**General Operations:**

* clear = Clears the terminal

**Directory Operatings:**

* pwd = Show current directory. Example Output: /home/ec2-user
* ls = List folders and files. Example Output: Desktop Downloads Pictures Documents
* cd [dirname] = Change directory to [dir]
* mkdir [dirname] = Make directory [dirname]
* cd .. = Go up a directory

**File Operations:**

* touch [filename] = Create [filename]
* rm [filename] = Delete [filename]
* rm -r [dirname] = Delete a non-empty directory and all the files in it
* rm -rf =delete a dir with all contents in it
* rm -d [dirname] or rmdir [dirname] = Delete an empty directory

**Navigating in the File System:**

* cd /usr/local/bin = Navigate multiple dirs (relative path - relative to current dir). Move to bin directory
* cd ../.. = Move up 2 hierarchies, so go to 'usr' directory
* cd /usr = Alternative to go to 'usr' directly (absolute path)
* cd [absolute path] = Move to any location by providing the full path
* cd /home/ec2-user = Go to my home directory (absolute path)
* cd ~ = Shortcut alternative to go to home directory
* ls /etc/network = List folders and files of 'network' directory

**More File and Directory Operations**

* mv [filename] [new\_filename] = Rename the file to a new file name
* cp -r [dirname] [new\_dirname] = Copy dirname to new\_dirname recursively meaning including the files
* cp [filename] [new\_filename] = Copy filename to new\_filename

**Some more useful commands**

* ls -R [dirname] = Show dirs and files but also sub dirs and files
* history = Gives a list of all past commands typed in the current terminal session
* history 20 = Show list of last 20 commands
* CTRL + r = Search history
* CTRL + c = Stop current command
* CTRL + SHIFT + v = Paste copied text into terminal
* ls -a = See hidden files too
* cat [filename] = Display the file content

**Display OS Information**

* uname -a = Show system and kernel
* cat /etc/os-release = Show OS information
* lscpu = Display hardware information, e.g. how many CPU you have etc.
* lsmem = Display memory information

**Execute commands as superuser**

* sudo [some command] = Allows regular users to run programs with the security privileges of the superuser or root
* su - admin = Switch from regular user to admin

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Yum package manager:**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++yum install git -y

which git

Yum remove git\* -y

cd /etc/yum.repos.d

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Vim Commands:**

* vim [filename] = Open file with Vim
* Press i key = Switch to Insert Mode
* Press esc key = Switch to Command Mode
* Type :wq = Write File to disk and quit Vim
* Type :q! = Quit Vim without saving the changes
* Type dd = Delete entire line
* Type d10 = Delete next 10 lines
* Type u = Undo
* Type A = Jump to end of line and switch to insert mode
* Type 0 = Jump to start of the line
* Type $ = Jump to end of the line
* Type 12G = Go to line 12
* Type 16G = Go to line 16
* Type /pattern = Search for pattern, e.g. /nginx
  + Type n = Jump to next match
  + Type N = Search in opposite direction
* Type :%s/old/new = Replace 'old' with 'new' throughout the file

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

sudo passwd ec2-user

sudo adduser tom

sudo cat /etc/passwd

sudo groups tom

sudo cat /etc/group

1 groups tom

sudo groupadd devops = Create new group (System assigns next available GID)

2 sudo usermod -g devops tom = Assign 'devops' as the primary group for 'tom' user

3 groups tom

6 sudo groupdel tom = Removes group 'tom'

7 sudo groups tom

8 sudo groups

9 sudo usermod -aG adm tom(appends him to adm group)

10 sudo groups tom

14 groups tom

ls

ls -l

touch demo.txt

mkdir demofolder

ls -l

ls -l /etc/passwd

* sudo chown tom:adm demo.txt = Change ownership of 'test.txt' file to 'tom' and group 'admin'
* sudo chown ec2-user demo.txt = Change ownership of 'test.txt' 'admin' user
* sudo chgrp devops demo.txt = Make 'devops' group owner of test.txt file

—-----------------------------------------------------------------------------------

