 Creating an ec2 instance (Server, Virtual Machine, Node)

• Launch ec2 instance in AWS

• Create an AWS account

• wait for the account to get activated

• Search for EC2 service  Goto Instances  Launch Instances

a. Give a name to the machine (instance)

b. Choose the OS type (AMI type)

c. Create and save the Key Pair to login

• Create an instance

 How to connect to remote Linux?

ssh (Secure Shell)

• The ssh command is used to securely log into a remote machine and execute commands on that machine.

• The basic syntax of the command is “ssh user@host”, where the user is the username on the remote machine and the host is the IP address or DNS or hostname of the remote machine.

• Syntax: ssh -i <identiy\_key\_path> <username>@<ip\_address>

• Example: ssh -i my\_ec2.pem ubuntu@192.168.43.120

Windows: We can connect using third-party tools like PuTTY, MobaXterm, Git Bash etc. to execute SSH

Mac and Linux: We can directly execute the SSH command in the Terminal.

 Linux keynotes to know:

• In Linux commands are case-sensitive

• A folder is called a Directory in Linux

 COMMANDS

1. pwd (present working directory) - Know your current path/location

2. ls: list files and directories

3. ls -l: To list with details (long format)

4. ls -t: To list with creation time sorted (recent at the top)

5. ls -r: To list reverse order: -------------------

6. ls -lrt OR ll: All the above together

7. cd (change directory): Switch/change to a folder/Directory in Linux

• Syntax: cd <path>/<location>

• Example: cd test - To change to a Directory in the current location

8. mkdir (make directory) - To create a directory

• Syntax: mkdir <directory\_name>

• Example: mkdir test

• To create multiple directories:

mkdir <directory\_name> <directory\_name1> <directory\_name2> ..... <directory\_name>

Example: mkdir test test1 test2 ...

Note: directory names with space separated.

9. tree (structured and recursive directory listing)

By default, the tree command is not installed. To install the same

• To install in RedHat (RHEL) or CentOS

 sudo yum update

 sudo yum install tree

• To install in Ubuntu

 sudo apt update

 sudo apt install tree

10. rm: Delete files/directories - Use the rm command to delete files and directories. But rm cannot simply delete a directory.

11. rm -r (OR) rmdir: To delete a directory. In this case, it deletes both the folder and the files in it.

• To remove/delete a file

 rm file\_name

• To remove/delete a Directory

 rm -r directory\_name (OR) rmdir directory\_name

12. mv - Command to move files and directories through the command line. Rename or move the file from one location to another

We can also use the mv command to rename a file/directory.

• To rename a file/directory

 mv old\_filename new\_filename

 mv old\_directory new\_directory

• To move to a different location (cut and paste)

 mv <source> <destination>

13. cp - Command copies files and directories through the command line. Copy the file from one location to another (cp)

• To copy to a different location

 cp <source> <destination>

 Vi Editor

Command Mode - we can't add any content. This is used to run vi-related commands.

Initially when we do vi to a file, command mode is the default mode.

• Only Save the changes to the file - :w

• To quit from vi - :q!

• Save and quit from vi - :wq! or (Press Escape - Hold shift + zz)

• Save to another file - :w <new\_file>

• To search a word - /<word>

• To display line numbers - :set nu

• To replace word - s/<old\_word>/<new\_word>/g ------------------

• To display line numbers - :nu

• dd - cut the entire line where the cursor is

• yy - copy the entire line where the cursor is

• p - paste the clipboard content next to the cursor

Insert Mode – It is used to add the content to the file

• Press “I” to go to insert mode

• Press '<escape>' to exit insert mode and move to command mode.

14. cat - To display/print the contents of files) – It is short for “concatenate “, which allows us to create single or multiple files, view content of files, concatenate files and redirect output in the terminal or files.

• Syntax: cat <file\_name>

15. echo - The echo command is a built-in command in most Unix-like operating systems, including Linux and macOS. It is used to display text to the standard output or console or in the terminal.

• Syntax: echo <text\_line to display>

• Example: echo "My name is Chandan......"

Linux path - Always Linux path is separated with /

• . -> current directory

• .. -> one step back / Previous one directory

• ../.. -> Previous two directory

16. grep (To search pattern/string in a file or content) - The grep command is a filter that is used to search for lines matching a specified pattern and print the matching lines to standard output.

