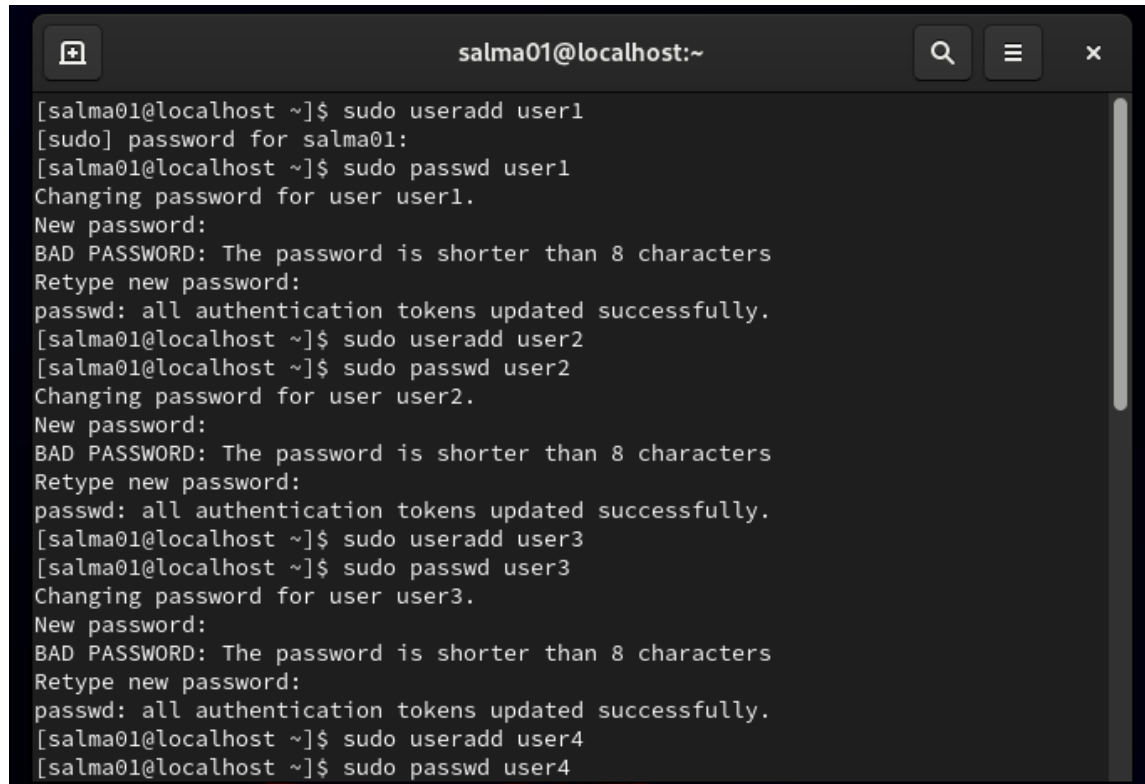


Lab

1. Using the `useradd` command, add accounts for the following users in your system: `user1`, `user2`, `user3`, `user4`, `user5`, `user6` and `user7`. Remember to give each user a password.



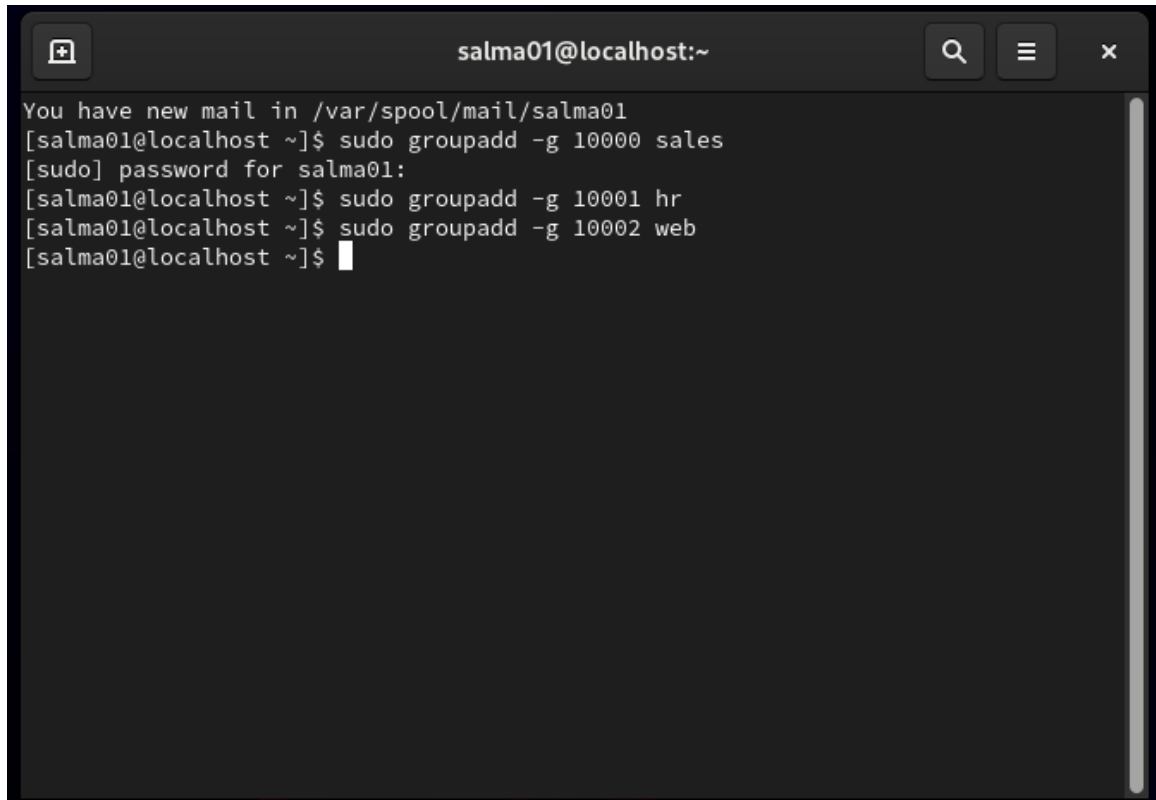
```
salma01@localhost:~  
[salma01@localhost ~]$ sudo useradd user1  
[sudo] password for salma01:  
[salma01@localhost ~]$ sudo passwd user1  
Changing password for user user1.  
New password:  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[salma01@localhost ~]$ sudo useradd user2  
[salma01@localhost ~]$ sudo passwd user2  
Changing password for user user2.  
New password:  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[salma01@localhost ~]$ sudo useradd user3  
[salma01@localhost ~]$ sudo passwd user3  
Changing password for user user3.  
New password:  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[salma01@localhost ~]$ sudo useradd user4  
[salma01@localhost ~]$ sudo passwd user4
```

