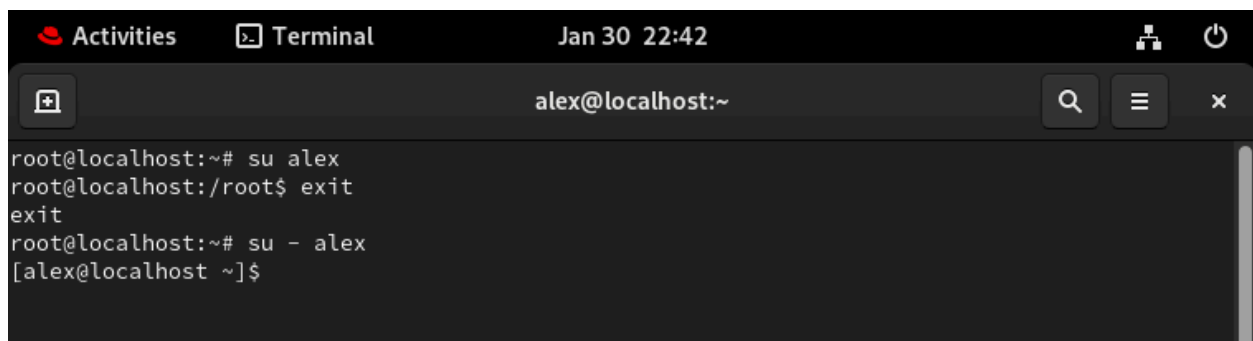


1. Create some users: o Named "alex" with its home directory at /home/user1 and give password "pass1". o Named "brew" with its home directory at /mnt/user 2 and give password "pass2". o Named "nora" without its home directory o Named "panny" with custom UID 2112, and assign password "pass-4" o Named 'texas' without using the useradd or adduser commands. *(Hint: Make changes in the 7 user configuration files)

```
root@localhost:~# useradd -m -d /home/user1 alex
root@localhost:~# passws alex
bash: passws: command not found
root@localhost:~# passwd alex
Changing password for user alex.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
root@localhost:~# useradd -m -d /mnt/user2 brew
root@localhost:~# passwd brew
Changing password for user brew.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
root@localhost:~# useradd -M nora
root@localhost:~# useradd -u 2112 panny
root@localhost:~# passwd panny
Changing password for user panny.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
root@localhost:~# nano /etc/passwd
root@localhost:~# vim /etc/passwd
root@localhost:~# passwd texas
Changing password for user texas.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
root@localhost:~#
```

2. Log in as user alex using the su and su - commands, and explain their differences.



The screenshot shows a terminal window with the title bar "Activities Terminal Jan 30 22:42". The terminal prompt is "alex@localhost:~". The user has executed the following commands:

```
root@localhost:~# su alex
root@localhost:/root$ exit
exit
root@localhost:~# su - alex
[alex@localhost ~]$
```

3. Set a password policy for all above users with the following requirements:
- o The maximum password age should be 30 days, and the minimum password age should be 10 days.
 - o Set the password expiry date for all users to December 31, 2025.

