Linux Commands -- 14-06-2021 Class - 1 -- Create File

touch <filename> -> Create a file touch test -> Creates a file named test --- List Files and Directories

ls -> List the files and directories ls -l -> List the files and directories in long format ls -lt -> List based on time ls -lrt -> List in reverse order ls -a -> List all the Hidden files ls -ltrd -> to display single directory in long format ls -d*/ ->To display only subdirectories

-- pwd -> Shows Present Working Directory -- mkdir <dir_name> -> To create a directory mkdir -p dir1,dir2,dir3 -> To create a multiple directories cd <dir_name> -> Change Directory cd .. -> To go back one step/ To go back to previous directory cd -> To go back to home -- Text Editor

vi <filename> -> To create a file esc + i -> Insert Mode esc + :wq! -> Save and Quit w-write q-quit !-forcefully esc + :q! -> Quit without saving forcefully esc + :w -> Save

cat <filename> -> To display the contents of a file tac <filename> ->to display the content of the file reverse order

esc + :set nu -> TO set line numbers esc + :set nonu -> Remove line numbers -- Find and Replace

esc + :%s/<old_word>/<new_word>/g -> Replace a string esc + :%s/linux/windows/g esc + :%s/<old_word>/<new_word> ->Replace a string in first line esc + :s/<old_word>/<new_word> ->Replace the first instance of the line

% -> All lines s -> substitute g -> globally

esc+ :%s/linux/windows/ig -> Replace a string case insensitive

esc +:2s/linux/windows/g -> Replcae string in second line esc +:2,4s/linux/windows/g -> Replace string from lines 2-4 esc +:2,$s/linux/windows/g -> Replace string from line 2 to end of the document

esc +:4 -> Move cursor to the 4th line

esc +dd -> Delete a whole line -- Memory Check

du -sh <file_name> -> To check the size of the files/directories du -sh * -> To check the size of all the files inside that directory

df -h -> To check the disk size

free -h or -m-> To check system memory/RAM -- Print in Linux

echo -> To print in linux echo ”Hi” echo -e ”Hi \nHow are you” -- Redirect and Append

redirect (>) - Used to write output of a command to a file,it overwrite the file. If the file is not present it will create a new file

append (») - it will write the output of a command at the end of the file. it will also create a new file if the file is not created. -- 15-06-2021 Class - 2 -- Copy

cp file1 file2 -> Copy contents of file1 to file2 cp file_name dir_name/ --> Copy contents of file to a directory cp -R dir_name1 dir_name2 --> Copy contents of a dir_name1 to dir_name2 -- Move or Rename

mv file1 file2 --> Move from file1 to file2 mv file_name dir_name/ --> Move contents of file to a directory mv dir1 dir2/ --> Move the dir1 into dir2 -- Word Count

wc file_name --> To check the number of lines, words and characters in a file wc -l file_name --> Only no. of lines wc -w file_name --> Only no. of words wc -c file_name --> Only no. of characters -- grep - It is used to search for a string inside a file

grep ”pattern” file_name --> It will display all the lines with ”pattern” present grep -i ”pattern” file_name --> Case insensitive

grep -e ”pattern1” -e ”pattern” file_name --> It will check for both the patterns

grep -w ”pattern” file_name --> It will check for full strings not sub-strings

grep -l ”pattern” * --> Checks for the pattern in all the files in the directory and prints the filenames grep -l -R ”pattern” * --> Checks recursively

grep -L ”pattern” * --> It dispaly all the files in a directory that does’t have pettren in file and display the filenames.

grep ”^pattern” file_name --> Prints all the lines starting with pattern grep ”pattern$” file_name --> Prints all the lines ending with pattern

grep -c ”pattern” file_name --> To count the number of lines the pattern is present

grep -v ”pattern” file_name --> Prints all the lines without the pattern

grep -x ”pattern” * -->It print only the those lines which matches the search string

-- - - No Permission r- Read Permission w - Write Permission x - Execute Permission rw- owner rw- group r-- others

r - 4 2^2 w - 2 2^1 x - 1 2^0 - - 0

chmod 777 file_name chmod 444 file_name chmod 766 file_name chmod 644 file_name

rwx - owner, rw - group, rw - others =766

rw - owner, rw - group, rw - other = 666 rw - owner, rw - group, r - other = 664

We can change the permission in another way.

u-owner g-group o-others

chmod o+rwx <filename> --> Adds read,write and execute permission to others. chmod g+r <filename>--> adds read permission to group. chmod g-r <filename>--> removes the read permission to group. chmod u-x <filename> --> removes execute permisssion to owner.

