Devops:-

Git:-

CVCS

VS

DVCS

Repository :

Commit:

Setting up git:

Commands :

Configuration: 1. git config –global user.name “ username”

2. git config –global user.mail “email id”

Configure our first repo:

mkdir

cd to directory

git init

git status

git commit –m “this coment”

git add mail.txt

git status

git log :-

git diff

git clone “giturl”

git push

1.“how can we add more number of file at a time “

git pull

git remote add origin <repository name>

git checkout –test.py

What is a branch: -

git checkout –b <name of the branch>

git checkout <branch name>

git merge <branch name>

Rebase:

git rebase –i

hosting reposi

Linux

>Download and install oracle virtual box

>Download and install vagrant

>we need to run ruby file

Using “vagrant up” in cmd at file location

>sign up into “aws management server”

>Ec2

>instances

>lounchences

LINUX Command’s Explanation:

------------------------------------------------------------------------------------------------------------------------------------------

In Linux/Unix, everything is a file. Regular file, Directories, and even Devices are files. Every File has an associated number called File Descriptor (FD).

Ex : Your screen also has a File Descriptor.

Whenever you execute a program/command at the terminal, 3 files are always open, viz., standard input, standard output, standard error.

1."sudo apt-get install <packagename>" :to install the software.

Ex: - sudo apt-get install vim --🡪 to install Vim editor in linux.

2."sudo apt-get update" : to update the system.

3.You can also send e-mails from terminal using the 'mail' command

Command: - "mail -s 'subject' -c 'cc-address' -b 'bcc-address' 'to-address' "

Ex: - mail -s "news" adkad@gmail.com

hi

this is news of the day.

1.adadddsd

2.sd........

Press Cntrl+D you are finished writing the mail. The mail will be sent to the mentioned address.

4. "cat > filename" Creates a new file

5. "cat filename" to display the content of the file

6. "cat file1 file2 > file3” Joins two files (file1, file2) and stores the output in a new file (file3)

7. "mv file "new file path" " moves a file or directory to another location in the directory structure.

8. "mv filename new\_file\_name" Renames the filename to a new filename

10. "sudo" Allows regular users to run programs with the security privileges of the super user or root

11. "rm" Deletes a file

Ex: - rm –rf <file path> or <file name>

rm <file name>

12. "man" Gives help information on a command

Ex : - man du -------🡪 gives du command information.

13. "history" Gives a list of all past commands typed in the current terminal session

14. "mkdir" Creates a new directory in the present working directory or at the specified path

15. "rmdir" Deletes a directory

16. "pr -x" Divides the file into x columns. normally pr used to paginate or columnate files for printing. It can also be used to compare two files side by side, as an alternative to diff.

16.1."pr -h" Assigns a header to the file.

16.2. "pr -n" Denotes the file with Line Numbers

17. "lp -nc" or "lpr c" Prints "c" copies of the File. The **lp command** is used to print files on **Unix** and Linux systems. The name "**lp**" stands for "line printer"

18. "lp -dprintrname" or "lp -Pprintername" Specifies name of the printer

19."chmod permissions filename" we can set permissions (read, write, execute) on a file/directory for the owner, group and the world.

example : "chmode o=rwx filename" or "chmode u+x file name" or "chmode g-w filename"......

u user/owner

g group

o other

a all

+ Adds a permission to a file or directory

- Removes the permission

= Sets the permission and overrides the permissions set earlier.

20."chown user" For changing the ownership of a file/directory,

21."chown user:group filename" In case you want to change the user as well as group for a file or directory.

22. "chgrp group\_name filename" In case you want to change group-owner only,

23. "groups" To find the all the groups you are a member of

24. "newgrp groupname " To work as a member a group other than your default group

25. "ls -al > listings" Here the output of command ls -al is re-directed to file "listings" instead of your screen. ">" is used for STDOUT output.

You can redirect standard output, to not just files, but also devices!

ex: cat music.mp3 > /dev/audio

The cat command reads the file music.mp3 and sends the output to /dev/audio which is the audio device.

If the sound configurations in your PC are correct, this command will play the file music.mp3

26. "Mail -s "Subject" to-address < Filename " The '<' symbol is used for input(STDIN) redirection

If you want to attach a File to email you can use the input re-direction operator

The Pipe is a command in Linux that lets you use two or more commands such that output of one command serves as input to the next.

The symbol '|' denotes a pipe

27. "cat filename | less " which will show you only one scroll length of content at a time.