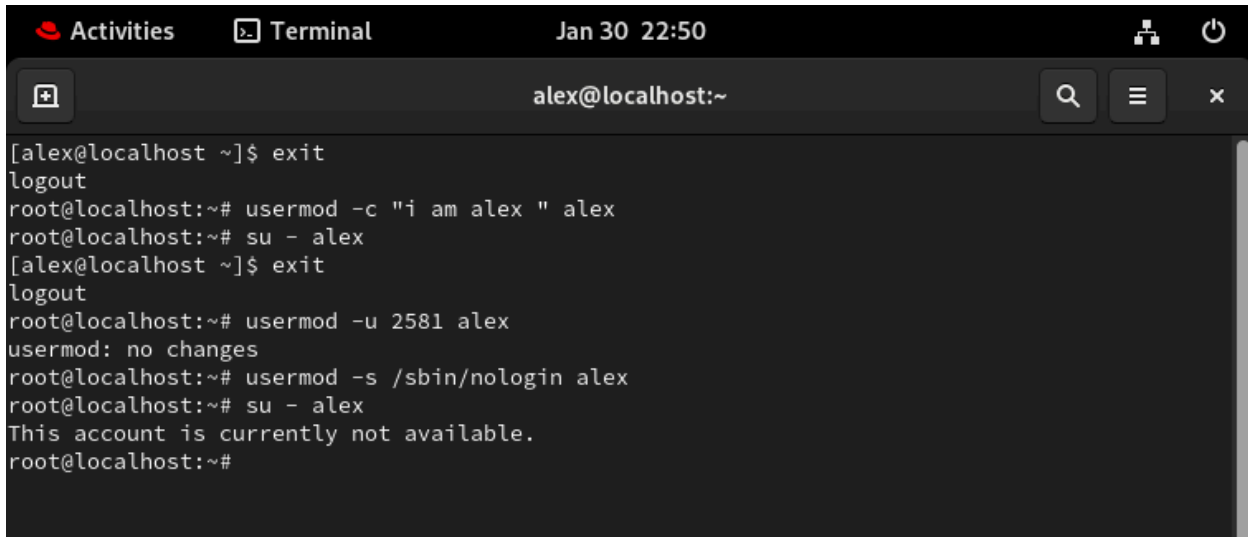


A terminal window titled 'Terminal' with a timestamp of 'Jan 30 22:44'. The window shows a user 'alex@localhost' attempting to run 'chage alex', which results in a 'Permission denied' error. The user then exits and logs out. The root user then runs 'chage alex' and sets the following password policy: Minimum Password Age [0]: 30, Maximum Password Age [99999]: 10, Last Password Change (YYYY-MM-DD) [2025-01-30]: 2025-12-31, Password Expiration Warning [7]: 5, Password Inactive [-1]: 5, and Account Expiration Date (YYYY-MM-DD) [-1]: 2026-1-1. Finally, the root user runs 'chage -M 30 -m 10 -E 2025-12-31 brew'.

```
[alex@localhost ~]$ chage alex
chage: Permission denied.
[alex@localhost ~]$ exit
logout
root@localhost:~# chage alex
Changing the aging information for alex
Enter the new value, or press ENTER for the default

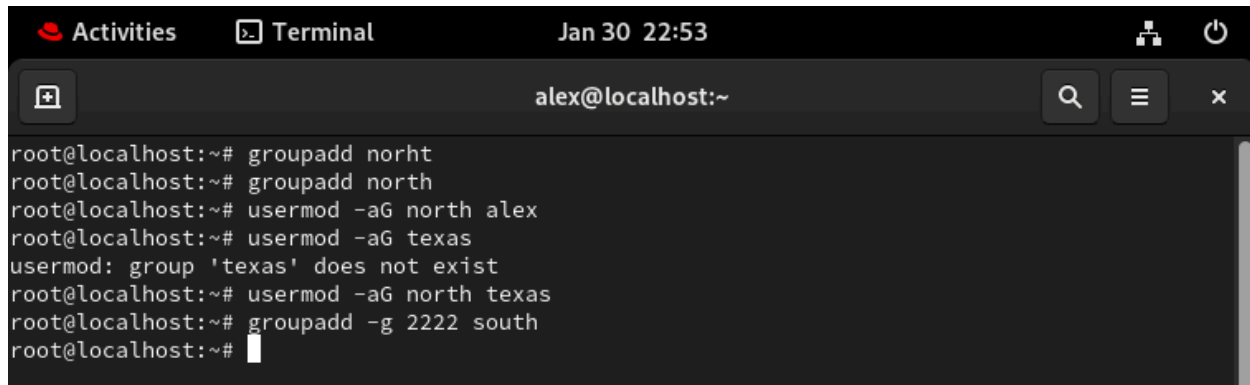
        Minimum Password Age [0]: 30
        Maximum Password Age [99999]: 10
        Last Password Change (YYYY-MM-DD) [2025-01-30]: 2025-12-31
        Password Expiration Warning [7]: 5
        Password Inactive [-1]: 5
        Account Expiration Date (YYYY-MM-DD) [-1]: 2026-1-1
root@localhost:~# chage -M 30 -m 10 -E 2025-12-31 brew
root@localhost:~#
```

4. Modify the user "alex":
- Add a comment: "I am alex"
 - Change the UID to 2581
 - Change the shell to "nologin"



A terminal window titled 'Terminal' with a timestamp of 'Jan 30 22:50'. The window shows a user 'alex@localhost' exiting and logging out. The root user then runs 'usermod -c "i am alex " alex', 'su - alex', and 'exit'. The root user then runs 'usermod -u 2581 alex', which results in 'usermod: no changes'. Finally, the root user runs 'usermod -s /sbin/nologin alex', 'su - alex', and 'exit', which results in 'This account is currently not available.'.

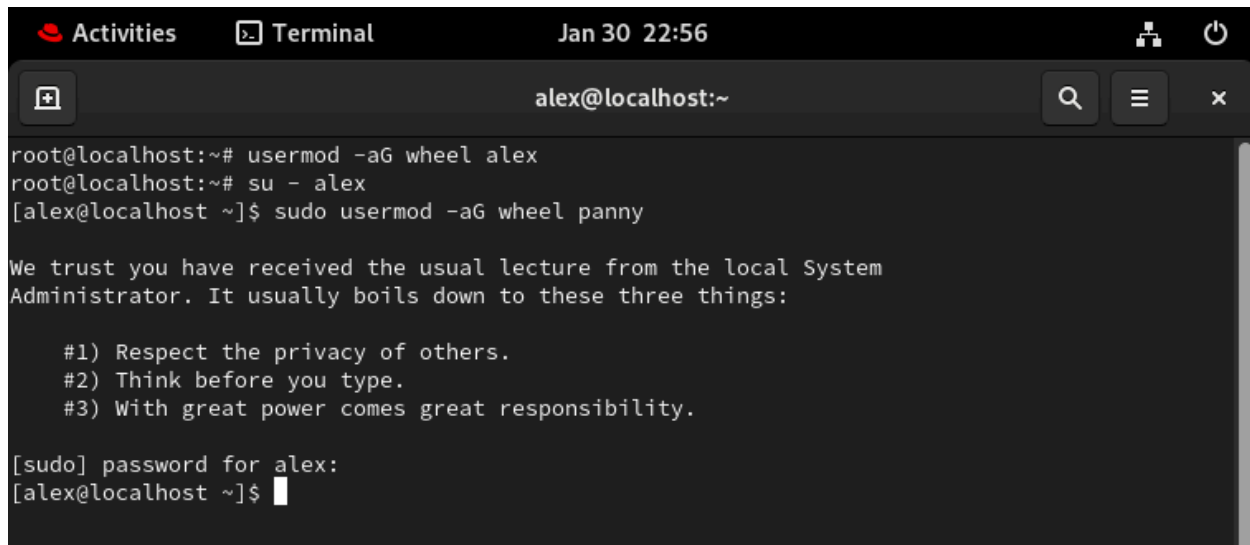
```
[alex@localhost ~]$ exit
logout
root@localhost:~# usermod -c "i am alex " alex
root@localhost:~# su - alex
[alex@localhost ~]$ exit
logout
root@localhost:~# usermod -u 2581 alex
usermod: no changes
root@localhost:~# usermod -s /sbin/nologin alex
root@localhost:~# su - alex
This account is currently not available.
root@localhost:~#
```



