In RedHat family OS, every command you execute is going to create a process in the system. Also that process is going to get one unique number allocated to it which is called as **"PID"** (Process ID)

Those process can be fetched from the command line using ps command.

ps

But the above command will show only the process related to this session which you logged in.

If you need all the process which belongs to the current user then you can use -u option along with it.

ps -u

An operating system also have some other users and as well as system process running in the background.

If you want to see every process related to the system and process belongs to other users then we have to use -e option.

ps -e

Option -e show all process along with PID and PNAME, But in case of more information then you can use -f option along with the -e option.

ps -ef

Ofcourse the above command shows all the process which we might look some process all the time, So you can filter the process by piping the output to grep command

ps -ef | grep ssh

As an administrator we might need to manage the process in the system. Some times we terminate process normally and sime times we might do it forcefully.

To manage the process or to be precise that to kill a process we use kill command.

Let us create a process in the background.

sleep 300 &

In above command sleep 300 is going to run the process for 300 seconds, Where as & symbol is pushing the process to run in the background. So it will be useful to get the terminal immediate and as well as such process will run in the background even if we close the session.

Ok, let us find out the process ID for sleep command.

ps -ef | grep sleep

Now we want to kill this process, So we would use kill command along with the PID.

kill <PID-OF-SLEEP-COMMAND>

Now you will not find the process anymore with ps command.

ps -ef | grep sleep

Sometimes due to some reason these process may not get killed with this command and hence we may need to find a hard way / force way to kill them, Lets see a scenario..

curl -s https://raw.githubusercontent.com/devopstrainings/linux-basics-katakoda/master/linux-adminstration/files/no-kill.sh -o /tmp/no-kill.sh

sh /tmp/no-kill.sh &

Now you can fetch the PID and try to kill the process, But you will not be able to.

ps -ef | grep sleep

kill <PID-OF-SLEEP-COMMAND>

So, we might need to kill such process using an force option -9 .

kill -9 <PID-OF-SLEEP-COMMAND>

Additional Commands for Practice

1. Process Priority - How to set some priority to a process.
2. Explore top command.

Self Exploring tasks

1. What happens to the childs if you kill a parent process.
2. What is an Orphan process / Zombie Process inside system.

Most of the times we run our applications as a normal user, So we should be having some knowledge on how to add a user.

In RedHat family OS, You cannot add a user without having a group.

So first lets add a group before adding a user.

To add a group in CentOS, We can use groupadd command. To that command we need to provide groupname as an argument.

groupadd devops

In above command groupadd is the command and where as devops is the group name.

Now the above command will not show any output saying that group has been created, So we need to verify that atleast that group has been created or not. To verify that a group has been created or not can be verified by checking a system file and its content.

cat /etc/group

Or you can also pipe the output to grep command to apply filter of out search as well.

cat /etc/group | grep devops

The above command showed the group has been created and also it has a number associated to it which is **GID** called as Group ID, which is a unique number associated to a group. We can also customize those numbers while creating the group or even after creating the group.

Now the group is ready and we can create a User now.

To add a user, Command is useradd .

useradd -g devops john

In the above command useradd is the command to add user, -g devops is an option instructing the useradd command to add a user in devops group and john is the username.

Alike groupadd command useradd also will not show any output on the screen showing that user has been created. So to check whether the user has been created or not, You can run the following command.

id john

The above command shows a number associated to the user which is **UID** (User ID) and which is unique number and it is allocated by system. But the same number also can be provided or managed by our own while creating the user or even after creating the user.

Generally the user primarily belongs to one group in system and at the same time he can be part of multiple groups as well. But those groups are called as supplementary groups.

To add user to multiple groups..

groupadd admins

usermod -a -G admins john

You can check all his groups information using the same id command.

id john

If this user needs to connect to the system then he needs a password, So let us assign a password to the user.

To set / reset a password to the user we use passwd command. ROOT user can set / reset the password of any user, whereas an individual user can only change his password by using the same command. passwd command prompts for entering the password and give an input and it sets as password to that user.

passwd john

Now, You can connect to the system using from another terminal and check whether the user can be able to login to the session or not.