2. Using the groupadd command, add the following groups to your system.

Group	GID
sales	10000
hr	10001
web	10002

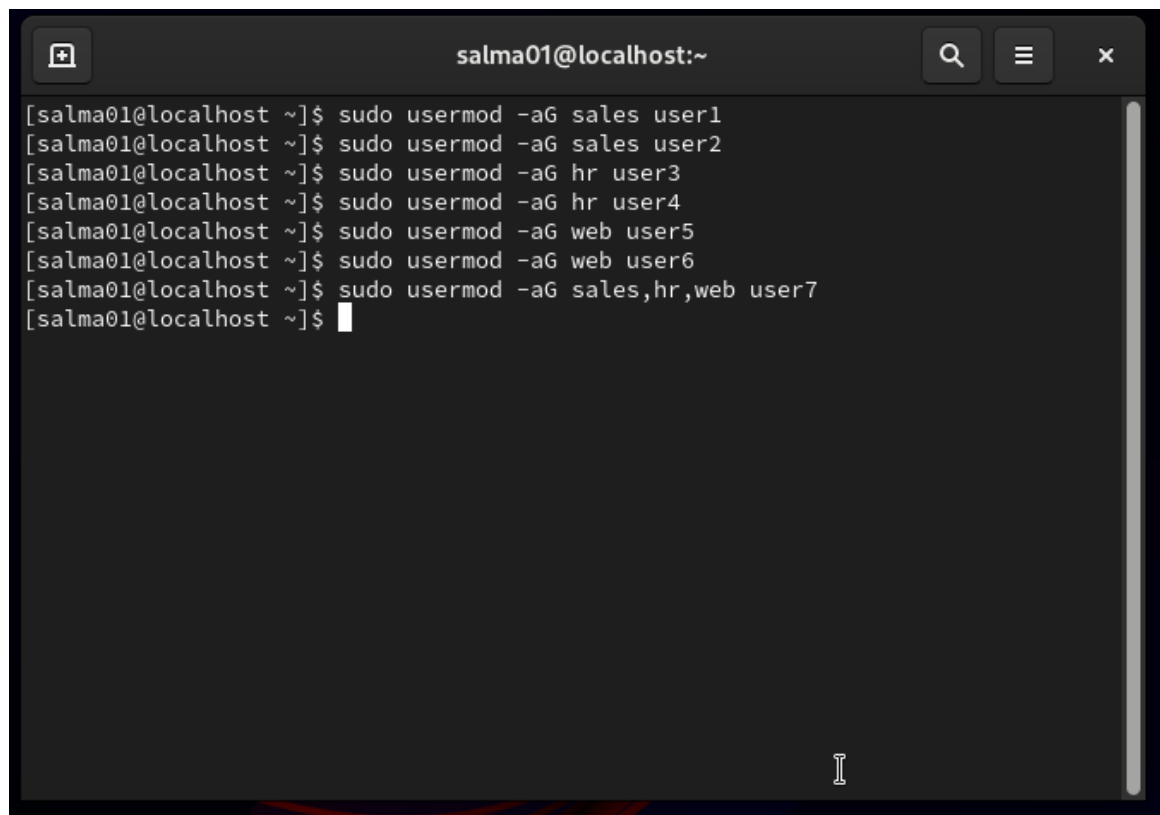
Why should you set GID in this manner instead of allowing the system to set the GID by default?

it allows you to have control over the numeric group identifier assigned to each group.



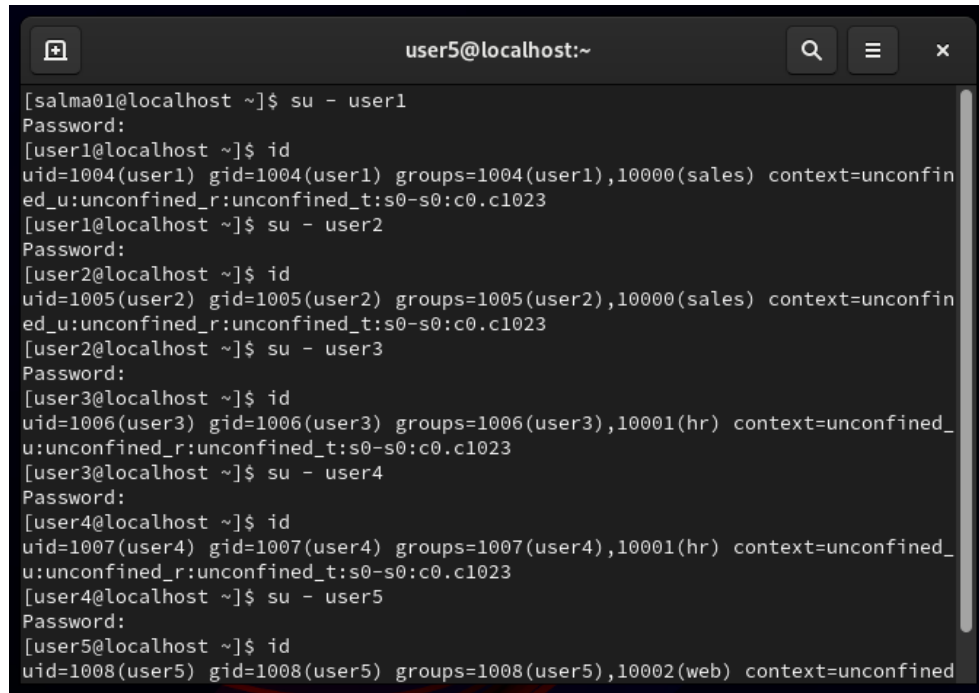
```
salma01@localhost:~  
You have new mail in /var/spool/mail/salma01  
[salma01@localhost ~]$ sudo groupadd -g 10000 sales  
[sudo] password for salma01:  
[salma01@localhost ~]$ sudo groupadd -g 10001 hr  
[salma01@localhost ~]$ sudo groupadd -g 10002 web  
[salma01@localhost ~]$
```

3. Using the usermod command to add user1 and user2 to the sales secondary group, user3 and user4 to the hr secondary group. User5 and user6 to web secondary group. And add user7 to all secondary groups.