A terminal window titled 'Terminal' with a timestamp of 'Jan 30 22:53'. The window shows a series of commands being executed as root on a local host. The commands are: `groupadd norht`, `groupadd north`, `usermod -aG north alex`, `usermod -aG texas`, `usermod -aG north texas`, and `groupadd -g 2222 south`. The output for the `usermod -aG texas` command is 'usermod: group 'texas' does not exist'. The prompt returns to root@localhost:~# after each command.

```
root@localhost:~# groupadd norht
root@localhost:~# groupadd north
root@localhost:~# usermod -aG north alex
root@localhost:~# usermod -aG texas
usermod: group 'texas' does not exist
root@localhost:~# usermod -aG north texas
root@localhost:~# groupadd -g 2222 south
root@localhost:~#
```

6. Grant user Alex administrative privileges through the wheel group so that Alex can add Panny to the admin group without requiring root access.



A terminal window titled 'Terminal' with a timestamp of 'Jan 30 22:56'. The window shows commands being executed as root: `usermod -aG wheel alex` and `su - alex`. Then, as alex, the command `sudo usermod -aG wheel panny` is executed. The terminal displays the standard sudo warning message: 'We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:'. It lists three points: '#1) Respect the privacy of others.', '#2) Think before you type.', and '#3) With great power comes great responsibility.'. It then prompts for the password for alex, which is entered successfully, returning the prompt to alex@localhost ~\$.

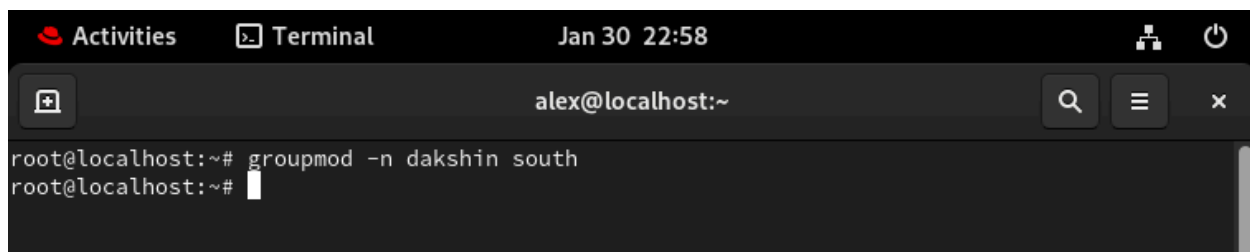
```
root@localhost:~# usermod -aG wheel alex
root@localhost:~# su - alex
[alex@localhost ~]$ sudo usermod -aG wheel panny

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for alex:
[alex@localhost ~]$
```

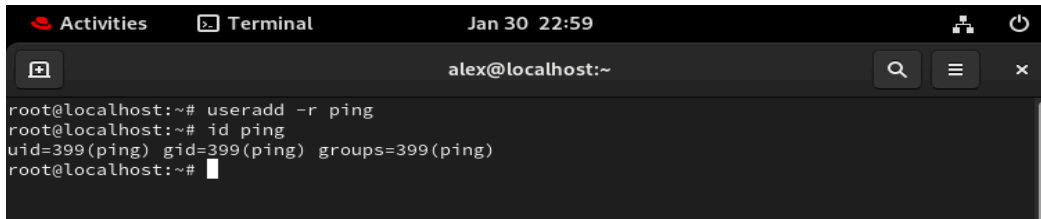
7. Change the group name from “south” to “dakshin”.



A terminal window titled 'Terminal' with a timestamp of 'Jan 30 22:58'. The window shows a single command being executed as root: `groupmod -n dakshin south`. The prompt returns to root@localhost:~# after the command is executed.

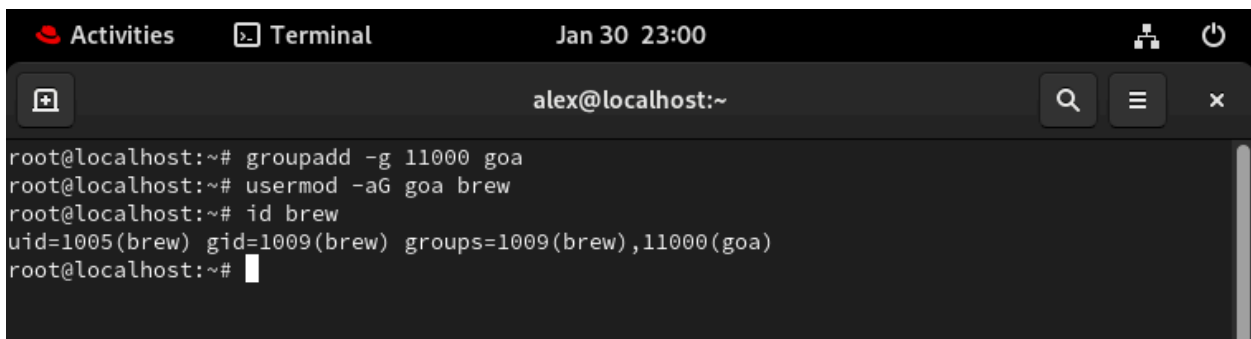
```
root@localhost:~# groupmod -n dakshin south
root@localhost:~#
```

