**Linux commands for accessing the virtual machine present in server using linux commands**

1. $ whoami :- command used to check the current user logged in to the server
2. $ date :- used to current server date and time
3. $ cal :- used to get the calender for that particular month
4. $ cal 2025 :- used to get the calender for that particular year
5. $ pwd :- command used to get the present working directory
6. $ mkdir <foldername> :- used to create a folder/directory
7. $ ls :- listing the content of that particular directory sorted in alphabetical order
8. $ ls -r :- listing the content of that particular directory sorted in descending alphabetical order or reverse sorting
9. $ ls –l :- longlisting the folder contents based on current date and time
10. $ ls –lr :- longlisting the file names in reverse order
11. $ clear :- command is used for clearing the screen
12. $ rmdir <foldername> :- is used for removing empty directory
13. $ ls –la :- used to get the hidden files in a particular folder
14. $ cd /home/ec2-user/a :- command is used for navigating to another directory folder
15. $ cd .. / cd ~ :- is used for going one folder back
16. $ touch <filename.txt> :- command used for creating a text file
17. $ rm –rf <folder name>:- recursively forcibly remove the folder even when the content is there inside it
18. $ exit :- command used for logout

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

1. $ mv <existing filename.txt> <expected filename.txt> command used for renaming the file name
2. $ mv f0.text(current file) /home/ec2-user/a (location and the folder name it has to be moved)
3. $ mv present location(file name) to location where it has to moved
4. $ cat > f4.txt enter ctr + C :- creating a file
5. $ cat >> f1.txt

Hello world

Hi

Used for writing contents into the file

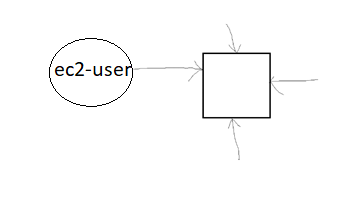
1. $ cat f1.txt :- command used for reading the contents of the file
2. $ cat –n f1.txt :- used for reading the contents of the file including the line numbers
3. $ cat >> f6.txt :- If a file is not present it is used for creating a file as well
4. $ touch f6.txt :- just updates the timestamp of the file it doesn’t alter the content of the file
5. $ tac f1.txt :- is used to read the file in reverse order
6. $ cp f1.txt(source file) f2.txt(destination file) :- used to copy the content from one file to another file
7. $ cat f1.txt f2.txt > f3.txt :- used to copy the contents of file f1, f2 to file f3.txt
8. $ head f1.txt :- used to read the first 10 lines of f1.txt file
9. $ head –n 5 f3.txt :- to read the first 5 lines of f3.txt file
10. $ grep “He” f1.txt :- global regular expression is used to find specific pattern or a word in a particular file
11. $ grep –i “He” f1.txt :- performs search operation based on the case insensitivity comparison
12. $ grep –n –i “He” f1.txt :- here -n is for line number and -i for ignoring the case sensitivity order of i & n doesn’t matter
13. $ grep -v -i -n “He” f1.txt :- here to search other content apart from “He” is used
14. $ grep -i ‘He’ \* :- this command is used for searching from all the files in that particular folder
15. $ grep -i -v ‘He’ \* :- apart from ‘He’ all other data is found and fetched on the screen
16. $ grep –i –v –n ‘He’ \* :- here including the line number if the content has to fetched then this is mainly used
17. $ tail f3.txt | grep –i ‘He’ :- to find the content in the last 10 lines of the file
18. $ tail –n 3 f3.txt | grep –i ‘He’ :- to find the content in the last 3 lines here we are using pipelines for storing multiple commands
19. $ cat f1.txt | nl :- command is used for reading contents of the file in order
20. $ tac f1.txt | nl :- command is used for reading contents of the file in reverse order
21. $ head –n 5 f3.txt | grep -i ‘He’ :- to find the content in first 5 lines of f3.txt
22. $ wc f3.txt :-used for getting the line, word & characters count in that particular file
23. $ wc –l f3.txt :- used to get only the number of lines present in the file
24. $ wc –w f3.txt :- used to get the count of words in the file
25. $ wc –m f3.txt :- gives us the number of characters present in that particular file
26. Use the cp command to copy the contents from file1.txt into the file2.txt
27. Use head to display the first 15 lines of logfile.log
28. Use grep to search all the occurrences of the word “error” in logfile.log
29. Use tail to show the last 20 lines of access.log
30. Use WC to count the number of words in the sample.txt

