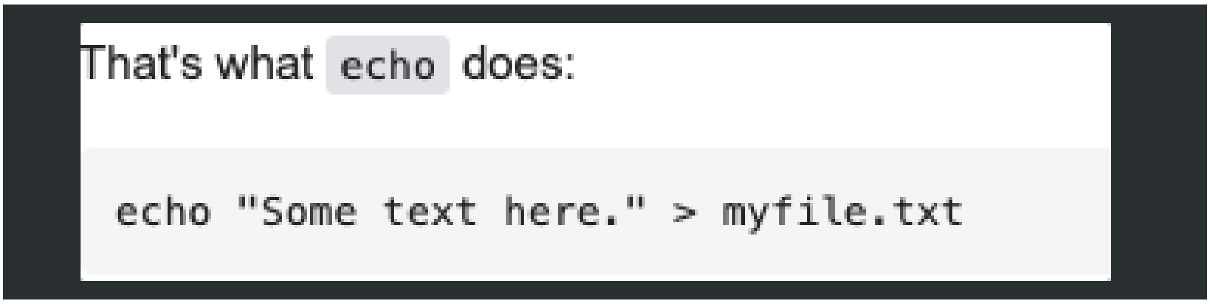
```
$ rm -rf ~/testbackup ✓
```



A Basic vi Session

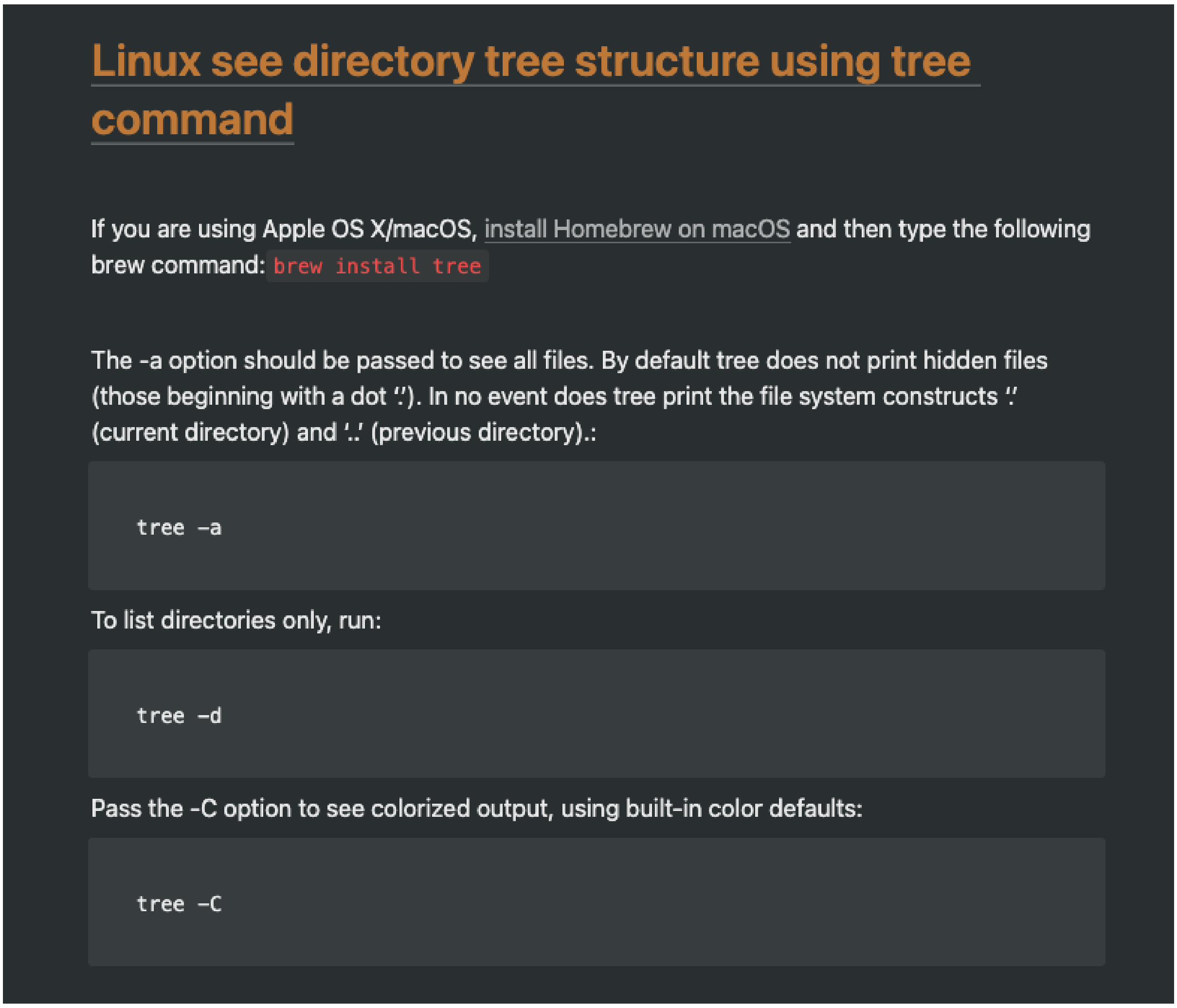
1. To enter vi, type: **vi** *filename* **<Return>**
2. To enter insert mode, type: **i**
3. Type in the text: **This is easy.**
4. To leave insert mode and return to command mode, press: **<Esc>**
5. In command mode, save changes and exit vi by typing: **:wq <Return>**