8. Create a system user named “ping” and check its UID.

A terminal window titled 'Terminal' with the date 'Jan 30 22:59' and the user 'alex@localhost:~'. The terminal shows the following commands and output:

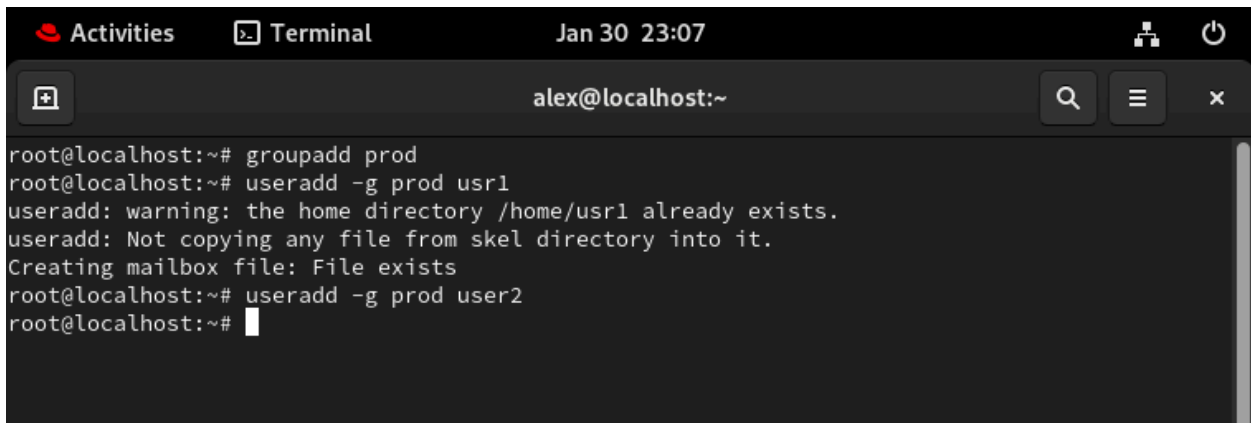
```
root@localhost:~# useradd -r ping
root@localhost:~# id ping
uid=399(ping) gid=399(ping) groups=399(ping)
root@localhost:~#
```

9. Create a group named goa with GID 11000. Set this group as the supplementary group for “brew”

A terminal window titled 'Terminal' with the date 'Jan 30 23:00' and the user 'alex@localhost:~'. The terminal shows the following commands and output:

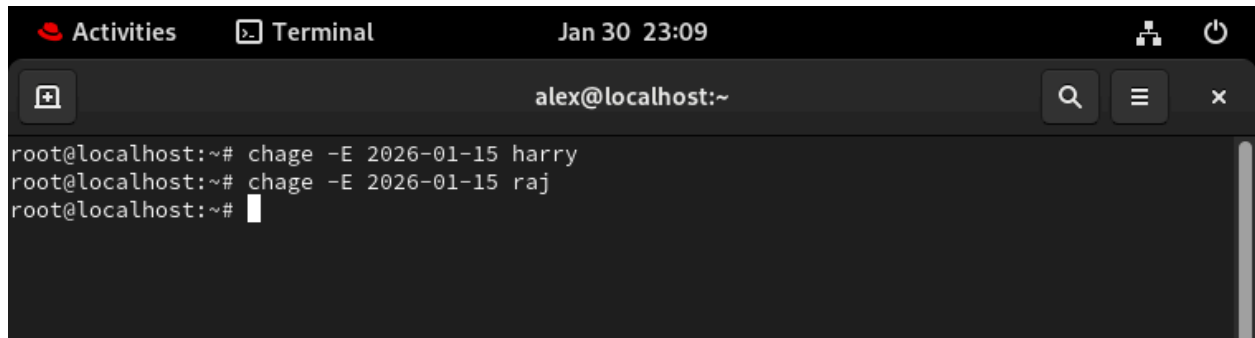
```
root@localhost:~# groupadd -g 11000 goa
root@localhost:~# usermod -aG goa brew
root@localhost:~# id brew
uid=1005(brew) gid=1009(brew) groups=1009(brew),11000(goa)
root@localhost:~#
```

10. Create a group named “prod”. Then, create two users, user2 and user1, and set both the user’s primary group to prod.

A terminal window titled 'Terminal' with the date 'Jan 30 23:07' and the user 'alex@localhost:~'. The terminal shows the following commands and output:

```
root@localhost:~# groupadd prod
root@localhost:~# useradd -g prod usr1
useradd: warning: the home directory /home/usr1 already exists.
useradd: Not copying any file from skel directory into it.
Creating mailbox file: File exists
root@localhost:~# useradd -g prod user2
root@localhost:~#
```

11. Change the password policy for the USER3 and USER4 accounts to expire on 2026-01-15.



A screenshot of a Linux terminal window. The window title bar shows 'Activities', 'Terminal', and the date/time 'Jan 30 23:09'. The terminal prompt is 'alex@localhost:~'. The user is root, and they have executed two commands: 'chage -E 2026-01-15 harry' and 'chage -E 2026-01-15 raj'. The terminal output shows the prompts and the commands being executed.

```
root@localhost:~# chage -E 2026-01-15 harry
root@localhost:~# chage -E 2026-01-15 raj
root@localhost:~#
```