You can run the following command to check the connection,

echo ssh john@$(curl -s ifconfig.co)

This user which we created so far is a real user, But also to run our software or applications inside the server we need those process associated to a user. In other words we don't run softwares or applications with root user directly and hence we create certain users in the system and it OS terminology we called them as Daemon Users . This helps in organizing the operations and as well as improvising the security.

The following command will run some services inside the system and you can check those process and we will find those are been ran by a certain user but not by a root user.

curl -s https://raw.githubusercontent.com/devopstrainings/linux-basics-katakoda/master/linux-adminstration/files/daemon-services.sh | bash

You can check the process of some system services runs as a normal user but not as a root user.

ps -ef | grep httpd

ps -ef | grep tomcat

ps -ef | grep mariadb

Even though those services are started by a ROOT user, Yet the service runs associated by a normal user but not a ROOT User.

----

Being a root user you can switch from one user to another user without any password. At the same time a normal user also can switch from one user to another but with a password.

Either of the way if you want to switch then use a command su

su - john

Additional Tasks for Student Practice

1. Create a group with custom *GID*
2. Create a user with custom *UID*
3. After creating user try to change the *GID*
4. Try to remove a supplementary group to a user

In companies even though we know the ROOT user password, We might not login directly with ROOT user. Usually that admin ROOT user login is used only during the maintenances of the system or in some extreme cases. Such practices will lead to an issue where we could not able to track of whom is performing which activities on the server.

So we usually create users in the system and those users login to the system to perform or do some work. But a normal user cannot do the admin activities and also as a standard practice we cannot login with ROOT user. Hence we need an alternate option to login as a normal user but perform the work as root user.

To gain the above scenario we need to use some Privilege Escalation tools and few of them are

1. SUDO
2. PowerBroker
3. Centrify... etc

You can refer this link. <https://www.sudo.ws/other.html>

Out of lot many tools **SUDO** is one of the default and widely used tool. Hence we are discussing SUDO further now..

Sudoers Configuration is available in the system, So need to customize that configuration and tell to **sudoers** that which users can perform which activities and then system allows only that users to perform those commands.

Config File : /etc/sudoers

Alternatively you can add some files under /etc/sudoers.d and sudo program will load them as well.

Sudoers configuration syntax looks like below:

user-name MACHINE=COMMANDS

So either you add the above entries directly in /etc/sudoers or in a separate file in /etc/sudoers.d/ directory.

Refer the following video for understanding the mistakes. <https://www.youtube.com/watch?v=OuKpAenxh94>

You can edit the configuration using the following command to avoid some mistakes.

visudo

ALL is a default keyword which is available in sudoers to add all the commands access to all the machines. Else we can give our own list of commands with list of machines.

Now the user can execute the commands as normal user by using sudo privilege and those commands will be executed as root user in the background.

First switch to normal user.