   You are back at the Unix prompt.



That's what `echo` does:

```
echo "Some text here." > myfile.txt
```



Linux see directory tree structure using tree command

If you are using Apple OS X/macOS, install Homebrew on macOS and then type the following brew command: `brew install tree`

The -a option should be passed to see all files. By default tree does not print hidden files (those beginning with a dot '.'). In no event does tree print the file system constructs '.' (current directory) and '..' (previous directory).:

```
tree -a
```

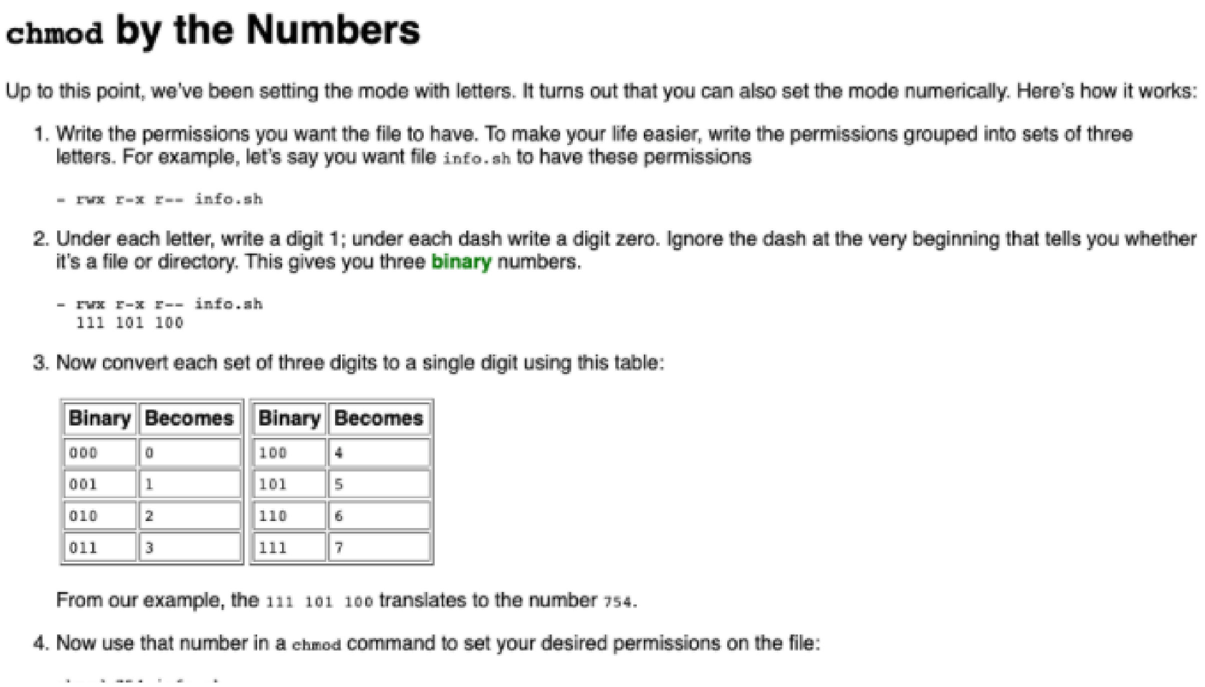
To list directories only, run:

```
tree -d
```

Pass the -C option to see colorized output, using built-in color defaults:

```
tree -C
```

Bash = Bourne Again Shell



chmod by the Numbers

Up to this point, we've been setting the mode with letters. It turns out that you can also set the mode numerically. Here's how it works:

1. Write the permissions you want the file to have. To make your life easier, write the permissions grouped into sets of three letters. For example, let's say you want file `info.sh` to have these permissions

   ```
   - rwx r-x r-- info.sh
   ```

2. Under each letter, write a digit 1; under each dash write a digit zero. Ignore the dash at the very beginning that tells you whether it's a file or directory. This gives you three **binary** numbers.