12. Configure administrative rights for all members of the Goa group to execute any command as any user.

```
Activities Terminal Jan 30 23:12 alex@localhost:~
Defaults env_keep = "COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS"
Defaults env_keep += "MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE"
Defaults env_keep += "LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES"
Defaults env_keep += "LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE"
Defaults env_keep += "LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY"

#
# Adding HOME to env_keep may enable a user to run unrestricted
# commands via sudo.
#
# Defaults env_keep += "HOME"

Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##     user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
%goa    ALL=(ALL)    ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)    ALL

## Same thing without a password
# %wheel    ALL=(ALL)    NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users    ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users    localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#include_dir /etc/sudoers.d
"/etc/sudoers" [readonly] 120L, 4356B 104,1 Bot
```

13. How would you check all failed login attempts on the system from the last 10 days? Write the command and display the output.

```
Activities Terminal Jan 30 23:13 alex@localhost:~
Defaults    env_keep += "LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE"
Defaults    env_keep += "LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY"

#
# Adding HOME to env_keep may enable a user to run unrestricted
# commands via sudo.
#
# Defaults    env_keep += "HOME"

Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
goa     ALL=(ALL)        ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)        ALL

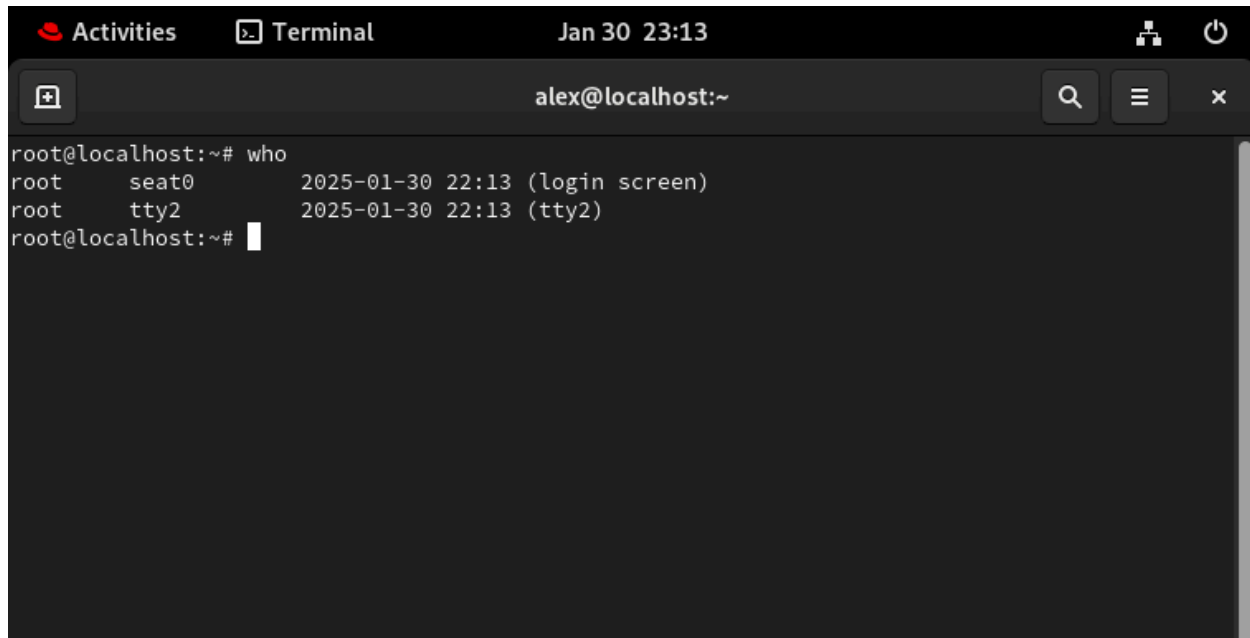
## Same thing without a password
# %wheel    ALL=(ALL)        NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users    ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users    localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
root@localhost:~# vim /etc/sudoers
root@localhost:~# vim /etc/sudoers
root@localhost:~# journalctl --since "10 days ago" | grep "Failed password"
root@localhost:~#
```