su - john

Then

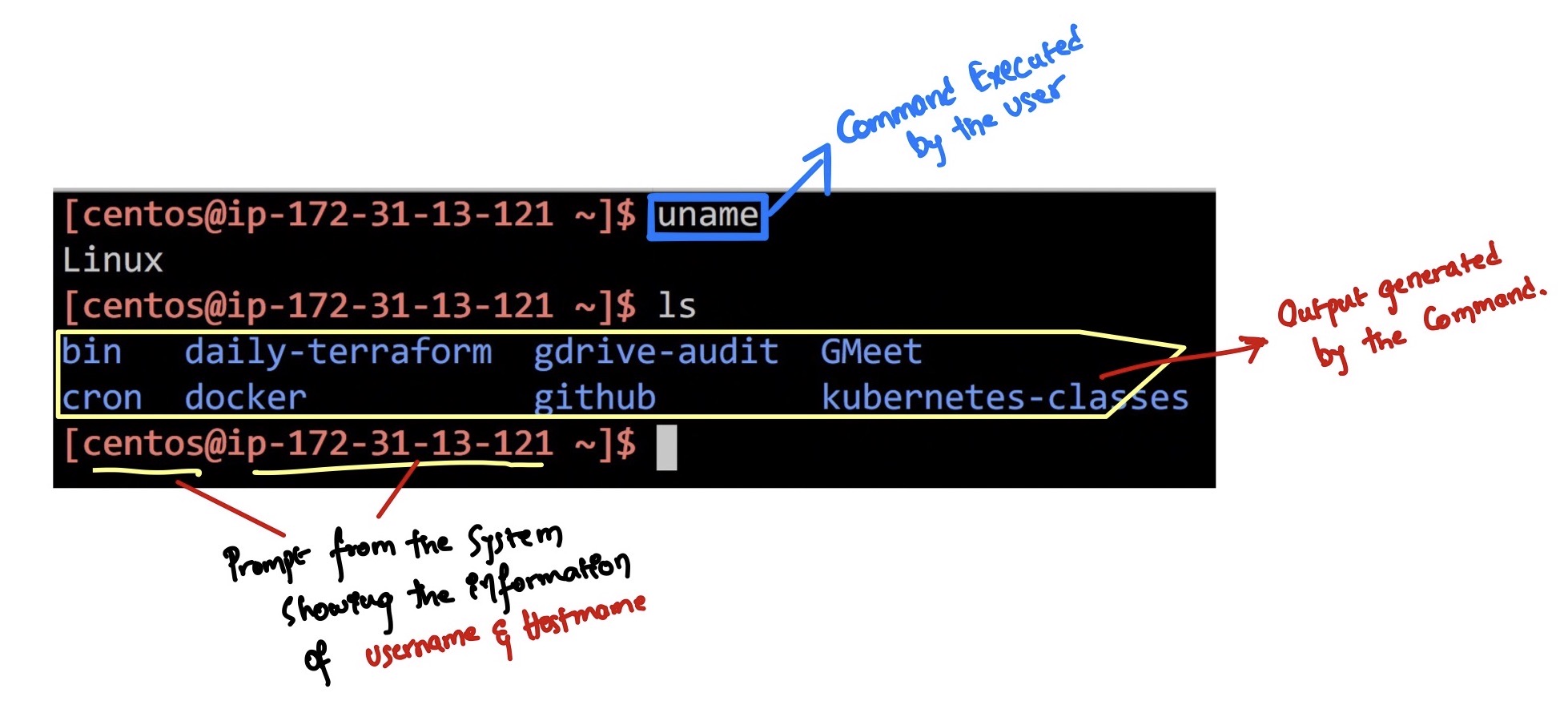
sudo useradd appuser1

Additional Tasks for Student Practice

1. Add sudo access to the group, Check whether it is working or not.

## **Command line Syntaxes.**

You can see in the picture like how commands would be in Linux.



In the above diagram-1 we are executing the commands on shell. Shell is a place where users will execute the command on server.

#### **Commands**

In terminal the first argument we give to execute is a command.

For example:

uname

uname is a command and that is the first word of a command line syntax.

#### **Options**

Certain commands are going to have options, Options in Linux Command line will be a second argument over the command line, Usually those options will be seen in three formats..

1. <command> -<single character> (Ex: -h , -v )
2. <command> --<single word> (Ex: --help , --version)
3. <command> -<single word> (Ex: -version, -help)

For Example:

uname -a

uname --all

#### **Inputs**

Certain commands require inputs, Inputs are given with options in some commands and without options for some commands.

For Example:

ls /boot

ls -d /boot

In above example ls is a command -d is an option and /boot is an input. Given the command with and without option changes the behavior of the command execution.

## **Some Quick Examples.**

In general when we purchase a new hardware like Laptops or Desktops we generally look into the configuration of the machine. So let us try to do the same for the server as well and this is minimum knowledge required while you are working on any machine either that is Desktop or Server.

So we need to login to the server to run the following commands.

To check the vendor of the operating system.

cat /etc/\*release

To check the CPU information

cat /proc/cpuinfo

To check the memory information

cat /proc/meminfo

To check the disk information

fdisk -l

or

lsblk

To check the architecture whether it is 32bit or 64bit

uname -i

32 bit -> i386/i586/i686

64 bit -> x86\_64

PS Note: Starting CentOS 7, We don't have operating systems coming in 32 bit any more. Hence, we always see 64 bit.

With this information in hand you will have an idea of what you are dealing with and the specifications of that Linux machine.

## **List Files**

In windows OS you generally see the list of files when you open a particular folder, But Linux is mostly command line and hence you may not see the files by default. Hence, you need to execute a command to check the list of files.

ls is a Linux shell command that lists directory contents of files and directories. Some practical examples of ls command are shown below.

