**LINUX SCRIPTING**

To create a new Virtual Machine Using vagrant.

1. vagrant box add name/machine
2. vagrant init jasonc/centos7
3. vagrant up
4. vagrant status
5. vagrant halt
6. vagrant ssh # to SSH into the virtual machine

**Shell Script:**

1. Start with Shebang - #!
2. # -> Sharp
3. ! -> Bang
4. First Line gives the path of the script - **#!/bin/”bash” – Since it is a shell Script**
5. To execute the script use /bin/bash <Filename>
6. Store the Shell Script using .sh
7. To run a script ->./<FILENAME>
8. Permissions explanation
9. '-rw-r--r-- 1 vagrant vagrant File name'
   1. For the owner
      1. r - Read (4)
      2. w - Write (2)
      3. x - Executable (1)
   2. For the Vagrant Group Members
   3. For Everyone
10. For Giving Permissions to the Files
    1. Executable -> **chmod 755 <FILENAME>**

Vagrant Password: Nirofall@18

|  |  |
| --- | --- |
| Linux Commands | Explanation |
| **ls -l** | Lists the permission for the folder or file inside the directory |
| **pwd** | Present Working Directory |
| **cd .** | Go back by one directory |
| **cd ..** | Go back to the root directory |
| **./** | This is how we can go to another directory |
| **type -a <SHELL Commands>** | Will display if it is a Shell Buitlin. |
| **help echo** | Will tells Shell Built In commands |
| **man uptime** | Will tell the manual of commands that are not Shell Built In |
| VIM Commands | * /<search variable> -> Press n to go to next * ?<search variable> -> Reverse Search |
| whoami | Prints the User Name |
| id -un | Prints the User Name |
| su | Super User will prompt to type the Password |
| sudo | This command will execute the script as a root user |
| echo “${?}” | This is use to check the last executed exit status |
| help <command> | less | It will lists the various built-in commands. | -> We pipe it. |
| Man useradd | This is command is used to create User Account |
| read | Use to read input from the user |
| ps -ef | Process Command to show every process |
| sudo useradd <Name> | Will Create another user |
| sudo su - <Name> | Will Login to the user account |
|  |  |
| su - <Name> | Super User will allow to form a command shell |
| Sudo su – root | Switch to the root user |
| sudo userdel <user name> | Deletes the created user. |
| Id -u <name> | Displays the UID of the username |
| Id -un <Name> | Displays the name of the user |
| Cat /etc/passwd | Displays the list of users available |
| | | Pipe takes the output from the previous command and give it as an input to the next (Eg: date +%s | sha256sum) |
| Head (-n, -c) (file) | Head command will print the characters or number of line of the file |
| Shuf | Generates Random Permutations |
| FOLD | To display the lines  **echo "${S}" | fold -w1 | shuf |head -c1** |
| Hash (Shell Builtin) | To remove the stored or hashed memory locations.   * **To remove : hash -r** |
| basename (path) | Returns the file name:  Basename /vagrant/shell.sh -> OP: shell.sh |
| dirname (path) | Returns the directory name:  Basename /vagrant/shell.sh -> OP: /vagrant |
| which | To ask what is there in the path.   * **which head** |
| ${0} | Tells the last executed command. i.e stores and displays |
| ${#} | Tells the number of parameters |
| !(filename) | To execute the last supplied command |
| true | Returns an exit status of 0 |
| sleep | Used to delay for a specified amount of time.   * sleep 1 -> Will wait for 1 sec. |
| shift | It shifts the positional parameters; |
| ${@} | Stores all the positional arguments after the first argument   * First use shift. |
| tail -3 /etc/passwd | Check to see if users were created or not. |
| head -n1 | Displays the first line from the file |
| > | Redirects output from one file to another file – Redirection Operator   * echo 'Niroman is learning Bash Scripting' > niro\_leanr.txt |
| < | Redirects as an input to the file. |
| >> | Use to append data to the file.   * echo 'Niroman is learning Bash Scripting' >> niro\_leanr.txt |
| File Descriptors | * FD0 – Standard Input * FD1 – Standard Output * FD2 – Standard Error |
| &> | In the command line store the I/O and error in a file. |
| |& | Pipes it in the output file. |
| 1> | Redirects the output to a file |
| 2> | Redirects the error to a file |
| 1>&2 | * “&” after a redirection operator will send it to a file. |