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

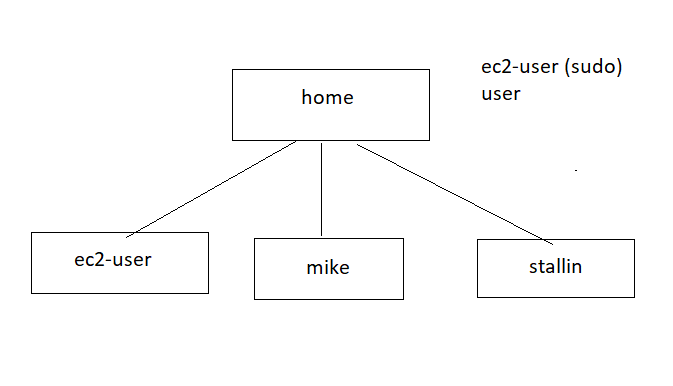
1. $ vi f3.txt :- Is a Visual Editor for Linux Os just like Notepad editor for Windows Os
2. $ vi f3.txt :- after entering the text content just type I for insert mode after entering the text press esc key and for saving and quit use :wq command
3. :w :- for just saving
4. :q :- for quit option
5. :q! :- for quit without saving
6. $ sed :- here sed stands from Stream Editor just like visual editor here stream editor can do text processing (substitution, deletion, printing and insertion). Performing operations without opening them
7. $ sed ‘s\Hello\World\’ f1.txt :- this command is used for replacing/substituting all the first occurences of Hello with World
8. $ sed ‘s\Hello\World\2’ f1.txt :- is used for replacing all the second instances of Hello in a line in the f1.txt file
9. $ sed ‘s\world\hello\g’ f1.txt :- is used to replace or substitute all the occurrences of world to hello
10. $ sed -i 's/world/kushal/g' f1.txt :- is used for permanently changing the content
11. $ sed -i '1d' f1.txt :- command is used for deleting the first line of the text file
12. $ sed -i ‘$d’ f1.txt :- command is used for deleting the last line
13. $ sed -i ‘1,5d’ f1.txt :- command is used for deleting range of lines
14. $ sed -i ‘1, $d’ f1.txt :- command for deleting all the lines in a text files

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

1. Difference b/w vi editor & stream editor is for vi editor we need to open the file and make changes whereas in stream editor without opening we can edit, substitute & delete the changes
2. $ sed –n ‘1,3p’ f1.txt :- command is used for printing the contents of the file(-n reading the file)
3. $ sed -i '2i\hello world' f1.txt :- command is used for adding content in the middle of the file
4. $ sed -i '$a\last line' f1.txt :- command is used for adding content at the last line



* If this is a server multiple users are getting connected to the server as EC2-User with admin (sudo) In linux there is no admin it’s called sudo user super user do(sudo).
* If a Linux server is created in AWS using EC2 it’s going to create a EC2-User sudo account by default
* By using aws we can create multiple users for ec2 by default there is linux vm created by user which will have a home directory and user will be ec2-user \home\ec2-user
* If another user is created called mike all his work will be stored in \home\mike directory under home folder
* If a third User is created Stallin under \home\stallin directory get’s created all these users are independent of each other



1. $ sudo useradd mike :- here admin is sudo user ec2-user responsible for adding user
2. $ sudo passwd mike :- here for creating password of user mike here after that need to give new password and retype the new password
3. $ grep mike /etc/passwd :- here grep command is used for searching a particular command all users can be seen in etc directory and under sub directory passwd under home all these folders are created
4. $ sudo useradd stallin
5. $ sudo passwd stallin (add password and confirm password)
6. $ grep stallin /etc/passwd :- grep command is for searching the stallin user
7. $ su mike :- here su is switch user and we must give the password for mike
8. $ pwd :- after logging in as mike it is still in { /home/ec2-user }
9. $ ls :- it shows cannot open directory ‘.’ : permission denied
10. $ cd ~ :- here it returns to the mike directory
11. $ ls
12. $ touch f1.txt
13. $ ls = touch f1.txt
14. $ exit :- back to the root account to sudo user /home/ec2-user
15. $ cd ..
16. $ ls = ec2-user mike stallin
17. $ su stallin :- for switching to stallin after enter hit password
18. $ sudo userdel mike --remove :- Is used for deleting the account in linux os
19. $ cd ..
20. $ ls = ec2-user stallin
21. $ cd ~
22. $ sudo useradd alice
23. $ sudo passwd alice :- here it asks for password and retype the same pwd for confirmation post that If want to update the password
24. $ sudo passwd alice :- then enter the new pwd for updating the password
25. Creating user groups putting person while creating a group manage the users by putting a permission on that user groups automatically users added to that user groups has all the permissions.