Syntax: ls <Options> <Path>

Note that ls command works without an input i.e both the options and path are optional. It works with or without them.

ls -ld /opt

Get list of files and directories but it may not show hidden files.

ls

Get list of hidden files and directories.

ls -A

Get list of files with long format, usually shows properties of a file

ls -l

We can combine multiple options as well.

ls -Al

**NOTE:** Giving multiple options depends on the command. ls accepts multiple options but it isn't applicable for all.

# Self Exploring tasks

1. Seven types of files in Linux.

BACKNEXT

## **Creating Files**

We can create files in multiple ways/commands in Linux. On a basic we will use touch command to create the file.

Syntax: touch <filename>

touch command by default creates an empty file.

touch file.txt

To check the file created.

ls -l

In the above ls command output you can see the file is an empty size file by referring the fifth colum.

touch command can create multiple files at a single go as shown.

touch sample notes.txt lambda.py

To check the file created.

ls -l

## **Important Takeaways:**

1. In Linux OS, there is no file-extensions. Extensions are given only for user understanding.

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Removing Files**

In Linux to remove files we have rm command, We can also use unlink command which does the same thing yet rm is widely preferred and used command.

Syntax: rm <filename>

rm sample

It may ask you for a prompt (yes/no) **[Not all the times]** to remove the files. You can suppress the prompt by adding <span class='kc-markdown-code-copy'></span>-f option in the command.

rm -f lambda.py

ls

**Note:** Be careful while removing a file as it deletes all the contents of the file and retrieving the lost data is not possible in most of the cases.

## **Thing you can explore.**

1. How to remove multiple files.

rm file1 file2

## **Renaming/Moving a File**

In Linux, in order to change the name of a file we use mv command.

Syntax: mv <source-file> <destination-file>

mv notes.txt note.txt

You can check whether the file has been renamed or not by referring ls command output.

ls

**NOTE:** Unlike Windows, Linux filesystem are Case-Sensitive ones, Meaning the file note.txt & NOTE.txt can be referred as two different files. But windows FAT & NTFS filesystem are Case-Insensitive, Meaning the file note.txt & NOTE.txt are same files and you cannot create multiple files with same name.

mv note.txt NOTE.txt

You can check whether the file has been renamed or not by referring ls command output.

ls

**NOTE:** If destination exists then it will overwrite the file and in some cases it will ask you for a prompt **(yes/no)** to overwrite the file or not.

mv command intention is to move the file from one location to another yet we use mainly to rename the files as well.

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Shell CLI Shortcuts.**

Some of the reference links are below.

<https://blog.ssdnodes.com/blog/cheatsheet-bash-shortcuts/>

<https://github.com/fliptheweb/bash-shortcuts-cheat-sheet>

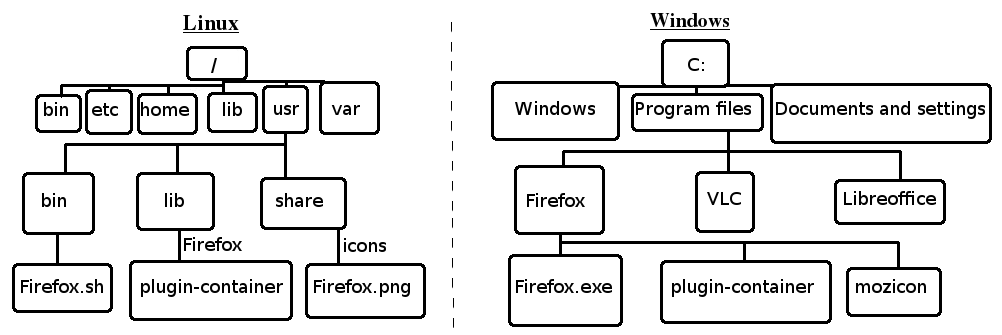
## **Shell Scripting reference guides.**

<https://devhints.io/bash>

## **Linux Directory Structure**

Alike any operating system Linux also has its own directory structure.

You can compare it with Windows in the following diagram.



In windows, we use \ (backslash) to give the path of a file or a directory but in Unix & Linux we use / (forward slash).

In Linux, we have a **ROOT DIRECTORY** where the path of any directory start here. A simple forward slash (/) is called as a **ROOT DIRECTORY**.