| dd | Delete in VIM |
| p | Paste in VIM |
| u | Undo in VIM |
| gg | Go to the First line of the text editor |
| Ctrl + v | Enter Visual Block |
| date | Will display the date |
| /var/log/messages | Stores the System Log Messages |
| getopts |  |
| logger | Stores the output in the Log messages   * logger -t(tagging) filename “Message” |
| locate | To find the files |
| find |  |
| touch | Creates a file if not found. |
| sudo updatedb | To update the new files. Like Refresh |
| sudo !! | Executes the last executed command with sudo permission. |
| find | Too find a particular file.   * find / -name userdel 2>/dev/null -> This redirects error messages to null file and will not display in screen. |
| Tar  (Tap Archives) | * tar -cf niro.tar -> Creates a tar * tar -cvf -> Creates a tarr and dispalys it. * Tar -tf -> Displays the files in the tar * tar -xvf -> To extract the files. |
| gzip | * gzip niro.tar -> It will gzip it and add .gz to the tar file. |
| tar + gzip | * tar -zcvf niro.tar /home/nito * tar -zxvf niro.tar.gz |
| !$ | * Executes the last executed file. |
| chage | * sudo chage -E 0 (Username) -> Expires an account * sudo chage -E 1 (Username) -> Unlock an account |
| usermod | To change the user details. |
| cut – To cut the given arguments and display it | * cut -c 1 /etc/passwd -> Displays the first character of the files * cut -c 4-7 /etc/passwd -> Displays the fcharacter of the files in that range * cut -c -4 /etc/passwd -> Displays the first four character of the files * cut -c 1,3,5 /etc/passwd -> Displays characters in 1,3,5 * cut -b 1 /etc/passwd -> Displays bytes * echo “Niro” | cut -c 1 -> We can also send as a standard input. * cut -f 1 /etc/passwd -> Displays bytes * echo -e 'one\ttwo\tthree' | cut -f 3 -> Use tab to specify it. * echo 'one,two' | cut -d ',' -f 1 -> Use a delimiter to specify it. |
| grep -> Used to search | * grep ‘firrst’ people.csv * ^ -> Carrrot Sign * For perfect match **grep ‘^first,second$’ people.csv** * For inverting the search **grep -v ‘^first,second$’ people.csv** * grep -v '^first,second$' people.csv | cut -d ',' -f 1 * netstat -nutl | grep -v '^Active' | grep -v '^Proto' * netstat -nutl | grep -Ev '^Active'| '^Proto' * grep ‘:’ * grep -c |
| awk | * awk -F 'DATA:' '{print $2}' people.dat * netstat -nutl | grep ':' | awk '{print $4}' | awk -F ':' '{print $NF}' * print NF -> Prints the last value after the delimiter. |
| netstat | netstat -nutl or netstat -4nutl   * N – Display Number * U – Display UDP * T – Display TCP * L – Display live ports |
| sort | * sort -r -> Sorts the files in reverse order * sort -nr -> Sort Numbers in reverse order with a numerical value. * sort -nu -> Sorts unique values numerically * cut -d ':' -f 3 /etc/passwd | sort -nr * sudo du -h /var | sort -n * cat /etc/passwd | sort -t ':' -k 3 -n -> Sort uses white space as |
| du | * Disk Usage * du -h -> With human readable numbers. |
| uniq | To display unique values.   * netstat -nutl | grep ':' | awk '{print $4}' | awk -F ':' '{print $NF}' | sort -n | uniq -c |
| wc | Word Count   * wc -w -> Total Words * wc -c -> Total Characters * wc -l -> Total Number of lines. * wc -l people.csv * grep /etc/passwd | wc -l |
| >&2 | echo "Cannot open ${LOG\_FILE}" >&2   * Redirects the output to a null file |
| $? | echo $?   * Last executed exit status is printed. |
| sed | Stream Editor   * sed ‘s/search-pattern/replace-string/flags’ filename. * sed 's/is learning/learnt/' niro\_leanr.txt -> Replaces the text in the file temporarily. * sed 's/is learning/learnt/i' niro\_leanr.txt * ‘I’ Flag -> I used to change even if it is in lower or upper case * ‘g’ flag -> It is used to change all the occurrences. * sed **-i.bak** 's/is learning/learnt/g' niro\_leanr.txt -> Creates a backup file * sed ‘s/is learning/learnt/gw niro.txt' niro\_leanr.txt -> Creates a backup file and displays it in the terminal. * Sed ‘/then/d’ -> This is the delimiter used in sed. * sed ‘/^#/d’ -> This is the delimiter used in sed. To match all the lines that begin with ‘#’. * sed ‘/^#/d ; /^$(To remove white spaces)/d’ File name * sed -f filename1 filename2 -> Takes the input from filename 1 and executes in filename 2. |
| ${0} | Name of the script always. |
|  |  |
|  |  |
|  |  |
|  |  |