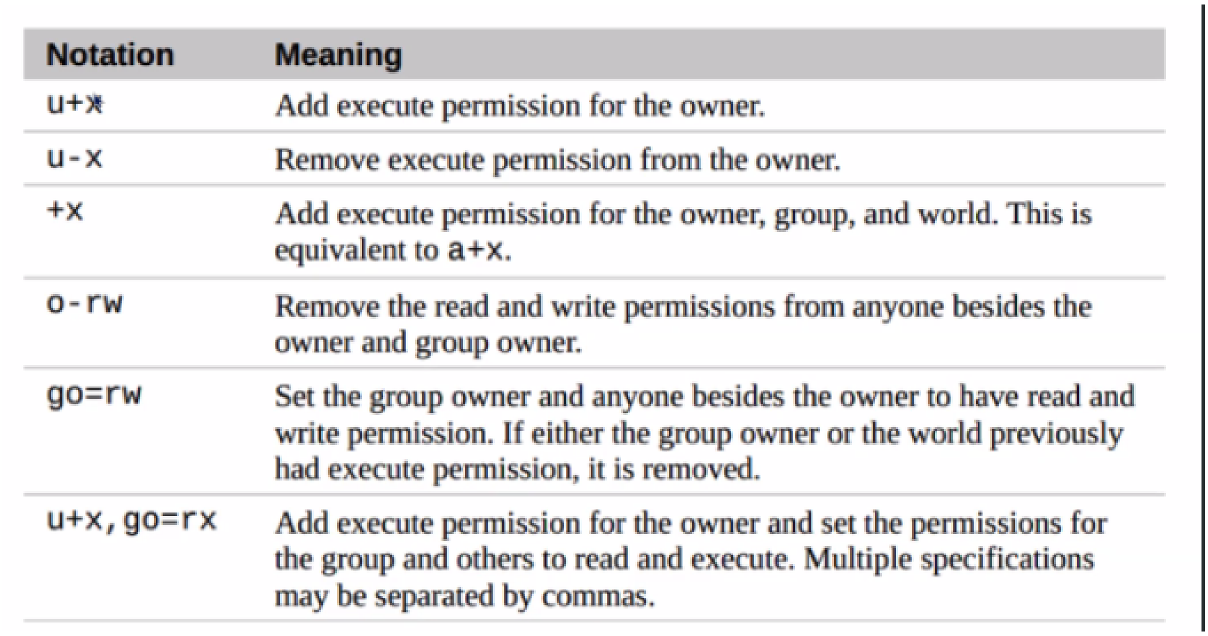
   ```
   - rwx r-x r-- info.sh
     111 101 100
   ```