Additionally, to the list of directories provided in the above diagram we have more directories. Each and every default directory under / have some purpose in the operating system.

Unlike Windows, Linux is Command line based OS, So if you want to move from one directory to another directory we would be using commands to get it done.

## **Linux All Directories Explained.**

<https://www.howtogeek.com/117435/htg-explains-the-linux-directory-structure-explained/>

BACKNEXT

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Create Directories**

Creating Directory is same as creating a Folder in your windows machine. You can create a directory using mkdir command.

mkdir demo

This will create a new directory with the name demo . you can check using ls command.

ls

Now you can see demo directory listed.

But to determine demo is a directory, better always use ls -l command output. Directories start with d character of ls -l output.

mkdir -p demo/new/item1

-p option is used to create the directory recursively even if the parent directory is missing

mkdir demo1 demo2 demo3 demo4

You can also create multiple directories.

ls

## **Additional things to learn.**

There are total seven type of files in Linux. Explore each of them. Directory and regular file is two of them.

Types: Directory, Regular File, Block Special File, Character Special File, Named pipe file, Link file, Socket file.

BACKNEXT

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Copy Directories**

Copying directories can be done with the same command cp that is used to copy the files but while copying the directories we need mention -r option.

Syntax: cp -r dir1 dir2

It copies all the contents of dir1 into dir2.

**Note:** If dir2 already exists dir1 will be copied inside dir2

cp -r demo1 demo2

Copy always changes the behavior based on Target Directory.

cp SOURCE TARGET

-> if TARGET exists and if it is a file then it is invalid operation.

-> if TARGET exists and if it is a dir then it copies the file inside the directory.

-> if TARGET doesnt exist then it will copy the directory to that TATGET name

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Moving Directories**

Moving directories or renaming directories can be done using mv command.

Syntax: mv source destination

1. If destination doesn't exist it renames the directory
2. If destination exists the source will be moved into the directory

mv demo4 DEMO4

This will rename the directory demo4 as DEMO4

ls

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Removing Directories**

To remove a directory we use rmdir command in linux. Removing directories also deletes all the files that the directory holds inside it.

Syntax: rmdir <directory>

mkdir demo1

ls

rmdir demo1

ls

Check the output to see if the directory is deleted or not.

It deletes the directory with the name demo1 but in the following example you will see an error saying the directory is not empty. That is because we have already created sub-directories named new and test inside demo1 .

mkdir -p demo1/{new,test}

rmdir demo1

To delete them recursively we use -r option.

rm -r demo1

ls

Sometimes you might be prompted for (yes/no) to delete the files, and if we want to make a forceful delete without prompting then we use -f option.

mkdir -p demo1/{new,test}

ls

rm -r -f demo1

ls

**Note:** Once the files are removed there is no way of retrieving them.

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Concatenate File Content**

cat (concatenate) command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, concatenate files. So let us see some frequently used cat commands.

Syntax: cat <filename>

cat /etc/passwd

It shows the content of the file

cat -n /etc/passwd

It shows the content of the file along with the line numbers

tac /etc/passwd

It displays the content of the file in reverse order.

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

In many situations you might want to have only a certain number of lines from a file. You can use filter commands or a combination of them to get your work done.

Usually the filters are based on

1. Line Numbers
2. Row Filters
3. Column Filters

## **Head Command**

To filter the output based on line numbers and that to be from starting of the file then we use head command.

Syntax: head <filename>

By default head command gives you top 10 lines of the file but you can change it according to your needs.

head /etc/passwd

head -n 5 /etc/passwd

It gives the first 5 lines of the file

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Tail Command**

head command gives you the top lines of the file however if you want to print the last lines you can use tail command

Syntax: tail <filename>

tail command will print last 10 lines and however you can change them using -n option.

tail /etc/passwd

tail -n 5 /etc/passwd

It gives the last 5 lines of the file

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Grep command**

The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for globally search for regular expression and print out).

Syntax: grep <word> <filename>

grep root /etc/passwd

It fetches all the lines which have the word root in them.

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Awk Command**

In some cases the content needs to be filtered based on the columns in that case we use awk command.