-- 16-06-2021 Class - 3 ---

Default Linux

Files 666 Directories 777

For new files 664

666 666 666 002 222 000 664 444 666(rw-rw-rw-)

umask 222 - Tochange the default permissions of files and directories

/etc/profile - To change the umask value permanently

-- sudo - Super User Does

It gives permission for us to execute commands as a root user --- sudo su - --> To goto root user

To give an user sudo permission we have to add him in /etc/sudoers file user_name ALL=(ALL) NOPASSWD: ALL -- Add Users

sudo useradd <user_name> --> To add a new user sudo passwd <user_name> --> To set password for the user sudo userdel <user_name> --> To delete a user

To check all the users --> cat /etc/passwd [getent passwd]

-- Add Groups

groupadd --> To create a group groupdel --> To delete a group

usermod -aG <group_name> <user_name> - To add user to a group(to change user to a group)

To check all the groups --> cat /etc/group [getent group] --- chown <username> <file_name> --> To change owner of a file chgrp <group_name> <file_name> --> To change group that has acces to fileb

chown user_name:group_name <file_name> --> To change the owner and the group ---

who --> To check all the users that have logged into the system whoami --> To check the current user hostname --> To check the ip address curl ifconfig.me --> To display the public ip address uname --> to check the operating system and all its details uname -a --> To check all the details of the system --

Linux Architecture

Hardware - Kernel - Shell - User

hardware - It consists of all Pheripheral devices (RAM/CPU/HDD etc).

Kernel - Is the core part of linux, which is responsible for all major activities. It interacts directly with the underlying hardware

Shell: It is an interface between the users and the kernel, It takes the commands from the user and executes the kernel's functions

Cshell KShell Bash Shell Z Shell

chsh/lchsh <shell_name> --> To change the current shell echo $SHELL --> To check the current shell --- Head - To print required starting number of line

head <file_name> --> By default it displays starting 10 lines head -n <file_name> --> It will display the starting n lines --

Tail - To print last part of a file

tail <file_name> --> By default it displays last 10 lines tail -n <file_name> --> It will display the last n lines -- --- 17-06-2021 Class - 4 --- Pipe [|] --> To pass the output of one command to the next command

head -4 <file_name> | tail -1 --> To display only the 4th line tail -8 <filename> | head -1--> To display only the 8th line from bottom -- sed - Stream editor, used to find and replace

sed 's/old_word/new_word/g' <file_name>--> It will replace old_word with new_word and displays the output (But it wont make changes to the file)

sed -i 's/old_word/new_word/g' <file_name> --> It will make changes to the file as well sed -i ’1s/old_word/new_word/g’ <file_name> --> It will make changes to only the first line sed -i 1,3s/old_word/new_word/g <file_name> --> It will make changes from 1st to 3rd line sed -i 2,$s/old_word/new_word/g <file_name> --> It will make changes from 2nd to end of the file

sed ’2d’ <file_name> --> It will delete the second line sed ’2,5d’ sed ’2,$d’

sed -n ’2p’ <file_name> --> It prints the second line -- cut - Used to cut a file column wise

cut -d ” ” -f1 <file_name> --> It will display only the first with a single space as a seperator cut -d ” ” -f1,3 <file_name> --> It will display columns 1 and 3 cut -d ” ” -f1-3 <file_name> --> It will display columns from 1 to 3 -- awk - Used to cut a file column wise, row wise and there are multiple functions available

awk -F ” ” ’{print$1}’ <file_name> --> It will display only the first line awk -F ” ” ’{print$1,$3}’ <file_name> --> Will display 1 and 3 column awk -F ” ” ’{print$NF}’ <file_name> --> To display the last column awk -F ” ” ’{print$(NF-1)}’ <file_name> --> To display 2nd last column awk ’{print NF}’--> print the number of columns -- find - To find file or directories in linuxwqew find -name ”test” --> It will display all the file that are named test find -iname ”test” --> case insensitive search

find -mtime +90 --> Find all the files that are created more than 90 days ago find -mtime -10 --> Find all the files that are created in the last 10 days