A terminal window titled 'salma01@localhost:~' with search, menu, and close icons in the title bar. The terminal displays a series of 'sudo usermod' commands being executed. The commands are: 'sudo usermod -aG sales user1', 'sudo usermod -aG sales user2', 'sudo usermod -aG hr user3', 'sudo usermod -aG hr user4', 'sudo usermod -aG web user5', 'sudo usermod -aG web user6', and 'sudo usermod -aG sales,hr,web user7'. Each command is followed by a new prompt line. A vertical scrollbar is on the right side of the terminal area.

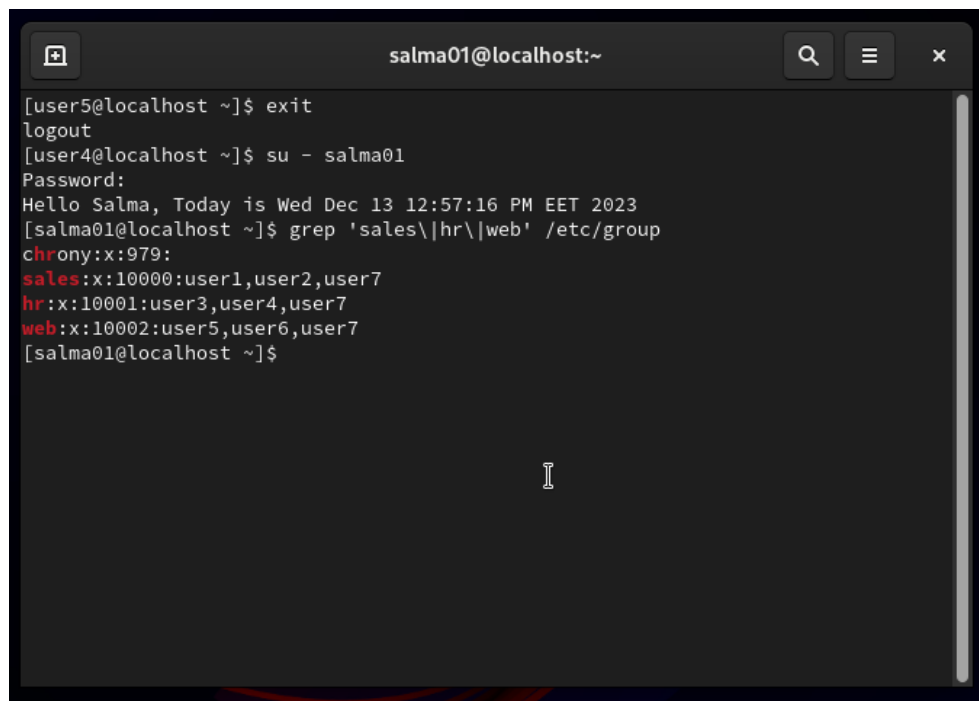
```
[salma01@localhost ~]$ sudo usermod -aG sales user1
[salma01@localhost ~]$ sudo usermod -aG sales user2
[salma01@localhost ~]$ sudo usermod -aG hr user3
[salma01@localhost ~]$ sudo usermod -aG hr user4
[salma01@localhost ~]$ sudo usermod -aG web user5
[salma01@localhost ~]$ sudo usermod -aG web user6
[salma01@localhost ~]$ sudo usermod -aG sales,hr,web user7
[salma01@localhost ~]$
```

4. Login as each user and use id command to verify that they are in the appropriate groups. How else might you verify this information?