Instead of 'less', you can also use. 'pg' and 'more' commands,The less ,pg and more commands are used for dividing a long file into readable bits

28. "grep search\_string" To search existing string in a perticuler file or directory.

Option Function

-v Shows all the lines that do not match the searched string

-c Displays only the count of matching lines

-n Shows the matching line and its number

-i Match both (upper and lower) case

-l Shows just the name of the file with the string

Ex: cat filename | grep -i a Using the 'i' option grep has filtered the string 'a' (case-insensitive) from the all the lines.

29. "sort Filename" This command helps in sorting out the contents of a file alphabetically.

Optio Function

-r Reverses sorting

-n Sorts numerically

-f Case insensitive sorting

Linux has a lot of filter commands like awk, grep, sed, spell, and wc. A filter takes input from one command, does some processing, and gives output.

When you pipe two commands, the "filtered " output of the first command is given to the next.

Linux/Unix Pipes , Grep & Sort Command

30. "cat sample | grep -v a | sort - r" We want to highlight only the lines that do not contain the character 'a', but the result should be in reverse order.

Regular expressions are special characters which help search data, matching complex patterns. Regular expressions are shortened as 'regexp' or 'regex'.

some of the basic Regex.

Symbol Descriptions

. replaces any character

^ matches start of string

$ matches end of string

\* matches up zero or more times the preceding character

\ Represent special characters

() Groups regular expressions

? Matches up exactly one character

31. " cat sample | grep -E p\{2}"

We want to check that the character 'p' appears exactly 2 times in a string one after the other. For this the syntax would be:

Expression Description

{n} Matches the preceding character appearing 'n' times exactly

{n,m} Matches the preceding character appearing 'n' times but not more than m

{n, } Matches the preceding character only when it appears 'n' times or more

Note: You need to add -E with these regular expressions.

32. "cat sample|grep "a\+t"" Suppose we want to filter out lines where character 'a' precedes character 't'

Expression Description

-------------------------------------------

\+ Matches one or more occurrence of the previous character

\? Matches zero or one occurrence of the previous character

The syntax for brace expansion is either a sequence or a comma separated list of items inside curly braces "{}".

The brace expansion is used to generate strings. It helps in creating multiple strings out of one.

33. "echo {0..11} " or "echo {a..z}" or "echo a{0..9}b"

34. "echo $LANG" environment variable stores the value of the language that the user understands.

PATH This variable contains a colon (:)-separated list of directories in which your system looks for executable files.

USER The username

HOME Default path to the user's home directory

EDITOR Path to the program which edits the content of files

UID User's unique ID

TERM Default terminal emulator

SHELL Shell being used by the user

35. "env" Command displays all the. Linux - Environment Variables

36. "VARIABLE\_NAME=variable\_value" You can create your own user defined environment variables.

37. "unset variablename" The following syntax can be used to remove a Variable from the system.

38. "export Variable=value" To set value of an environment variable

39. "SSH username@ip-address or hostname" The syntax to log into a remote Linux machine using SSH

40. "ping hostname=""" To check whether your connection to the server is healthy or not.You may Press Ctrl + c to exit from the ping loop.

ping also used for

Analyzing network and host connections

Tracking network performance and managing it

Testing hardware and software issues

FTP is file transfer protocol. It's the most preferred protocol for data transfer amongst computers.

You can use FTP to -

Logging in and establishing a connection with a remote host

Upload and download files

Navigating through directories

Browsing contents of the directories

41. "ftp hostname=""" To establish an FTP connection to a remote host, Once you enter this command, it will ask you for authentication via username and password.

Once a connection is established, and you are logged in, you may use the following commands to perform different actions.

Command Function

dir Display files in the current directory of a remote computer

cd "dirname" change directory to "dirname" on a remote computer

put file upload 'file' from local to remote computer

get file Download 'file' from remote to local computer

quit Logout

Telnet helps to - connect to a remote Linux computer run programs remotely and conduct administration.

42. "telnet hostname=""" Telnet utility. The utility will ask your username and password.

Once authenticated, you can execute commands just like you have done so far, using the Terminal. The only difference is,

if you are connected to a remote host, the commands will be executed on the remote machine, and not your local machine.

You may exit the telnet connection by entering the command 'logout'

43. "top" This utility tells the user about all the running processes on the Linux machine.