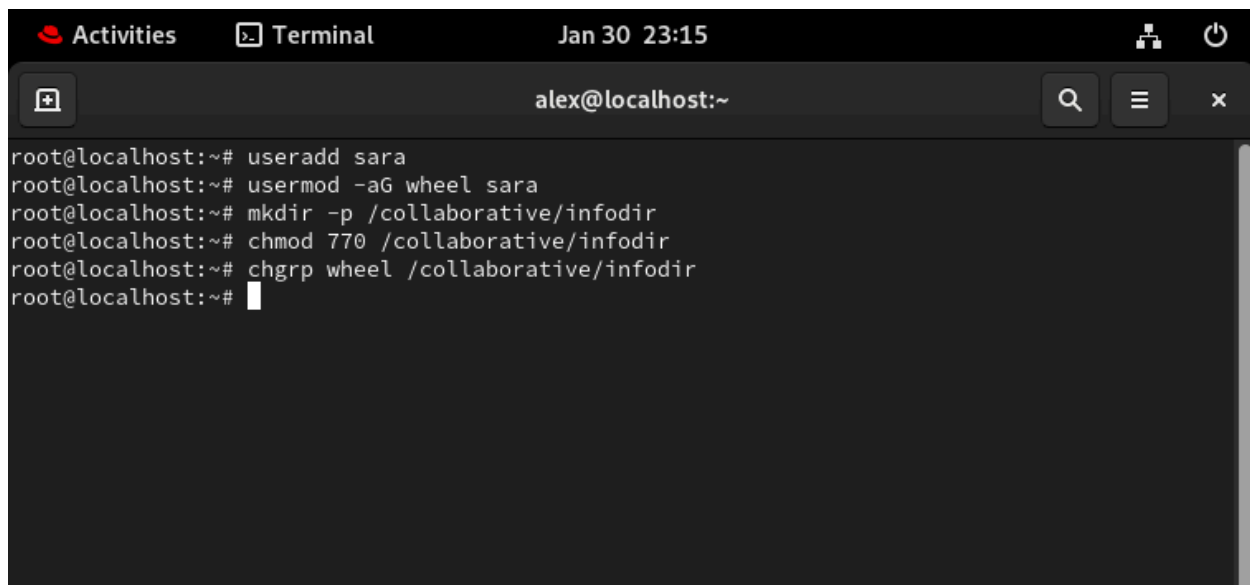
14. How would you determine how many users are currently logged into the system? Write the command to achieve this.



A terminal window titled 'Terminal' with the date and time 'Jan 30 23:13'. The window shows the command 'who' being executed as root. The output lists two sessions: one on 'seat0' and another on 'tty2', both dated '2025-01-30 22:13'. The prompt 'alex@localhost:~' is visible at the top right of the terminal area.

```
root@localhost:~# who
root    seat0          2025-01-30 22:13 (login screen)
root    tty2            2025-01-30 22:13 (tty2)
root@localhost:~#
```

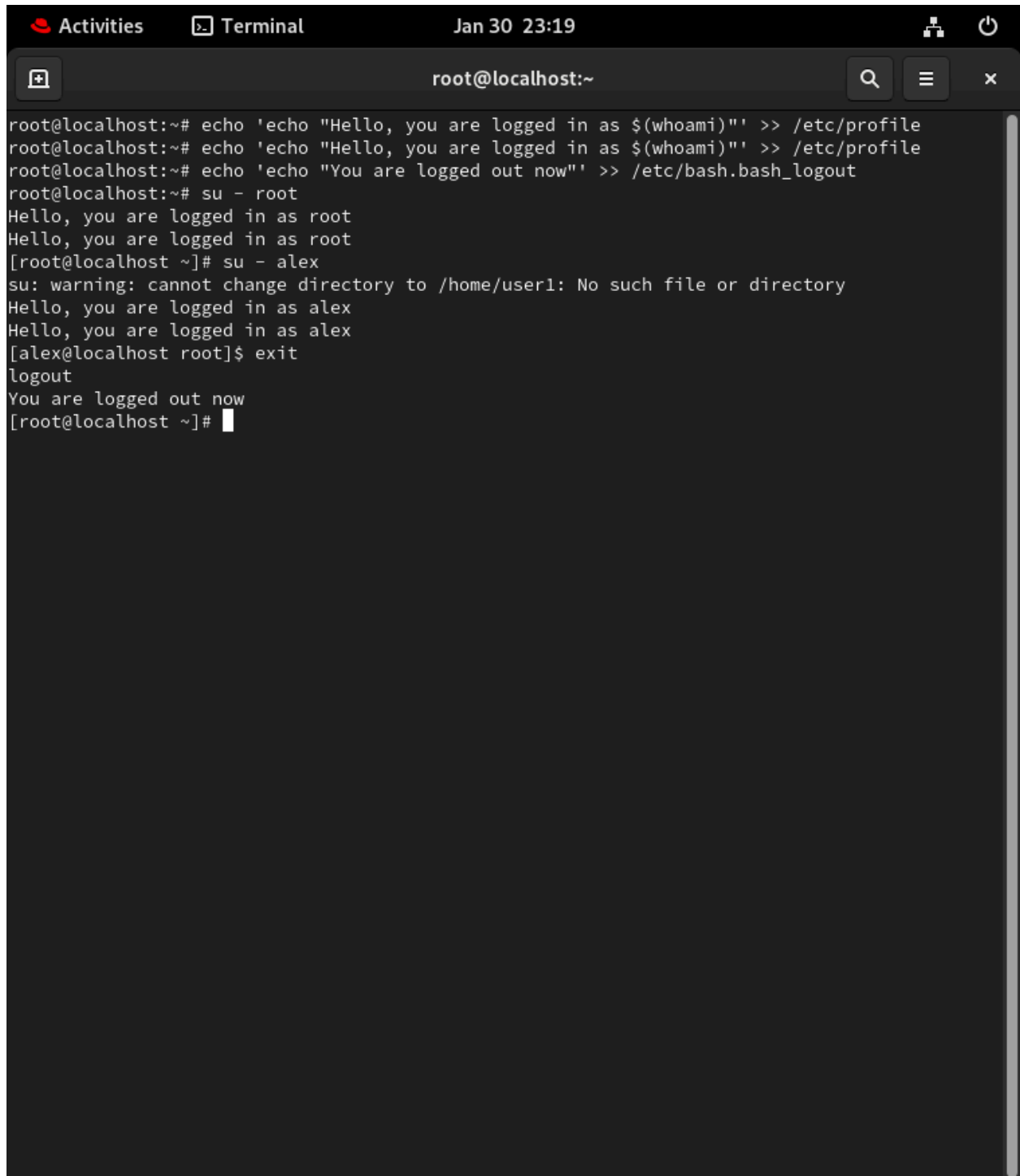
15. Add the user "sara" to the "wheel" group and create a collaborative directory /collaborative/infodir.