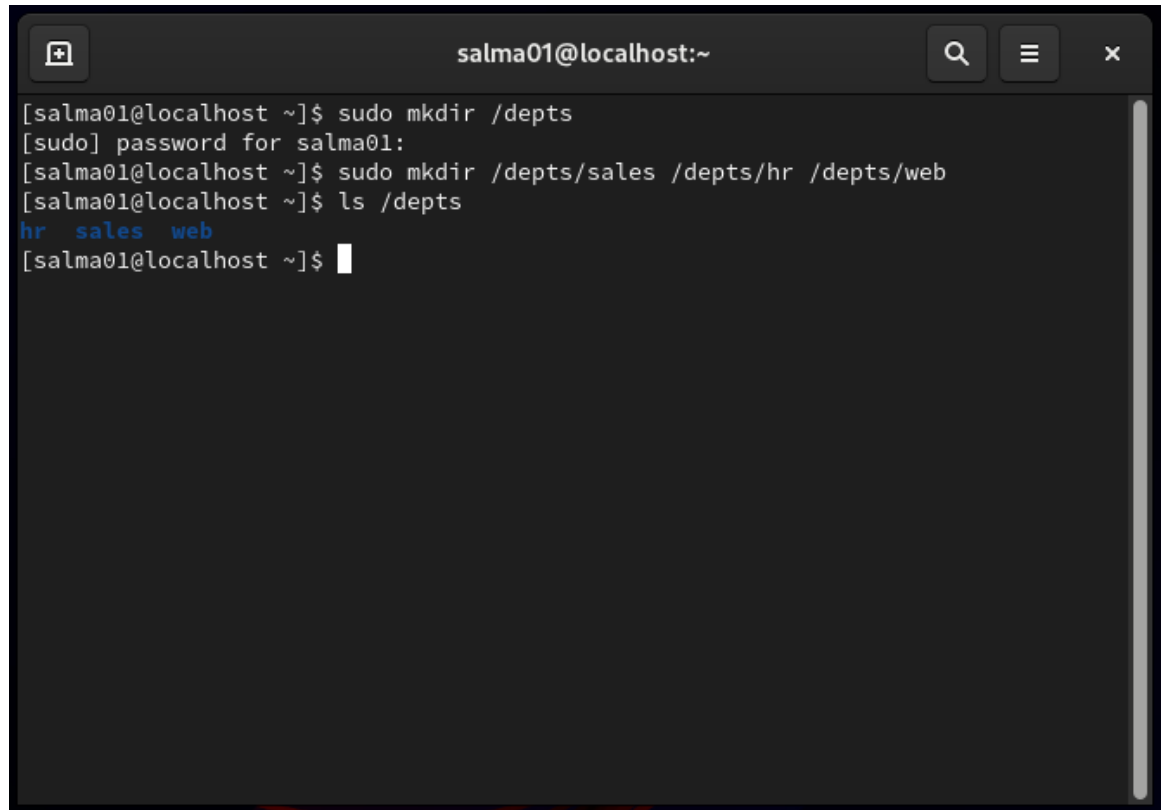
```
user5@localhost:~  
[salma01@localhost ~]$ su - user1  
Password:  
[user1@localhost ~]$ id  
uid=1004(user1) gid=1004(user1) groups=1004(user1),10000(sales) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[user1@localhost ~]$ su - user2  
Password:  
[user2@localhost ~]$ id  
uid=1005(user2) gid=1005(user2) groups=1005(user2),10000(sales) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[user2@localhost ~]$ su - user3  
Password:  
[user3@localhost ~]$ id  
uid=1006(user3) gid=1006(user3) groups=1006(user3),10001(hr) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[user3@localhost ~]$ su - user4  
Password:  
[user4@localhost ~]$ id  
uid=1007(user4) gid=1007(user4) groups=1007(user4),10001(hr) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[user4@localhost ~]$ su - user5  
Password:  
[user5@localhost ~]$ id  
uid=1008(user5) gid=1008(user5) groups=1008(user5),10002(web) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Another way: by using grep command



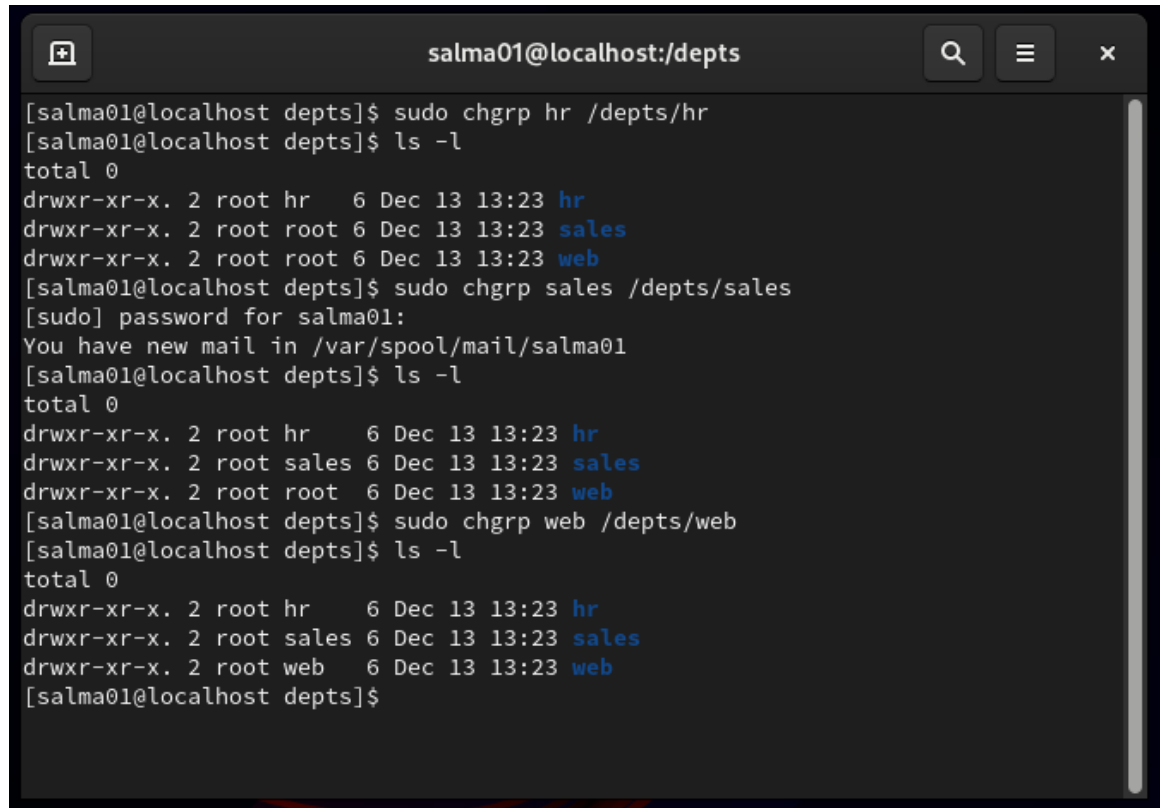
```
salma01@localhost:~  
[user5@localhost ~]$ exit  
logout  
[user4@localhost ~]$ su - salma01  
Password:  
Hello Salma, Today is Wed Dec 13 12:57:16 PM EET 2023  
[salma01@localhost ~]$ grep 'sales\|hr\|web' /etc/group  
chorny:x:979:  
sales:x:10000:user1,user2,user7  
hr:x:10001:user3,user4,user7  
web:x:10002:user5,user6,user7  
[salma01@localhost ~]$
```

5. Create a directory called /depts with a sales, hr, and web directory within the /depts directory.



```
salma01@localhost:~  
[salma01@localhost ~]$ sudo mkdir /depts  
[sudo] password for salma01:  
[salma01@localhost ~]$ sudo mkdir /depts/sales /depts/hr /depts/web  
[salma01@localhost ~]$ ls /depts  
hr  sales  web  
[salma01@localhost ~]$
```

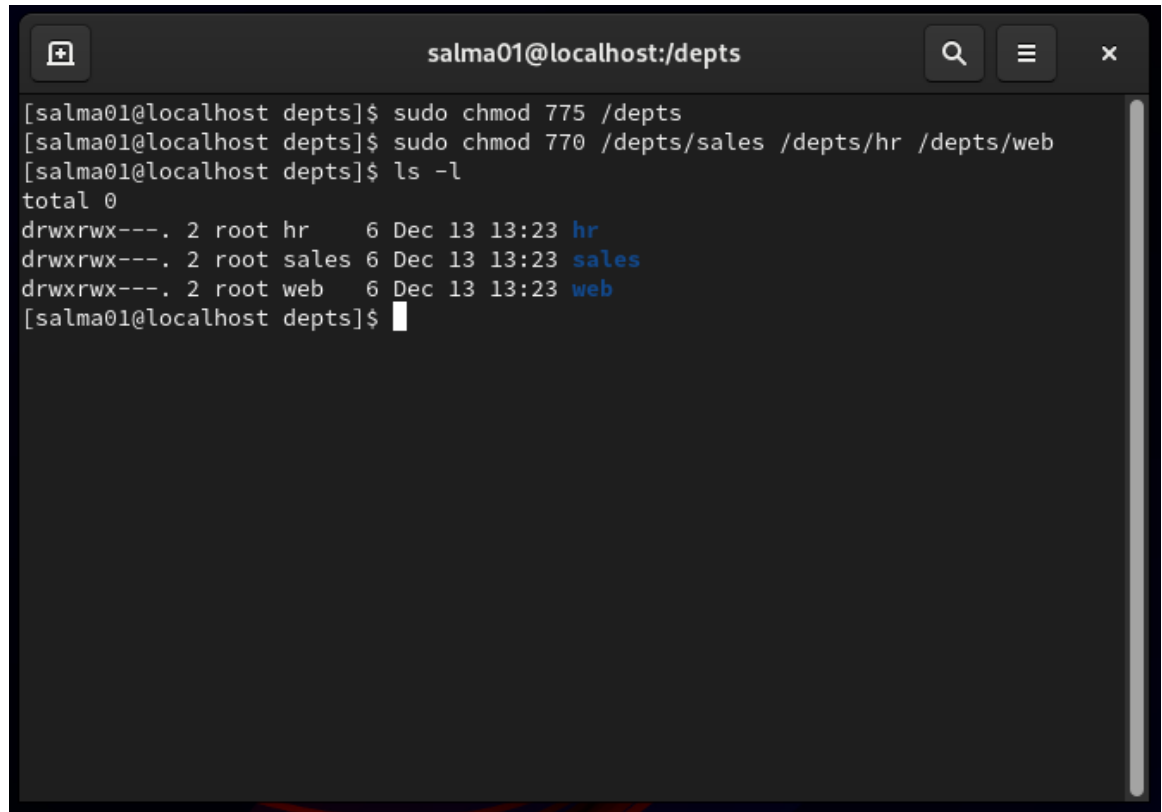
- Using the chgrp command, set the group ownership of each directory to the group with the matching name.