Syntax: awk -F 'delimiter' '{print $column-number}' <filename>

awk -F : '{print $1}' /etc/passwd

It will print the first column of the file

awk -F : '{print $1,$2}' /etc/passwd

It will print the first and second column of the file

## **Additional things to learn.**

Regular Expressions.

<https://www.grymoire.com/Unix/Regular.html#uh-6>

SED & AWK

<https://github.com/chiranjibKonwar/Documentation/blob/master/Sed%20%26%20awk%20101Hacks%20%20.pdf>

BACKNEXT

* [Editor](https://killercoda.com/rkalluru/scenario/linux-cli-syntaxes)
* [Tab 1](https://killercoda.com/rkalluru/scenario/linux-cli-syntaxes)
* [**+**](https://killercoda.com/rkalluru/scenario/linux-cli-syntaxes)

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Linux Editors.**

There are so many editors which are part of different Linux Operating Systems. Editors like vi , vim , nano , gedit , emacs and more are mostly known editors. Among these 90% of the operating systems comes with vi editor as default editor.

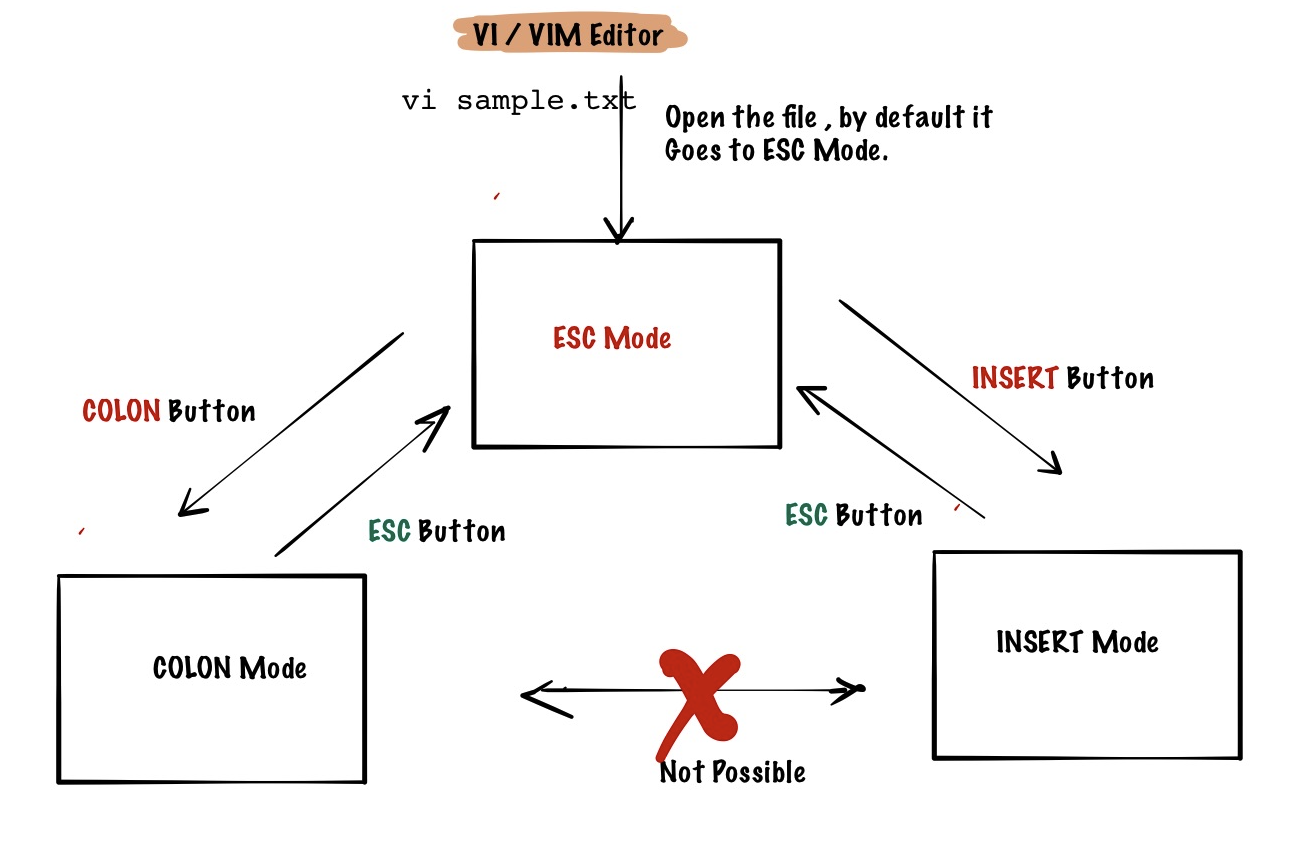
vi is very powerful editor and it comes with much enhanced options in vim . Hence, we choose to go with vim in our sessions.