• Match all lines that start with ‘hello’ Eg: “hello there”  grep “^hello” file1

• Match all lines that end with ‘done’ Eg: “welldone”  grep “done$” file1 -------------

• Match all lines that contain any of the letters ‘a’,‘b’,‘c’,‘d’or‘e’  grep “[a-e]” file1

• Match all lines that do not contain a vowel grep “[^aeiou]” file1

• Match all lines that start with a digit following zero or more spaces. Eg: “1.”or“2.”

 grep “ \*[0-9]” file1

• Searching in all files recursively using  grep-r grep -r "ramesh" \*

• Search multiple patterns  grep -v -e "pattern" -e "pattern"

17. find - Find command can be used to find files and directories and perform subsequent operations on them. It supports searching by file, folder, name, creation date, modification date, owner and permissions. By using the ‘-exec’ other commands can be executed on files or folders found.

• SYNTAX: find <location\_to\_find> [options]

a. Search a file with specific name find . -name file.txt

b. Search a file with specific name with ignore case find . -iname file.txt

c. Search a files in multiple directories  find . /home /user -name file.txt

d. Search only files containing name  find . -type f -iname file.txt

e. Search only directories containing name  find . -type d -iname file.txt

f. Search for empty files and directories  find . -empty

g. Search for file with permissions (655)  find . -perm 655

h. Search text within multiple files  find./-typef-name"\*.txt"-execgrep'search\_string' {}\;

i. find files by last modification time  find . -mtime days

• 24 hours -> find . -mtime 1

• modified less than 7 days (7 days to till) -> find . -mtime -7

• Last 50-100 Days Modified Files -> find . -mtime +50 –mtime -100

j. Find Last 50 Days Accessed Files  find . -atime 50

k. Find Changed Files in Last 1 Hour  find / -mmin -60

l. Find Accessed Files in Last 1 Hour  find / -amin -60

18. free - Free command is used to check the used and available space of physical memory and swap memory (ram/memory) in KB. The free command displays:

• Total amount of free and used physical memory

• Total amount of swap memory in the system

• Buffers and caches used by the kernel

19. du - “du” (Disk Usage) is a standard Unix/Linux command, used to check the information of disk usage of files and directories on a machine.

20. du -h - Using “-h” option with the “du” command provides results in Human Readable Format. This means you can see sizes in Bytes, Kilobytes, Megabytes, Gigabytes etc.

21. df - The command df stands for "disk filesystem". With the -h option (df -h) it shows the disk space in "human readable" form

22. ps - ps command is used to display information about the processes that are running in the system.

23. ps -ef - To view current running processes

24. top - The top command also gives you a real-time update on how much of your swap space is being used.

NOTE:

• /proc/meminfo

• cat /proc/meminfo will contain dynamic information about the kernel and the system.

25. chmod

• The chmod command is used to change the access mode of a file.

• The name is an abbreviation of change mode.

Example:

1. Read, write and execute permissions to the file owner:

chmod u+rwx <file\_name> (OR) chmod 700 <file\_name>

2. Read, write to file owner, Read to groups and Write, Execute to Others:

chmod u+rw, g+r, o+wx <file\_name> (OR) chmod 643 <file\_name>

 | (pipe)

• Using it the output of one command can be given as input to another command

• The Unix/Linux systems allow the stdout of a command to be connected to the stdin of another command.

• The pipe is used to combine two or more commands, and in this, the output of one command acts as input to another command

command\_1 | command\_2 | command\_3 | .... | command\_N

1. Print a particular directory name - ls | dir1

2. Finding all the file patterns – find . | grep file

3. Finding all the files which have pattern (file) and then getting a particular file

find . | grep file | grep profile

4. To count the number of lines in find . ‘s input – find . | wc -l

(wc -l  Total lines, wc -w  Total words, wc -c  Total characters)

5. Getting the total of the number files names file - find . | grep find | wc -l

 > (Write to a File using the Redirection Operator)

• Redirection allows you to capture the output from a command and overwrite it as input to another command or file.

• It will overwrite/Erase all the content of a file and add new content to the file.

• Whatever the content it will written to a file and nothing will be printed in the console.

echo "This is Redirection of text to file" > file.txt

 >> (Append to a File using the double Redirection Operator)

• Double Redirection allows you to capture the output from a command and send it as input to another command or file.

• It will append the content to the end of the file by keeping the old content.

echo "This is Redirection of text to file" >> file.txt

 tee

• To write to a file and also to print the written content in the console/terminal

echo "This is Redirection of text to file" | tee file.txt

• To append to a file and also to print the written content in the console/terminal

echo "This is Redirection of text to file" | tee -a file.txt

(OR)

echo "This is Redirection of text to file" | tee --append file.txt

 head

• As the name implies, print the top N number of data of the given input.

• By default, it prints the first 10 lines of the specified files.