```
salma01@localhost:/depts

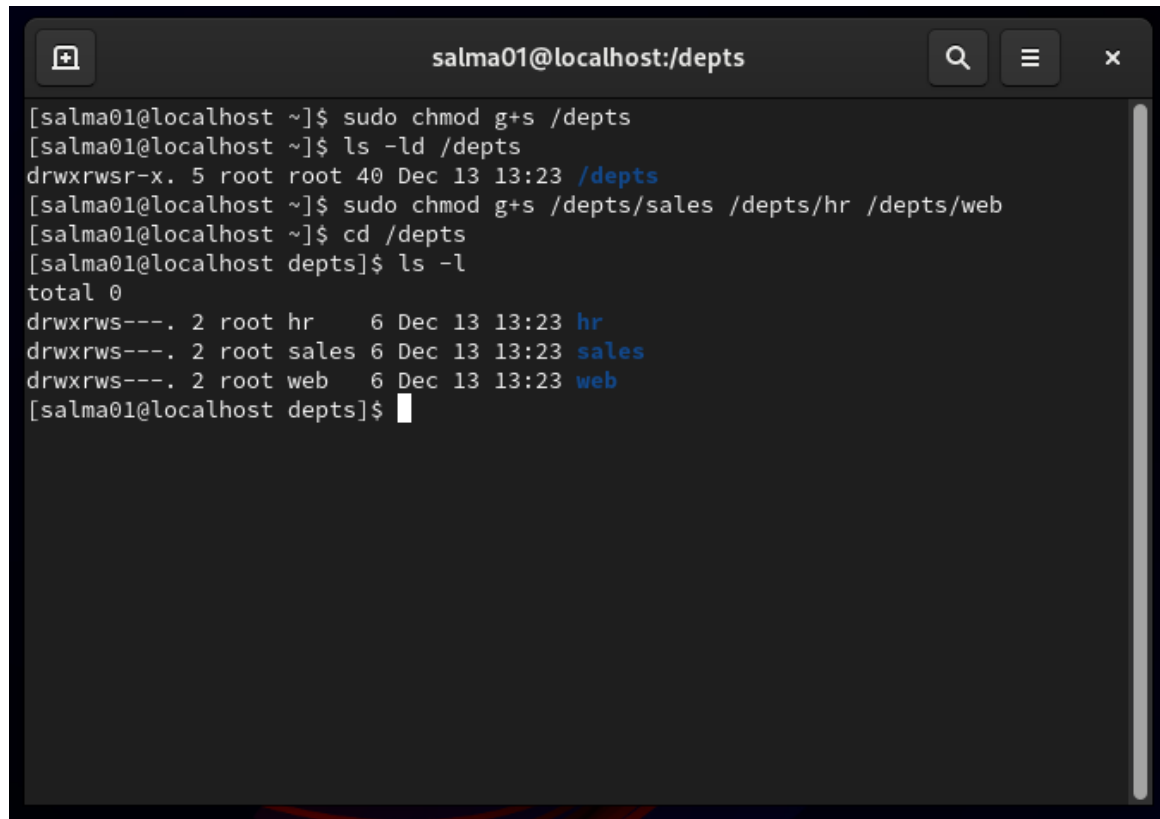
[salma01@localhost depts]$ sudo chgrp hr /depts/hr
[salma01@localhost depts]$ ls -l
total 0
drwxr-xr-x. 2 root hr    6 Dec 13 13:23 hr
drwxr-xr-x. 2 root root  6 Dec 13 13:23 sales
drwxr-xr-x. 2 root root  6 Dec 13 13:23 web
[salma01@localhost depts]$ sudo chgrp sales /depts/sales
[sudo] password for salma01:
You have new mail in /var/spool/mail/salma01
[salma01@localhost depts]$ ls -l
total 0
drwxr-xr-x. 2 root hr    6 Dec 13 13:23 hr
drwxr-xr-x. 2 root sales 6 Dec 13 13:23 sales
drwxr-xr-x. 2 root root  6 Dec 13 13:23 web
[salma01@localhost depts]$ sudo chgrp web /depts/web
[salma01@localhost depts]$ ls -l
total 0
drwxr-xr-x. 2 root hr    6 Dec 13 13:23 hr
drwxr-xr-x. 2 root sales 6 Dec 13 13:23 sales
drwxr-xr-x. 2 root web   6 Dec 13 13:23 web
[salma01@localhost depts]$
```

7. Set the permissions on the /depts directory to 755, and each subdirectory to 770



```
salma01@localhost:/depts
[salma01@localhost depts]$ sudo chmod 755 /depts
[salma01@localhost depts]$ sudo chmod 770 /depts/sales /depts/hr /depts/web
[salma01@localhost depts]$ ls -l
total 0
drwxrwx---. 2 root hr    6 Dec 13 13:23 hr
drwxrwx---. 2 root sales 6 Dec 13 13:23 sales
drwxrwx---. 2 root web   6 Dec 13 13:23 web
[salma01@localhost depts]$
```

8. Set the set-gid bit on each departmental directory.