3. Now convert each set of three digits to a single digit using this table:

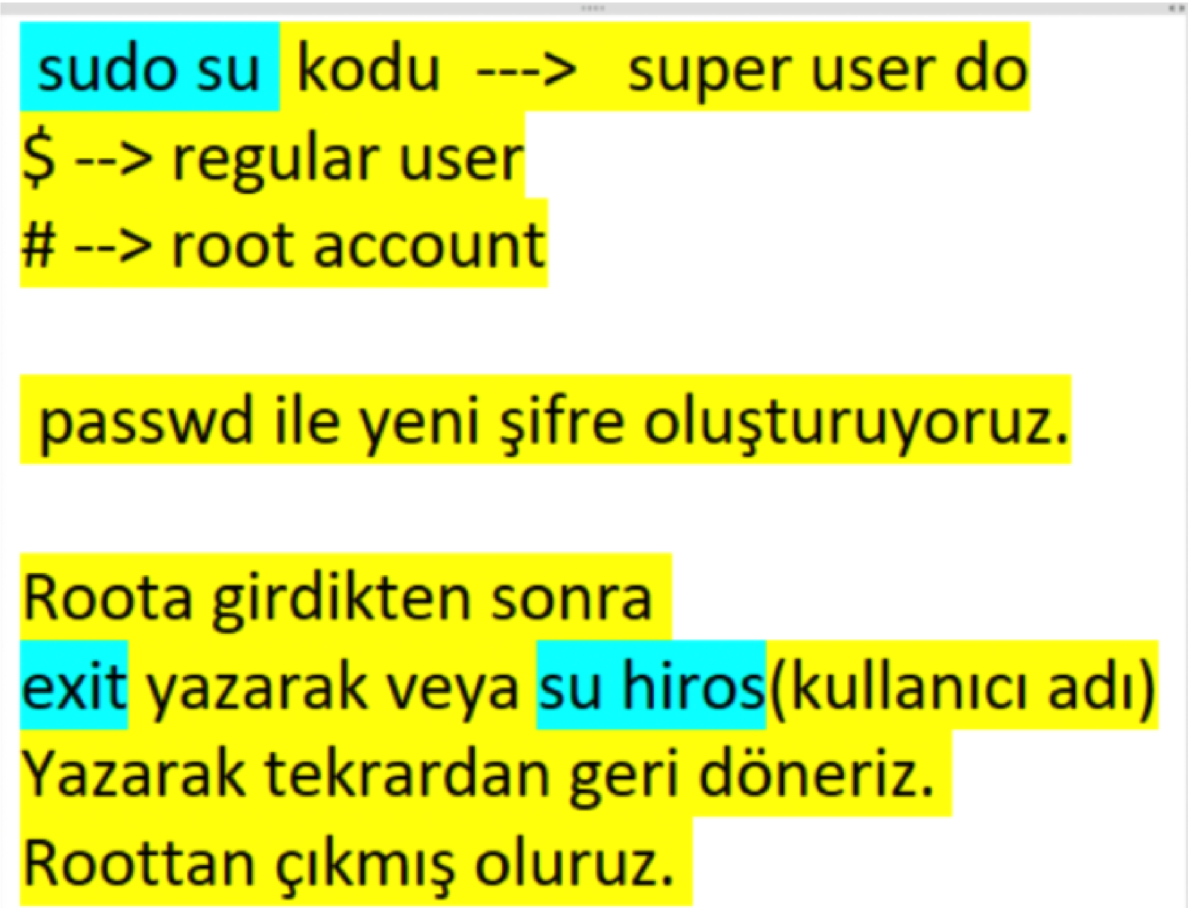
Binary	Becomes	Binary	Becomes
000	0	100	4
001	1	101	5
010	2	110	6
011	3	111	7

   From our example, the `111 101 100` translates to the number `754`.

4. Now use that number in a `chmod` command to set your desired permissions on the file:



Notation	Meaning
`u+x`	Add execute permission for the owner.
`u-x`	Remove execute permission from the owner.
`+x`	Add execute permission for the owner, group, and world. This is equivalent to `a+x`.
`o-rw`	Remove the read and write permissions from anyone besides the owner and group owner.
`go=rw`	Set the group owner and anyone besides the owner to have read and write permission. If either the group owner or the world previously had execute permission, it is removed.
`u+x,go=rx`	Add execute permission for the owner and set the permissions for the group and others to read and execute. Multiple specifications may be separated by commas.



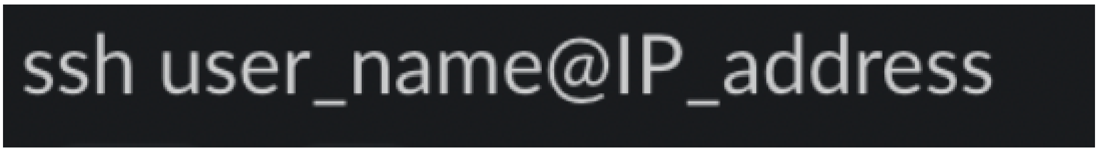
sudo su kodu ---> super user do

$ --> regular user

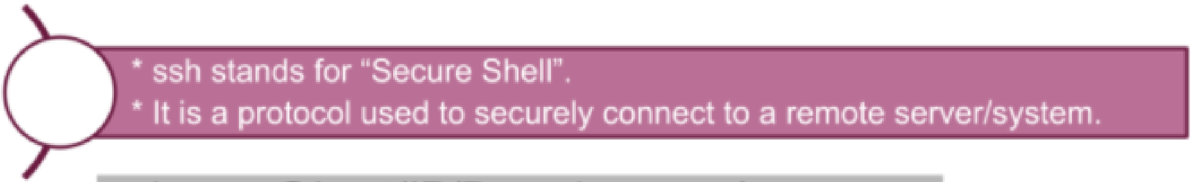
--> root account

passwd ile yeni şifre oluşturuyoruz.