• If more than one file name is provided then data from each file is preceded by its file name.

head -n N <file\_name>

(OR)

head -N <file\_name>

 tail

• As the name implies, print the bottom N number of data of the given input.

• By default, it prints the first 10 lines of the specified files.

• If more than one file name is provided then data from each file is preceded by its file name.

tail -n N <file\_name>

(OR)

tail -N <file\_name>

Example:  To print 20 lines from top: head -20 <file\_name>

 To print 6 lines from bottom: tail -6 <file\_name>

 sed

• SED command in UNIX stands for stream editor and it can perform lots of functions on file like searching, finding and replacing, insertion or deletion.

• Consider the following text file as the input file for all cases below. (file.txt)

• Syntax: sed ‘Operation-type (s)/Source/Destination/Scope (g) file\_name.txt

• S  Substitute, g  Scope (Occurrence, here “g” means changes as to be made in whole file)

• We can also use “ | ” instead of “ / ”, Eg: sed 's|unix|linux|g' file.txt

“Unix is a great os. Unix is open source. Unix is a free os.

learn operating systems.

Unix Linux which one you choose.

Unix is easy to learn. Unix is a multiuser os. Learn unix. Unix is a powerful.”

1. Replacing or substituting string - sed 's/Unix/Linux/g' sedexample.txt

2. Above command prints in console only, won’t alter anything original file

To save changes in original file – sed -i 's/Unix/Linux/g' sedexample.txt

3. To save changes + case insensitive – sed -i 's/ linux/unix /ig' sedexample.txt

4. Replacing the nth occurrence of a pattern in a line - sed 's/unix/linux/2' sedexample.txt (Replaces 2nd occurences of a word, for example – If we have 2 “unix” words in one line, it will replace only 2nd occurrence of that particular line)

5. Replacing all the occurrences and saving – sed -i 's/ linux/unix/g' sedexample.txt

6. Replacing from nth occurrence to all occurrences in a line - sed 's/unix/linux/3g' sedexample.txt

(Replacing starts from 3rd occurrence and because of “g”, it replaces all the unix words to linux thereafter)

7. Replacing string on a specific line number - sed '3s/unix/linux/' sedexample.txt

8. Replacing all the given strings on a specific line number – sed ‘3s/unix/linux/g’ sedexample.txt

9. Printing only the replaced lines - sed -n 's/choose/select/p' sedexample.txt

10. Deleting lines from a particular file, say 5 in this example - sed '5d' sedexample.txt

11. To Delete a last line - sed '$d' sedexample.txt

12. To Delete line from range x to y - sed '3,6d' sedexample.txt

13. To Delete from the nth to the last line - sed '12,$d' sedexample.txt

14. To Delete pattern-matching line - sed '/pattern/d' sedexample.txt

15. Print a specific line from a file - sed -n '2p' sedexample.txt

 Uname

• uname command displays important information about the system such as — Kernel name, Host name, Kernel release number, Processor type, etc.,

• To check the linux version

cat /etc/os-release (OR) lsb\_release -a

 whereis (OR) which

When you want to find out where a specific Unix command exists (for example, where does the ls command exist?), you can execute the following command.

 whatis - Whatis command displays a single line description about a command.

 who - who command is used to find out the following information:

• Time of last system boot

• Current run level of the system

• List of logged-in users and more.

a. To display all details of the current logged-in user: who -a

b. To display a current run level of the system: who -r

c. To show the time of the system when it booted last time: who -b -H

d. To show a list of users logged in to the system: who -u

 whoami - To display the system’s username

 cut - The cut command in UNIX is a command for cutting out the sections column using a delimiter from each line of files and writing the result to standard output.

Syntax: cut -d "delimiter" -f (field number) file.txt

• To cut the data with “ “ (space) as a delimiter and print the first column of data

cut -d " " -f 1 state.txt

• To cut the data with “ “ (space) as delimiter and print first to fourth column data

cut -d " " -f 1-4 state.txt

 awk - The awk command in UNIX is a command for cutting out the sections column using a field separator from each line of files and writing the result to standard output.

Consider the following text file as the input file for all cases below.

$cat > employee.txt

Ajay manager account 45000

Sunil clerk account 25000

varun manager sales 50000

Amit manager account 47000

a. Default behaviour of Awk: By default, Awk prints every line of data from the space-separated field of the file.

awk '{print $1}' employee.txt

(or)

awk -f " " '{print $1}' employee.txt

b. To print the last column of a file using NF – Number of Fields/Columns

How to get the last word from a line in the file.

awk ‘{print $NF}’ employee.txt