```
salma01@localhost:/depts

[salma01@localhost ~]$ sudo chmod g+s /depts
[salma01@localhost ~]$ ls -ld /depts
drwxrwsr-x. 5 root root 40 Dec 13 13:23 /depts
[salma01@localhost ~]$ sudo chmod g+s /depts/sales /depts/hr /depts/web
[salma01@localhost ~]$ cd /depts
[salma01@localhost depts]$ ls -l
total 0
drwxrws---. 2 root hr    6 Dec 13 13:23 hr
drwxrws---. 2 root sales 6 Dec 13 13:23 sales
drwxrws---. 2 root web   6 Dec 13 13:23 web
[salma01@localhost depts]$
```


9. Use the su command to switch to the user2 account and attempt the following commands:

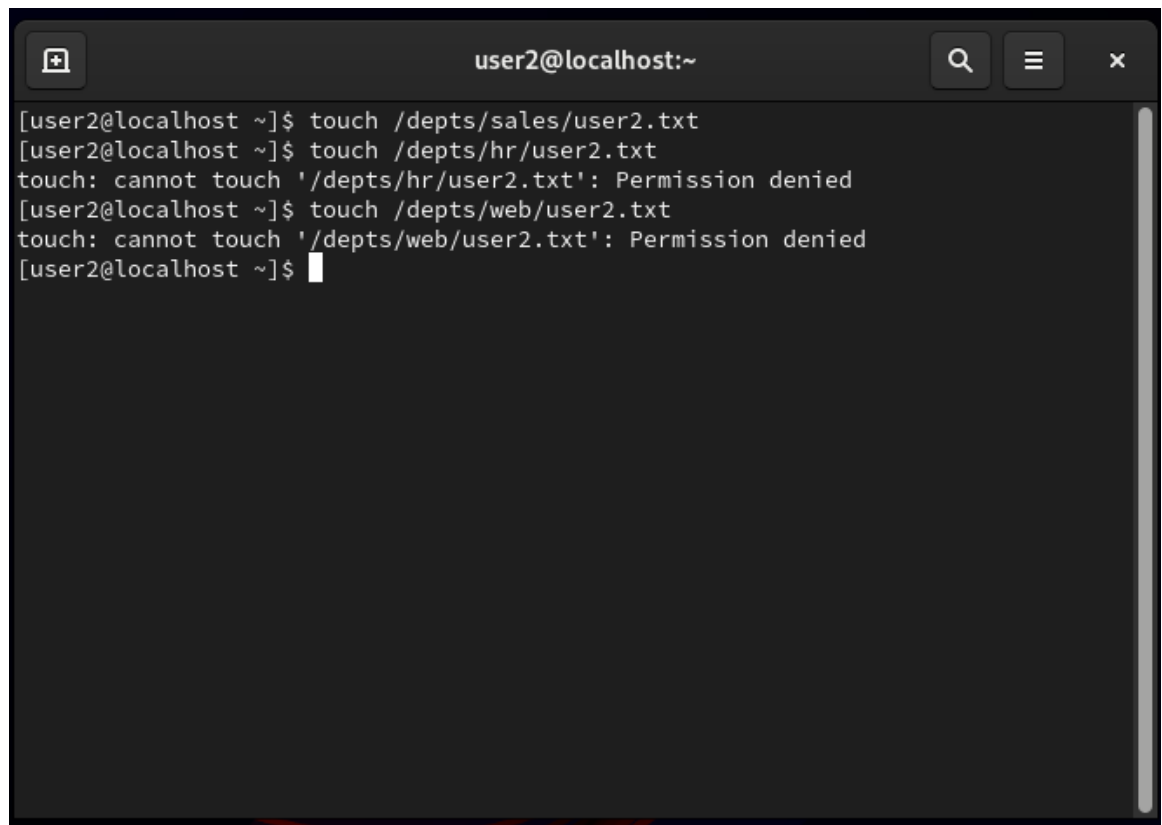
```
touch /depts/sales/user2.txt
```

```
touch /depts/hr/ user2.txt
```

```
touch /depts/web/ user2.txt
```

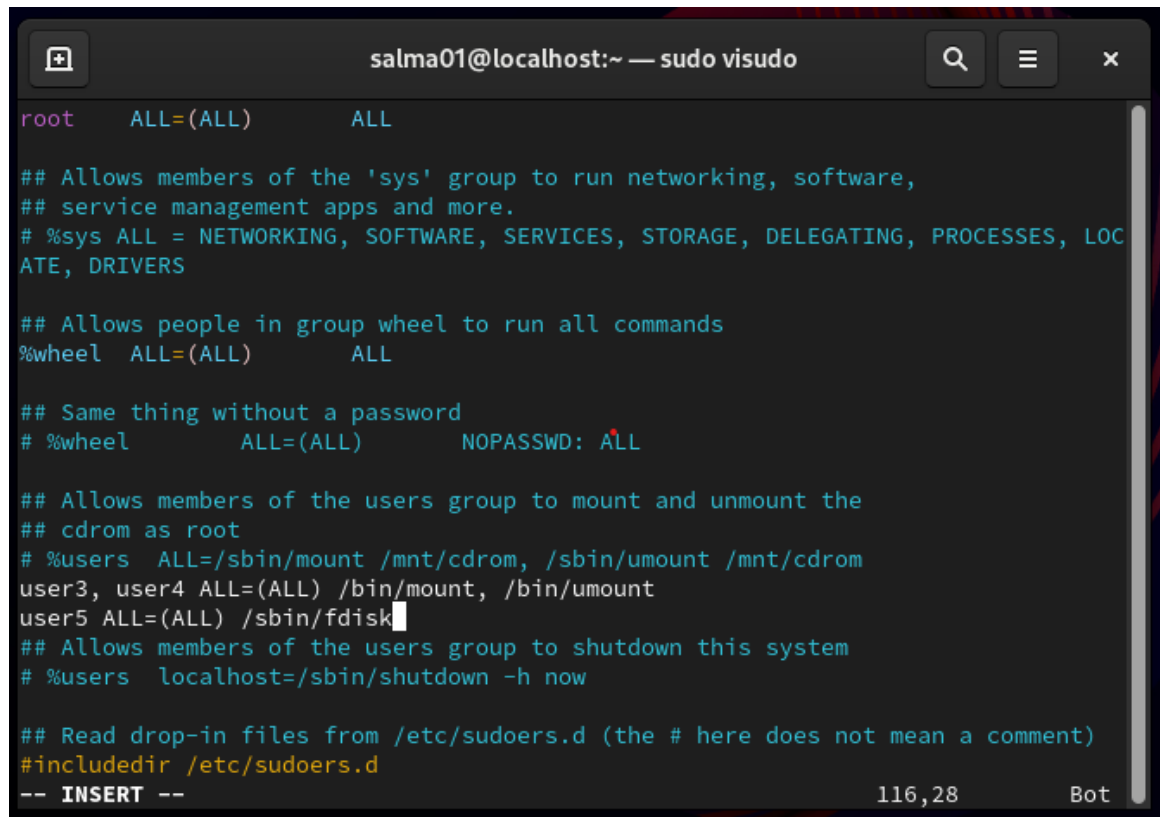
Which of these commands succeeded and which failed? What is the group ownership of the files that were created?

Only first command succeeded because user2 belongs only to sales group.

A terminal window titled 'user2@localhost:~' with search, menu, and close buttons in the title bar. The terminal shows the following commands and output:

```
[user2@localhost ~]$ touch /depts/sales/user2.txt
[user2@localhost ~]$ touch /depts/hr/user2.txt
touch: cannot touch '/depts/hr/user2.txt': Permission denied
[user2@localhost ~]$ touch /depts/web/user2.txt
touch: cannot touch '/depts/web/user2.txt': Permission denied
[user2@localhost ~]$
```

10. Configure sudoers file to allow user3 and user4 to use /bin/mount and /bin/umount commands, while allowing user5 only to use fdisk command.



The image shows a terminal window titled "salma01@localhost:~ — sudo visudo". The window displays the contents of the sudoers file. The configuration includes: root with ALL=(ALL) ALL; a comment about the 'sys' group; %sys with ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS; a comment about the 'wheel' group; %wheel with ALL=(ALL) ALL; a comment about NOPASSWD; %wheel with ALL=(ALL) NOPASSWD: ALL; a comment about the 'users' group; %users with ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom; user3 and user4 with ALL=(ALL) /bin/mount, /bin/umount; user5 with ALL=(ALL) /sbin/fdisk; a comment about shutdown; %users with localhost=/sbin/shutdown -h now; a comment about drop-in files; #includedir /etc/sudoers.d; and -- INSERT -- at the bottom. The terminal shows a cursor at the end of the user5 line.

```
root    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, LOC
ATE, DRIVERS