A terminal window titled 'Terminal' with the date and time 'Jan 30 23:15'. The window shows a series of commands being executed as root to add a user 'sara' to the 'wheel' group and create a directory. The commands are: 'useradd sara', 'usermod -aG wheel sara', 'mkdir -p /collaborative/infodir', 'chmod 770 /collaborative/infodir', and 'chgrp wheel /collaborative/infodir'. The prompt 'alex@localhost:~' is visible at the top right of the terminal area.

```
root@localhost:~# useradd sara
root@localhost:~# usermod -aG wheel sara
root@localhost:~# mkdir -p /collaborative/infodir
root@localhost:~# chmod 770 /collaborative/infodir
root@localhost:~# chgrp wheel /collaborative/infodir
root@localhost:~#
```

16. Configure login/logout messages: o When you log in with a new user, display a message: "Hello, you are logged in as USER" (where USER is replaced with the logged-in username). o When you log out, display: "You are logged out now".



A terminal window titled "Terminal" with a timestamp of "Jan 30 23:19". The window shows a series of commands and their outputs. The user is initially root@localhost. They run three echo commands to append messages to /etc/profile and /etc/bash.bash_logout. Then, they switch to root using 'su - root', which shows two "Hello, you are logged in as root" messages. Next, they switch to alex using 'su - alex', which shows a warning about the directory and two "Hello, you are logged in as alex" messages. Finally, they exit the alex session with 'exit', showing "logout" and "You are logged out now", and return to the root prompt.

```
root@localhost:~# echo 'echo "Hello, you are logged in as $(whoami)"' >> /etc/profile
root@localhost:~# echo 'echo "Hello, you are logged in as $(whoami)"' >> /etc/profile
root@localhost:~# echo 'echo "You are logged out now"' >> /etc/bash.bash_logout
root@localhost:~# su - root
Hello, you are logged in as root
Hello, you are logged in as root
[root@localhost ~]# su - alex
su: warning: cannot change directory to /home/user1: No such file or directory
Hello, you are logged in as alex
Hello, you are logged in as alex
[alex@localhost root]$ exit
logout
You are logged out now
[root@localhost ~]#
```

17. Configure system parameters for newly created users: o Warning period for password expiry: 5 days o Minimum user UID: 2000 o Maximum user UID: 70000

```
Activities Terminal Jan 30 23:23 root@localhost:~
# Password aging controls:
#
#     PASS_MAX_DAYS    Maximum number of days a password may be used.
#     PASS_MIN_DAYS    Minimum number of days allowed between password changes.
#     PASS_MIN_LEN     Minimum acceptable password length.
#     PASS_WARN_AGE    Number of days warning given before a password expires.
#
PASS_MAX_DAYS    99999
PASS_MIN_DAYS    0
PASS_WARN_AGE    5

# Currently PASS_MIN_LEN is not supported

# Currently SU_WHEEL_ONLY is not supported

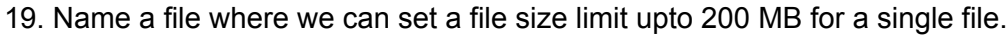
# Currently CRACKLIB_DICTPATH is not supported

#
# Min/max values for automatic uid selection in useradd(8)
#
UID_MIN          2000
UID_MAX          70000
# System accounts
SYS_UID_MIN      201
SYS_UID_MAX      999
# Extra per user uids
SUB_UID_MIN      100000
SUB_UID_MAX      600100000
SUB_UID_COUNT    65536

#
# Min/max values for automatic gid selection in groupadd(8)
#
GID_MIN          1000
GID_MAX          60000
# System accounts
SYS_GID_MIN      201
SYS_GID_MAX      999
# Extra per user group ids
SUB_GID_MIN      100000
SUB_GID_MAX      600100000
SUB_GID_COUNT    65536

#
-- INSERT --
```

18. Create a directory /data and configure the system so that all newly created users get /data as their home directory by default.



19. Name a file where we can set a file size limit upto 200 MB for a single file.

```
Activities Terminal Jan 30 23:28 root@localhost:~
#
#Where:
#<domain> can be:
#   - a user name
#   - a group name, with @group syntax
#   - the wildcard *, for default entry
#   - the wildcard %, can be also used with %group syntax,
#       for maxlogin limit
#
#<type> can have the two values:
#   - "soft" for enforcing the soft limits
#   - "hard" for enforcing hard limits
#
#<item> can be one of the following:
#   - core - limits the core file size (KB)
#   - data - max data size (KB)
#   - fsize - maximum filesize (KB)
#   - memlock - max locked-in-memory address space (KB)
#   - nofile - max number of open file descriptors
#   - rss - max resident set size (KB)
#   - stack - max stack size (KB)
#   - cpu - max CPU time (MIN)
#   - nproc - max number of processes
#   - as - address space limit (KB)
#   - maxlogins - max number of logins for this user
#   - maxsyslogins - max number of logins on the system
#   - priority - the priority to run user process with
#   - locks - max number of file locks the user can hold
#   - sigpending - max number of pending signals
#   - msgqueue - max memory used by POSIX message queues (bytes)
#   - nice - max nice priority allowed to raise to values: [-20, 19]
#   - rtprio - max realtime priority
#
#<domain>    <type>  <item>        <value>
#
#*           soft   core          0
#*           hard   rss            10000
#@student    hard   nproc          204800
#@faculty    soft   nproc          204800
#@faculty    hard   nproc           50
#ftp         hard   nproc           0
#@student    -      maxlogins       4

# End of file
-- INSERT --
```