The terminology follows:

Field Description Example 1 Example 2

PID The process ID of each task 1525 961

User The username of task owner Home Root

PR Priority Can be20(highest) or -20(lowest) 20 20

NI The nice value of a task 0 0

VIRT Virtual memory used (kb) 1775 75972

RES Physical memory used (kb) 100 51

SHR Shared memory used (kb) 28 7952

S Status S R

There are five types:

'D' = uninterruptible sleep

'R' = running

'S' = sleeping

'T' = traced or stopped

'Z' = zombie

%CPU % of CPU time 1.7 1.0

%MEM Physical memory used 10 5.1

TIME+ Total CPU time 5:05.34 2:23.42

Command Command name Photoshop.exe Xorg

44. "ps ux" To check all the processes running under a user, The command stands for 'Process Status'.

It is similar to the "Task Manager" that pop-ups in a Windows Machine.

45. "kill PID" This command terminates running processes on a Linux machine.

To use these utilities you need to know the PID (process id) of the process you want to kill

46. "pidof Process name" To find the PID of a process

NICE

Linux can run a lot of processes at a time, which can slow down the speed of some high priority processes and result in poor performance. To avoid this, you can tell your machine to prioritize processes as per your requirements. This priority is called Niceness in Linux, and it has a value between -20 to 19. The lower the Niceness index, the higher would be a priority given to that task. The default value of all the processes is 0.

47. "nice -n 'Nice value' process name" To start a process with a niceness value other than the default value

48. "renice 'nice value' -p 'PID' " If there is some process already running on the system, then you can 'Renice' its value using syntax.

DF

This utility reports the free disk space (Hard Disk) on all the file systems.

49. "'df -h'" If you want information in a readable format,

50. " Free" This command shows the free and used memory (RAM) on the Linux system.

You can use the arguments

free -m to display output in MB

free -g to display output in GB

51. "bg" To run a stopped process in the foreground

52. "fg" To run a stopped process in the foreground

The VI editor is the most popular and classic text editor in the Linux family. Below, are some reasons which make it a widely used editor – It is available in almost all Linux Distributions It works the same across different platforms and Distributions It is user-friendly. Hence, millions of Linux users love it and use it for their editing needs To work on VI editor, you need to understand its operation modes. They can be divided into two main parts.

Command mode:

The vi editor opens in this mode, and it only understands commands In this mode, you can, move the cursor and cut, copy, paste the text This mode also saves the changes you have made to the file Commands are case sensitive. You should use the right letter case.

Insert mode:

This mode is for inserting text in the file. You can switch to the Insert mode from the command mode by pressing 'i' on the keyboard Once you are in Insert mode, any key would be taken as an input for the file on which you are currently working.

To return to the command mode and save the changes you have made you need to press the Esc key

To launch the VI Editor -Open the Terminal (CLI) and type

54. "vi <filename\_NEW> or <filename\_EXISTING>"

EX: "vi sample1"

vi editor opens in command mode

press "i" to enter insert mode

add the content

press "Esc" to enter to command mode

enter ":wq" and hit enter

to check the file content type "cat sample1" and hit enter

Note: You should be in the "command mode" to execute these commands. VI editor is case-sensitive so make sure you type the commands in the right letter-case.

Keystrokes Action

i Insert at cursor (goes into insert mode)

a Write after cursor (goes into insert mode)

A Write at the end of line (goes into insert mode)

ESC Terminate insert mode

u Undo last change

U Undo all changes to the entire line

o Open a new line (goes into insert mode)

dd Delete line

3dd Delete3 lines.

D Delete contents of line after the cursor

C Delete contents of a line after the cursor and insert new text. Press ESC key to end insertion.

dw Delete word

4dw Delete 4 words

cw Change word

x Delete character at the cursor

r Replace character

R Overwrite characters from cursor onward

s Substitute one character under cursor continue to insert

S Substitute entire line and begin to insert at the beginning of the line

~ Change case of individual character

Make sure you press the right command otherwise you will end up making undesirable changes to the file.

You can also enter the insert mode by pressing a, A, o, as required.

Keystroke Use

Shift+zz Save the file and quit

:w Save the file but keep it open

:q Quit without saving

:wq Save the file and quit

Learning to use this editor can benefit you in creating scripts and editing files.

SHELL Scripting:

-------------------------------------

An Operating is made of many components, but its two prime components are -

Kernel

Shell

A Kernel is at the nucleus of a computer. It makes the communication between the hardware and software possible. While the Kernel is the innermost part of an operating system, a shell is the outermost one.

A shell in a Linux operating system takes input from you in the form of commands, processes it, and then gives an output. It is the interface through which a user works on the programs, commands, and scripts.