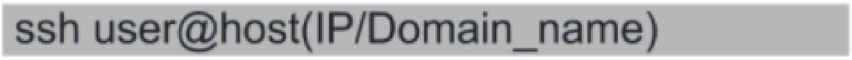
Roota girdikten sonra
exit yazarak veya su hiros(kullanıcı adı)
Yazarak tekrardan geri döneriz.
Roottan çıkmış oluruz.



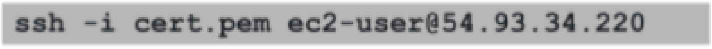
```
ssh user_name@IP_address
```



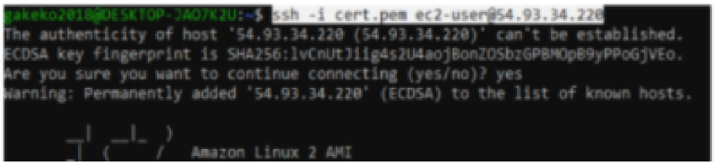
* ssh stands for “Secure Shell”.
* It is a protocol used to securely connect to a remote server/system.



```
ssh user@host(IP/Domain_name)
```



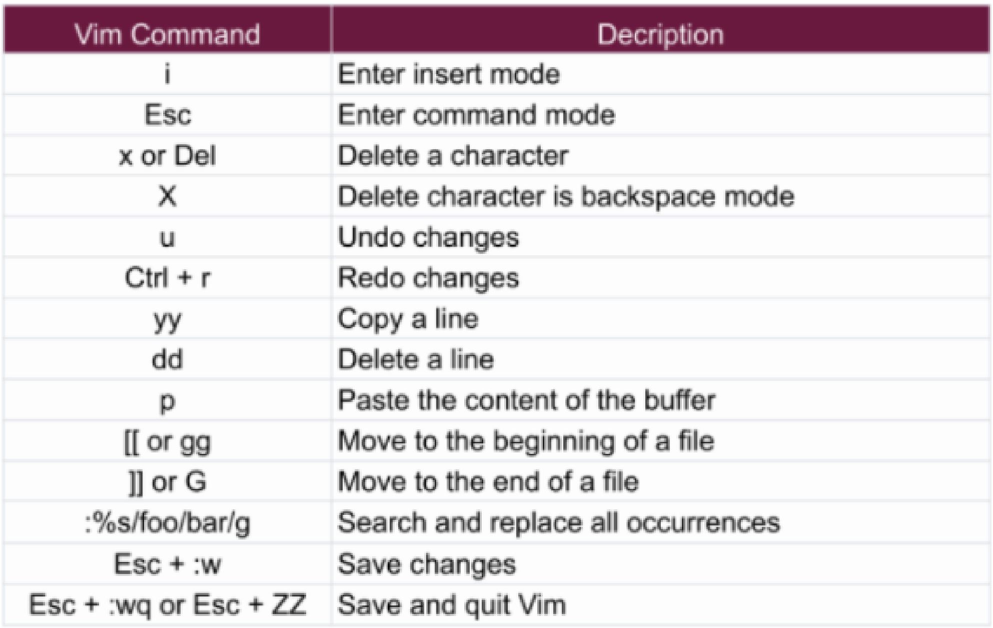
```
ssh -i cert.pem ec2-user@54.93.34.220
```