-type f --> to find the files only -type d --> to find directories only

find -type f -perm 0777 --> it will display all the files with permission 777

find -type f -empty --> It will display all the empty files -- Assignment

1. Find all the non empty files--> find. -L -maxdepth 1 -type f -size 0 2. Find all the file with size more than 1 mb--> find. -type f -size +1mb 3. Find and delete all the empty file-->find. <filename> | xargs -rm -- -- 18-06-2021 Class - 5 -- Link - To create a shortcut of a file

ln -s <original_file_path> softlink_file_name --> It will create a soft link of the file in the present directory

ln <original_file_path> hard_link_file_name --> It will create a hard link

A softlink will point to the path of the file so once the original get deleted/removed the soft link will not work

A hardlink will point to inode of a file, so even if the original gets deleted the hardlink will still work inode - is an unique identification number of a file which point to the file's memory block -- SSH - Secure Shell or Secure Socket Shell is a network protocol that gives users/system administrators a secure way to access a computer/server over an unsecured network

Defualt Ports SSH - 22 HTTP - 80 Jenkins - 8080 Apache Tomcat - 8080

Port - A port is a virtual point where network connections start and end. Each port is associated with a specific poricess or service

```
ssh <key> user@ip.address
```

```
ssh -i <aws_pemkey> <user_name>@hostname
```

SSH Passwordless Connection

ssh-keygen -t rsa --> Used to generate a rsa key with which we can setup passwordless connection

Copy that rsa key (id_rsa.pub) to authorized_keys in .ssh directory in the server that you want to connect without password

-- scp to copy files over ssh

```
scp -i <pem_file> <file_name> user_name@hostname:<destination_path>
scp <file_name> user_name@hostname:<destination_path>
```

rsync --> We can also copy files with rsync but the major difference with rsync if there were any failures rsync will resume where the copy process stoppped -- Devops ------- test ssh

id_rsa.pub from devops server and then copy it to test ssh server

test shh ------- devops

id_rsa.pub from test ssh server then copy it to devops server

devops(172.31.44.159) --------> test ssh (172.31.37.153)

I have logged into devops server using putty

```
ssh -i pem_key ec2-user@172.31.37.153
```

Now i have logged into test ssh server -- Assignment: Install Apache Tomcat By Saturday

Requirement : Java

```
sudo yum install <package_name>
```

to Download files/directories from a link

wget <link> -- -- 19-06-2021 Class - 6 -- Telnet

Telnet is a networking protocol which is used to create a remote connection, it uses telnet protocol. This protocol has some security problems as data transfered using this protocol is unencrypted.

Defualt port for telnet:23

syntax: telnet <hostname> <port_number>

password: -- Process (PS)

PID --> Process ID

ps -ef --> To show all running process ps -u <user_name> --> To check all the process started by an user

kill --> To kill process by PID killall --> To kill process by processname

To forcefully Kill a Process [-9]

kill -9 <PID> --> To kill process by given PID kilall -9 <Process_name> --> To kill process by given processname

killall -u <user_name> --> to kill all the process started by an user

sudo service <process_name> stop --> Gracefully Killing a process

sudo service <tomcat> stop --> It will stop tomcat service sudo service <docker> stop --> It will stop Docker service

top --> Check all the process running, memory usage, cpu usage etc -- bashrc and bashprofile

bashrc or bashprofile --> these files executes everytime automatically as the session starts

We generally set up environment variables here in bashrc/bashprofile

we can also setup alias in bashrc/bashprofile

alias FE="find -type f -empty"

bash_history --> It will store all the commands that was given to linux

history --> To check the commands given to linux

-- ping <hostname> --> To check and ping another server ping 0 --> To ping current server

-- uniq and sort

uniq <file_name> --> To display only the unique values in a file

Drawback of uniq command is that it will only remove the duplicate values if they are adjacent to each other

That is that reason we generally use sort and unique commands together sort <file_name> --> To sort values in a file sort -r <file_name> --> In reverse order --- tee

command > file_name

ls -lrt | tee <file_name> --> Used to write a output of a command to a file and also display on the terminal ls -lrt | tee -a <file_name> --> append

--- netstat --> To check the information about ports in linux

netstat -a --> To check all the available ports netstat -l --> To check the ports that are in used sudo netstat -tulnp --> To check which process are using which port --- & --> Run a command/script in background

syntax: command/script &

fg --> To bring the process/command/script to the foreground

syntax: fg PID --- Assignment : Difference btw .bashrc and .bashprofile ---