A shell is accessed by a terminal which runs it. When you run the terminal, the Shell issues a command prompt (usually $), where you can type your input,

which is then executed when you hit the Enter key. The output or the result is thereafter displayed on the terminal.

The Shell wraps around the delicate interior of an Operating system protecting it from accidental damage. Hence the name Shell

There are two main shells in Linux:

1. The Bourne Shell: The prompt for this shell is $ and its derivatives are listed below:

POSIX shell also is known as sh

Korn Shell also knew as sh

Bourne Again SHell also knew as bash (most popular)

2. The C shell: The prompt for this shell is %, and its subcategories are:

C shell also is known as csh

Tops C shell also is known as tcsh

Shell scripting is writing a series of command for the shell to execute. It can combine lengthy and repetitive sequences of commands into a single and simple script, which can be stored and executed anytime. This reduces the effort required by the end user.

55. "#!/bin/sh" "#!" is an operator called shebang which directs the script to the interpreter location. So, if we use"#! /bin/sh" the script gets directed to the bourne-shell(bash).

Ex: vi sample1

#!/bin/sh

ls

56. "bash file name" To exicute the script.

57. "#coments " To add coments with in the script.

#!/bin/sh

echo "what is your name?"

read name

echo "How do you do, $name?"

read remark

echo "I am $remark too!"

Virtual Terminals enable a number of users to work on different programs at the same time on the same computer. This is the reason they are one of the most distinguished features of Linux.

Usually, there are six (default) virtual terminals on a Linux operating system, and you can log into them as different users to conducts different tasks.

The steps to launch a Virtual terminal are:

1) Press Ctrl+Alt+F1

2) Enter User ID and Password

3) Now the Virtual Terminal is ready to work on

You can navigate between the 6 virtual terminals using the following command.

"Ctrl + Alt + F (1 to 6) key"

Virtual Terminal shortcuts

These are some of the shortcuts that you should be aware of while working on virtual terminals.

Shortcut Function

Home or Ctrl + a Move the cursor to the start of the current line

End or Ctrl + e Move the cursor to the end of the current line

Tab Autocomplete commands

Ctrl + u Erase the current line

Ctrl + w Delete the word before the cursor

Ctrl + k Delete the line from the cursor position to the end

reset Reset the terminal

history List of commands executed by the user

Arrow up Scroll up in history and enter to execute

Arrow down Scroll down in history and enter to execute

Ctrl + d Logout from the terminal

Ctrl + Alt + Del Reboot the system

To return to the home screen of the Linux system, use Ctrl+Alt+F7 and it would take to you the terminal

Creating a User:

In Linux, every user is assigned an individual account which contains all the files, information, and data of the user. You can create multiple users in a Linux operating system. The steps to creating a user are:

Using Terminal:

Step 1) Use command sudo adduser "sudo adduser name"

Step 2) Enter password for the new account and confirm

Step 3) Enter details of the new user and press Y

Using GUI:

Step 1) Go to the system settings look for an icon which says 'User Accounts'.

Step 2) Click on the unlock icon and enter a password when prompted, then click the plus sign.

Step 3) A new window would pop up, asking you for adding information to the new user account.

The account type offers two choices - standard and administration(Ubuntu Limitation).

If you want the new user to have administrative access to the computer, select Administrator as the account type.

Administrators can do things like add and delete users, install software and drivers, and change the date and time.

Step 4) The new account would show, but would be disabled by default.

To activate it, click the password option and add a new password. Click change to enable the account.

Deleting, disabling account:

Terminal

For disabling an account using Terminal, remove the password set on the account.

58. "sudo passwd -l 'username'"

To delete an account, use the command -

59. "sudo userdel -r 'username'"

You can view the existing groups on your Linux operating system by entering the following command:

60. "groupmod "Press Tab key twice""

To add a user to a group, use the following syntax:

61."sudo usermod -a -G GROUPNAME USERNAME"

62. "cat /etc/group" You can check whether the user is in a group by this command.

63. "sudo deluser USER GROUPNAME" Removing a user from Usergroup

If you do not want to run the commands in terminal to manage users and groups, then you can install a GUI add-on .

64. "sudo apt-get install gnome-system-tools"

Once done, type

65. "users-admin"

Finger:

This command is used to procure information of the users on a Linux machine. You can use it on both local & remote machines .

66. "finger" gives data on all the logged users on the remote and local machine.

The syntax 'finger username' specifies the information of the user.