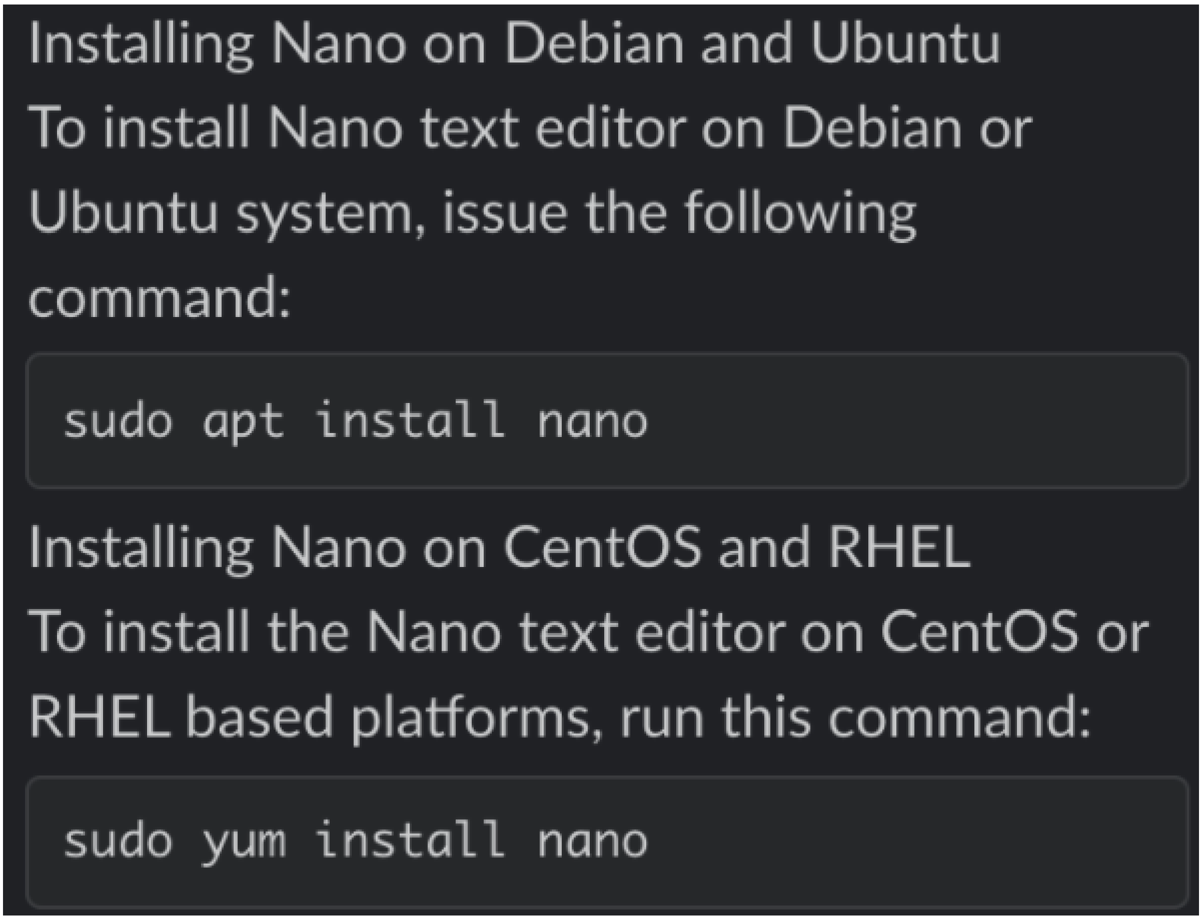
```
gakeko2018@DESKTOP-JAO7K2U:~$ ssh -i cert.pem ec2-user@54.93.34.220
The authenticity of host '54.93.34.220 (54.93.34.220)' can't be established.
ECDSA key fingerprint is SHA256:lvCnUtJiig4s2U4aojBonZOSbzGPBMOpB9yPPoGjVEo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.93.34.220' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
```

Vim Editor



Vim Command	Decription
i	Enter insert mode
Esc	Enter command mode
x or Del	Delete a character
X	Delete character is backspace mode
u	Undo changes
Ctrl + r	Redo changes
yy	Copy a line
dd	Delete a line
p	Paste the content of the buffer
[[or gg	Move to the beginning of a file
]] or G	Move to the end of a file
:%s/foo/bar/g	Search and replace all occurrences
Esc + :w	Save changes
Esc + :wq or Esc + ZZ	Save and quit Vim



Installing Nano on Debian and Ubuntu

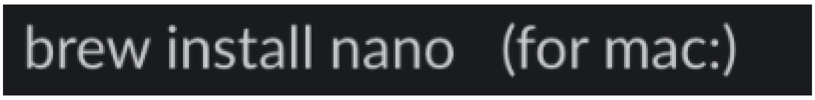
To install Nano text editor on Debian or Ubuntu system, issue the following command:

```
sudo apt install nano
```

Installing Nano on CentOS and RHEL

To install the Nano text editor on CentOS or RHEL based platforms, run this command:

```
sudo yum install nano
```



brew install nano (for mac:)