**File permissions:**

chmod -x demo.txt (remove execute for owner,grp )

ls -l  
chmod g-w demo.txt (remove permissions of grp)

ls -l

chmod g+x demo.txt

chmod g-x demo.txt

cp demo.txt script.sh

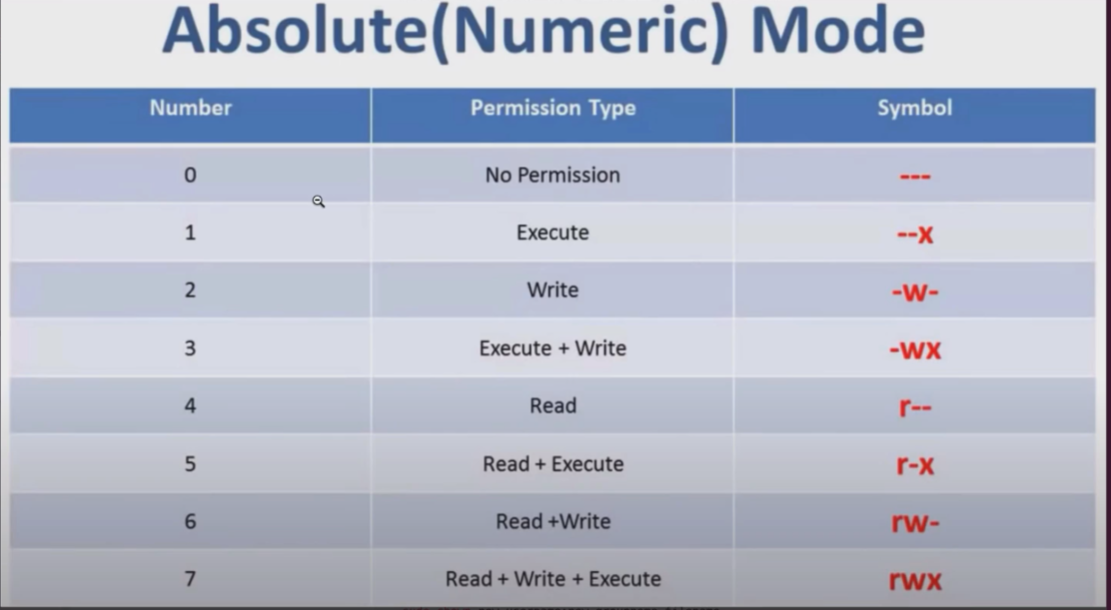
chmod u+x script.sh

chmod o+x script.sh

Otherway to change multiple permissions to file

chmod g=rwx test.txt

chmod g=r-w config.yml



chmod 777 script.sh

chmod u-w demo.txt

++++++++++++++++++++++++++++++++++++++++++++++++++

**Pipes and redirects:**

cat /etc/passwd | less ( esc or b)(output of 1 cmnd is piped to another prg)

history | less

**Grep :** stands for globally search for Regular exp and print out

Searched for a particular pattern of characters and displays all lines that contain that pattern

history | grep sudo

history | grep "sudo chmod"

history | grep sudo | less

ls /usr/bin/ | grep python

**Redirections:**

> character is the redirect operator

Takes the output from the previous cmnd and sends it to a file that give you

history | grep sudo > sudo\_commands.txt

cat sudo.commands.txt

cat sudo.commands.txt > sudo-rm-cmnds.txt

cat sudo-rm-cmnds.txt

history | grep rm > sudo-rm-cmnds.txt (it overwrites the existing content)

history | grep rm >> sudo-rm-cmnds.txt ( it appends to the existing content)

<https://www.tutorialspoint.com/unix/unix-file-operators.htm>

#!/bin/bash

echo "Server config"

file\_name=config.yaml

config\_dir=$1

if [ -d $config\_dir ]

then

echo "reading the content of the dir"

config\_files=$(ls "$config\_dir")

else

echo "config dir not found.creating one"

mkdir $config\_dir

touch "$config\_dir/sample.txt"

fi

echo "contents of $config\_dir dir: $config\_files"

user\_group=$2

if [ $user\_group == "preethi" ]

then

echo "Config the server"

elif [ $user\_group == "admin" ]

then

echo "admin the server"

else

echo "No permissions on the server"

fi

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

bash script.sh myfolder admin

! /bin/bash

echo "My first script"

echo "Setup and configure server"

file\_name=config.yaml

if [ -d newconfig ]

then

echo "reading the contents"

config\_files=$(ls newconfig)

else

echo "config dir not found creating one"

mkdir newconfig

touch newconfig/file2.txt

config\_files=$(ls newconfig)

fi

echo "using the $file\_name to configure"

echo "contents of config:$config\_files"

++++++++++++++++++++++++++++++++++++

#! /bin/bash

echo "My first script"

echo "Setup and configure server"

file\_name=config.yaml

config\_dir=$1

if [ -d $config\_dir ]

then

echo "reading the contents"

config\_files=$(ls newconfig)

else

echo "config dir not found creating on

mkdir newconfig

touch newconfig/file2.txt

config\_files=$(ls newconfig)

fi

echo "using the $file\_name to configure"

echo "contents of config:$config\_files"

user\_group=$2

if [ $user\_group == "preethi" ]

then

echo "Configure the server"

elif [ $user\_group == "admin" ]

then

echo "admin the server"

else

echo "No permission to configure the server"

fi