## **VIM Editor.**

vim editor has three modes and each and every mode has its own purpose and allows you to perform certain actions.

1. ESC Mode
2. COLON Mode
3. INSERT Mode

All three modes are shown as below.



Following are the operations done by each and every mode.

### **ESC Mode.**

1. Copy a line yy / nyy (2yy)
2. Paste line p
3. Delete (Cut) a line dd / ndd
4. Undo (CTRL +Z in Windows) u

### **COLON Mode**

COLON is used to perform the following options.

1. Search Operation.

Ex: Search a word.

1. Ensure you are in ESC mode and press ":" to go to COLON Mode.
2. :/WORD , GIve WORD which you want to search.
3. Search & Replace

Ex: Search a Word and Replace

* 1. Ensure you are in ESC mode and press ":" to go to COLON Mode.
  2. :%s/WORD1/WORD2/ -> This will replace WORD1 with WORD2
  3. Flags : g, i :%s/WORD1/WORD2/g -> global means all possibilities on the line will be changed. :%s/WORD1/WORD2/i -> case-sensitive replace.

1. File Operations

1. Save File

:w

2. Quit Editor

:q

:q! -> Quit with out saving

3. Save and Quit

:wq

### **INSERT Mode**

INSERT mode is used to add your own content, whereas above two modes are dealing with existing content on the file.

**NOTE**: There are lot many operations are available, But we are talking which is needed for DevOps prospective.

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **FInding Files**

Most of the times when you login you have no idea where the files are located at. Since Linux doesn't have an UI it would be a tedious job to traverse through all the directories to find a single file but linux provides find command to search for a file with the name.

Syntax: find <location-to-find> <search-criteria>

find / -name passwd

This command searches through all the directories as we have given the location as / . You will find many files, but you can narrow down the search nby providing /etc directory.

find /etc -name passwd

This command only searches /etc directory for the passwd file. You can observe that the results are less in number.

## **Additional things to learn.**

<https://alvinalexander.com/unix/edu/examples/find.shtml>

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Command line browser**

Most of the times you need to browse urls to and fetch that content to command line. Some times we need partial information of the URL or the full information. curl command is available to browse the content over command line.

Syntax: curl <url>

curl www.google.com

Using curl command we can download the files.

curl https://archive.apache.org/dist/tomcat/tomcat-8/v8.0.0-RC1/bin/apache-tomcat-8.0.0-RC1-deployer.tar.gz -o apache-tomcat-8.0.0-RC1-deployer.tar.gz

Above command will download the file to the given filename. But with out giving the filename also we can download it to the default file name.

curl -O https://archive.apache.org/dist/tomcat/tomcat-8/v8.0.0-RC1/bin/apache-tomcat-8.0.0-RC1-deployer.tar.gz

## **Extracting the files from tar**

Many times in Linux world all the softwares are packaged either in .zip or .tar format. To extract the files from .tar extension we can use tar command.

Syntax: tar -xf <filename>.tar.gz

tar -xf apache-tomcat-8.0.0-RC1-deployer.tar.gz

To extract archives we use -x option and -f means file.

## **Extracting the files from zip**

Syntax: unzip <filename>.zip

curl -L -o shipping.zip https://github.com/roboshop-devops-project/shipping/archive/refs/heads/main.zip

unzip shipping.zip

## **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **Pipes**

Pipes are used to send the output of one command to another command without storing the content anywhere physically on disk.

Syntax : com1 | com2

Ex: cat /etc/passwd | grep root

**Note**: All the commands will not accept inputs over pipes. In case if we need to take the input then we take the help of xargs command.

touch sample.txt

ls

echo sample.txt | rm -f

ls

Now you can use the xargs command.

echo sample.txt | xargs rm -f

ls