**[Commands for user groups]**

1. $ cat /etc/group :- all user groups are displayed
2. $ sudo groupadd <groupname> :- command for adding the group name or creating a user group

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

1. $ sudo useradd stallin
2. $ sudo passwd stallin
3. $ su stallin

The sudoers file controls which user can execute communicate commands as superusers

1. $ sudo visudo //this is for opening up the sudoers file

Scroll down here I want to make stallin as sudo user (admin user)

stallin ALL = (ALL:ALL) ALL //this command is used for giving all the access to the stallin to have the admin access here both ec2-user, stallin users has sudo user access details present in sudoers file

1. For exit(quit) : - ctrl + x + y + enter
2. Here permission for stallin has been set to a super user

Enable password Authentication

By Default all the users password authentication is not there

1. $ sudo vi /etc/ssh/sshd\_config
2. Scroll down search for (PasswordAuthentication yes)
3. Esc and then :wq
4. After modification of sshd\_config after setting PasswordAuthentication to yes restart the SSH Service
5. $ exit
6. Connect from ec2 to current device through ec2 commands copy paste those 2 lines in EC2 instance
7. $ sudo systemctl restart sshd //make sure that after the modifications everything has to be restarted
8. $ ssh mike@Ip4adress (copy it from ec2 dasboard after connecting to server).
9. It asks for password If these are given to others anyone can access this using these credentials
10. $ pwd
11. $ ssh stallin@Ipaddress
12. $ clear
13. $ id mike //each user will have a unique identifier associated with him
14. $ id ec2-user

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

1. $ sudo groupadd developers //here if & only if user is a sudo user he can create a group

Add Users to the group “developers”

1. $ sudo usermod –aG developers alice //here after creating a group called developers adding the user alice to developers group
2. $ sudo useradd alex
3. $ sudo passwd alex
4. $ clear
5. $ sudo usermod –aG developers alex :- append –a for Group G is the exact meaning

Comma/nd for displaying the number of users in the user groups

1. $ sudo lid –g developers <groupname> :- here the groupname is developers

Will get the list of all the users present in the user group developers

1. $ sudo lid –g developers :- command for checking the list of users in the user group

//mike (uid = 1001)

//stallin (uid = 1002)

Command for removing/deleting a particular user from that group

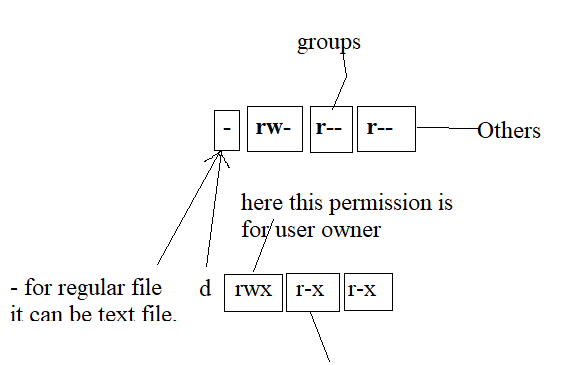
1. $ sudo gpasswd –d mike developer :- -d for remove or delete the user mike from the user group developer
2. $ sudo lid –g developers
3. $ id mike :- command is used for checking which all the groups mike belong to

Delete the user group

1. $ sudo groupdel <groupname> :- is used for deleting the group
2. $ sudo groupmod –n team team1 :- command used for modifying the group name here the first is the team name that has to be renamed and team1 is the existing group

Permissions in groups

1. $ touch file.txt
2. $ ls –l



Here first 3 characters :- User/Owner permissions

Middle 3 characters :- Group permissions

Last 3 characters :- Others permissions

In symbolic permissions are modified using operators:

+ : Add a permission, - Remove a permission, = set specific permission/overriding

1. $ chmod u+x f1.txt :- u represents execute permission for user
2. $ chmod g+w f1.txt :- g represents execute permission for group
3. $ chmod o+x f1.txt :- o represents execute permission for others
4. $ chmod u+x f1.txt :- provides the execute permission for user
5. $ chmod g+x f1.txt :- provides the execute permission for the file
6. $ chmod g+w f1.tx :- provides the write permission for the file present in group
7. $ ls –l
8. $ chmod g-w f1.txt :- for removing the write permission from the file
9. $ chmod o+w f1.txt :- for removing the o + w f1.txt
10. $ chmod
11. Permissions with numeric format

Here Read(r) = 4, (w) = 2, (x) = 1

1. $ chmod 111 file1.txt :- command is owner, groups, others has only only for execute permission
2. $ chmod 444 file1.txt :- command is owner, groups, others has only for read permission
3. $ ls -l
4. $ chmod 222 file1.txt :- command is owner, groups, others only for write permission
5. $ ls –l
6. $ chmod 777 file1.txt :- command is owner, groups, others has all permissions
7. $ ls –l
8. $ chmod 741 file1.txt :- user group can read, write & execute, group has permission only for read and others has the permission only to execute
9. $ chmod 742 file1.txt
10. $ls –l
11. $ chmod 000 file1.txt