56,48

Bot

20. Check the last three users who logged into your system.

```
Activities Terminal Jan 30 23:29
root@localhost:~
[root@localhost ~]# last -n 3
root      tty2          tty2          Thu Jan 30 22:13    still logged in
root      seat0          login screen   Thu Jan 30 22:13    still logged in
reboot    system boot    5.14.0-362.8.1.e Thu Jan 30 22:13    still running

wtmp begins Mon Jan 20 15:28:32 2025
[root@localhost ~]#
```

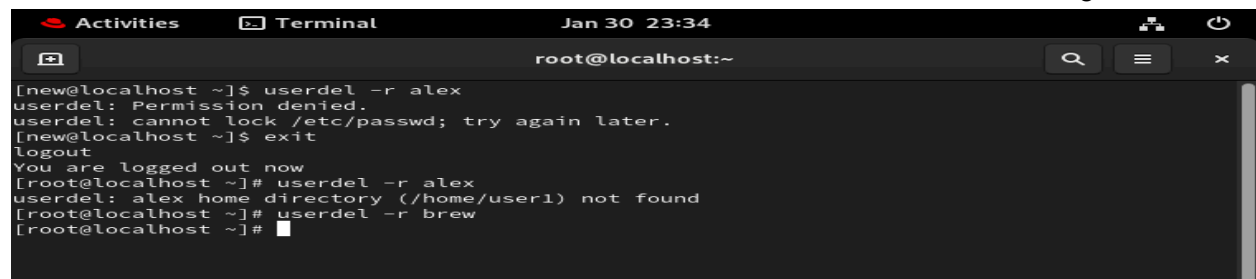
21. As a system administrator, how would you configure the system to ensure that:

- o Automatically create an instructions.txt file in the home directory of every new user upon account creation.
- o Ensure that the mail directory for every newly created user is set to /home/spool/mail/ by default?"

```
Activities Terminal Jan 30 23:32
new@localhost:~
[root@localhost ~]# echo "Welcome to your new account!" > /etc/skel/instructions.txt
[root@localhost ~]# useradd new
[root@localhost ~]# su - new
Hello, you are logged in as new
Hello, you are logged in as new
[new@localhost ~]$ ls -l new
ls: cannot access 'new': No such file or directory
[new@localhost ~]$ ls
instructions.txt
[new@localhost ~]$
```

```
Activities Terminal Jan 30 23:33 new@localhost:~
# Currently ISSUE_FILE is not supported
# Currently TTYTYPE_FILE is not supported
# Currently FTMP_FILE is not supported
# Currently NOLOGINS_FILE is not supported
# Currently SU_NAME is not supported
# *REQUIRED*
#   Directory where mailboxes reside, _or_ name of file, relative to the
#   home directory. If you _do_ define both, MAIL_DIR takes precedence.
#
MAIL_DIR      /home/spool/mail/
MAIL_FILE     .mail
#
# If defined, file which inhibits all the usual chatter during the login
# sequence. If a full pathname, then hushed mode will be enabled if the
# user's name or shell are found in the file. If not a full pathname, then
# hushed mode will be enabled if the file exists in the user's home directory.
#
#HUSHLOGIN_FILE .hushlogin
#HUSHLOGIN_FILE /etc/hushlogins
# Currently ENV_TZ is not supported
# Currently ENV_HZ is not supported
#
# The default PATH settings, for superuser and normal users.
#
# (they are minimal, add the rest in the shell startup files)
#ENV_SUPATH    PATH=/sbin:/bin:/usr/sbin:/usr/bin
#ENV_PATH      PATH=/bin:/usr/bin
#
# Terminal permissions
#
#       TTYGROUP      Login tty will be assigned this group ownership.
#       TTYPERM       Login tty will be set to this permission.
#
# If you have a write(1) program which is "setgid" to a special group
# which owns the terminals, define TTYGROUP as the number of such group
-- INSERT -- W10: Warning: Changing a readonly file 71,27-34 24%
```

22. Delete some users o Named 'alex' and 'brew' with its all data contents including mail data.

A terminal window titled 'Terminal' with a timestamp of 'Jan 30 23:34'. The window shows a sequence of commands and their outputs. First, a user 'new' runs 'userdel -r alex', which results in 'Permission denied' and 'cannot lock /etc/passwd; try again later.'. Then, the user runs 'exit', leading to a 'logout' message and 'You are logged out now'. Finally, the user switches to 'root' and runs 'userdel -r alex', which returns 'alex home directory (/home/user1) not found'. The user then runs 'userdel -r brew' and the prompt returns.

```
root@localhost:~  
[new@localhost ~]$ userdel -r alex  
userdel: Permission denied.  
userdel: cannot lock /etc/passwd; try again later.  
[new@localhost ~]$ exit  
logout  
You are logged out now  
[root@localhost ~]# userdel -r alex  
userdel: alex home directory (/home/user1) not found  
[root@localhost ~]# userdel -r brew  
[root@localhost ~]#
```