Linux/Unix user management commands

Here is a list of linux user management commands

Command Description

sudo adduser username Adds a user

sudo passwd -l 'username' Disable a user

sudo userdel -r 'username' Delete a user

sudo usermod -a -G GROUPNAME USERNAME Add user a to a usergroup

sudo deluser USER GROUPNAME Remove user from a user group

finger Gives information on all logged in user

finger username Gives information of a particular user

67. "last -a" To check who logedin into the mition

68. "who -u" To check who working on the mition

69. "less /var/log/secure" To check logs of all the attempts made for login to your machine with authentication reports

70. "tail –f" you want to view the content of the log files real time.

Linux Log Files:

Linux provides a centralized repository of log files that can be located under the /var/log directory.

The log files generated in a Linux environment can typically be classified into four different categories:

Application Logs

Event Logs

Service Logs

System Logs

71. "last reboot" To find out when was the system last rebooted,

72. "lastlog | more" To see when did someone last log in to the system,

73. "<comand> --help" To know about command specified by the user

74. "sudo chattr +i -V /path of the file" To set the file to immutable (or undeletable).

75. "lsattr" To check the permitions of a file or directory.

76. "sudo chattr +i -RV /file path" you can recursively change attributes of directories and their contents

77. "sudo chattr -i /file path" To retrive the file to mutable.

78. "lslogins -u" Shows information about known users in the system, the -u flag only displays user accounts.

79. "users" Shows the usernames of all users currently logged on the system like so.

* Monitoring the server’s and services using Sensu and Load Balancing tool HAProxy.
* Escalated mail to respective team if any critical alert’s observed in sensu.
* If any down alert’s camed from Pingdom, escalated that mail’s to systems team and follow up until it’s resolved.
* If any services are down in sensu, browsing that service’s by pointing that locally. and If the service is not browsing well, then informed to respective team.
* Informed to systems team if the server go’s down multiple times within a day.
* Informed to on call person’s quickly regarding issues.
* Got appreciation from Mr.Kannaiah regarding quick response .

I would like to develop skills towards DevOps Engineer with scripting language’s Python and Ruby to automate some tasks. And solve some issues which are coming regularly.

Inter View Questions

1. What is Shell?

A shell is software that provides an interface for an operating system's users to provide access to the kernel's services.

1. How to check how many shells that Linux/Unix server will support?

The /etc/shells is a text file which contains the full pathnames of valid login shells.

>>cat /etc/shells

1. What are the different shell types?
2. Bash
3. **Ksh**
4. **Zsh**
5. Fish
6. How many types of comments does shell script support?

Two types of comments will supported by the shell.

1. Single line comment. Using #
2. Multi line comment By using << COMMENT Name

Comment1.

Comment2.

.

.

COMMENT Name

(OR)

: ‘

Comment1.

Comment2.

‘

1. What is command line arguments?

The UNIX shell is used to **run** commands, and it allows users to pass **run** time arguments to these commands. These arguments, also known as command line parameters that allows the users to either control the flow of the command or to specify the input data for the command.

1. What is the difference between $\* and $@?

Both the $\* and $@ behave the same. When we pass the command line argument in double quotes ("c d"), the $\* does not consider them as a single entity, and splits them. However, the $@ considers them as a single entity

1. Write down the syntax for if condition?

The basic syntax of an *if … then* statement is like this:

if *<condition>*;

then

*<commands>*

fi

1. Write down the syntax for for loop?

for var in word1 word2 ... wordN

do

Statement(s) to be executed for every word.

done

1. Write down the syntax for function and write one function and call that function?

To declare a function, simply use the following syntax

function\_name () {

list of commands

}

EX:-

Define your function here

Hello () {

echo "Hello World"

}

# Invoke your function

Hello

1. What is the difference between > and >> and < and what is the standard output and standard error codes?

“>” redirects output to a file, overwriting the file.

“>>” redirects output to a file appending the redirected output at the end.

"<" symbols are used when inputting data from a file into a command.

1. How to display one variable value?

By using echo command and $ symbol before variable name.

Ex:- echo $variable-name

1. How many types of variables in Shell scripting?

system defined variables and user defined variables

1. Write System defined variables?

| **Name** | **Meaning** |
| --- | --- |
| $HOME | Absolute pathname to your home directory. |
| $PATH | List of directories to search for commands. |
| $USER | Your user name. |
| $SHELL | Absolute pathname of your login shell. |
| $TERM | The type of your terminal. |

1. What is String?
2. How to find the length of the given string?

Using ${#string} to get the length of $string