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

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

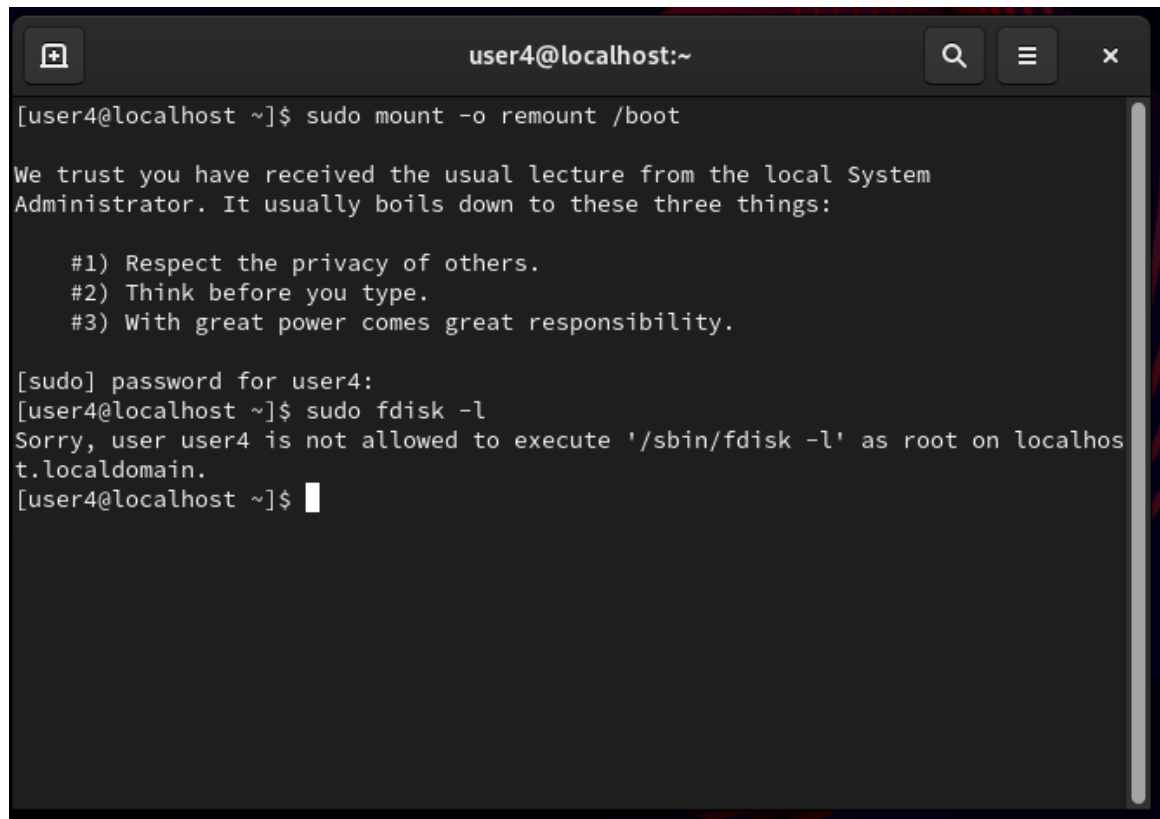
## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom
user3, user4 ALL=(ALL) /bin/mount, /bin/umount
user5 ALL=(ALL) /sbin/fdisk
## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#include_dir /etc/sudoers.d
-- INSERT --
```

116,28 Bot

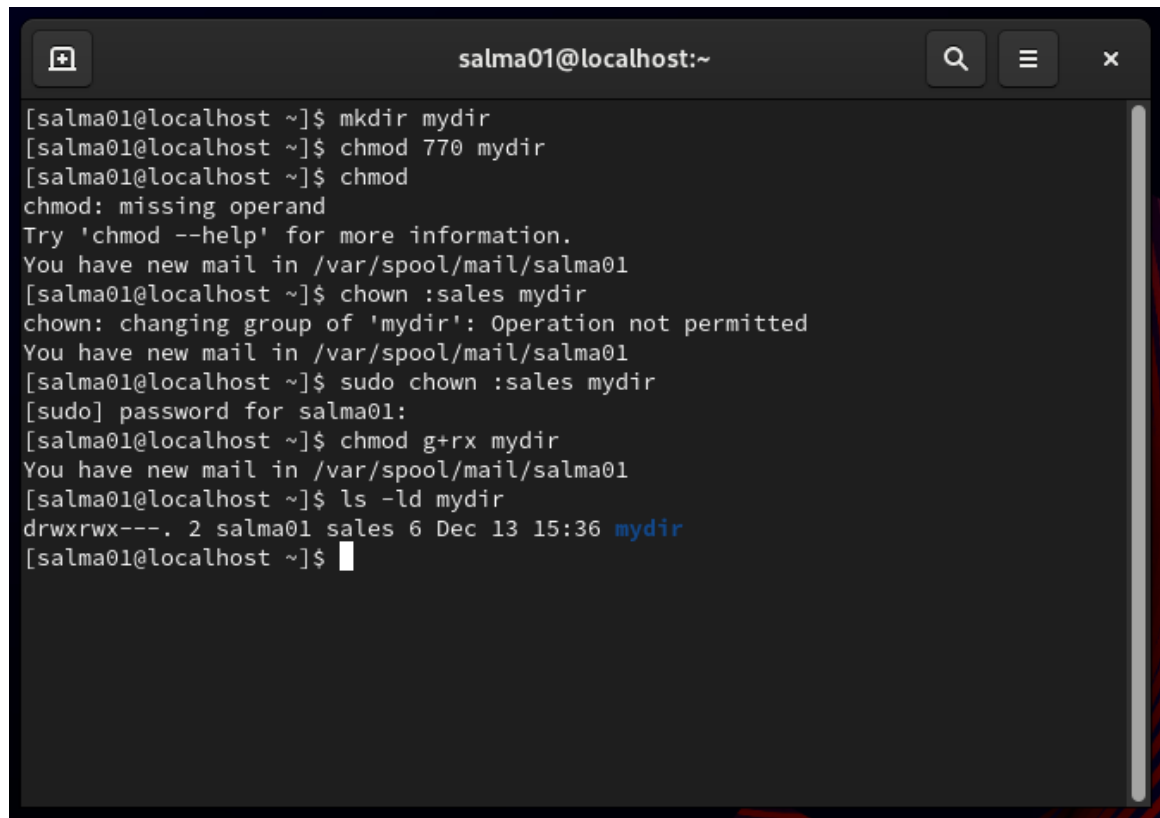
11. Login by user3 and try to unmount /boot.
 Su – user3
 sudo umount /boot

12. Login by user4 and remount /boot. Also try to view the partition table using fdisk. User4 does not has access to view the partition table using fdisk command.



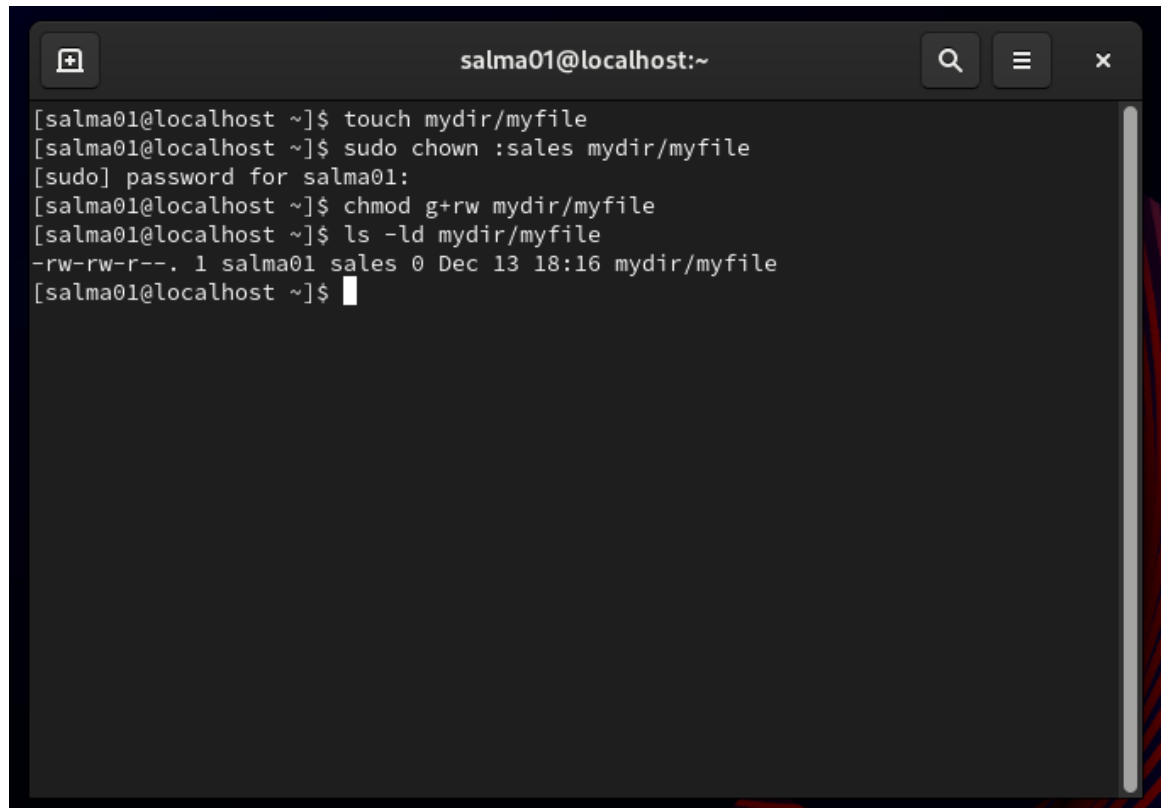
```
user4@localhost:~  
[user4@localhost ~]$ sudo mount -o remount /boot  
  
We trust you have received the usual lecture from the local System  
Administrator. It usually boils down to these three things:  
  
    #1) Respect the privacy of others.  
    #2) Think before you type.  
    #3) With great power comes great responsibility.  
  
[sudo] password for user4:  
[user4@localhost ~]$ sudo fdisk -l  
Sorry, user user4 is not allowed to execute '/sbin/fdisk -l' as root on localhos  
t.localdomain.  
[user4@localhost ~]$
```

13. Create a directory with permissions `rw-rwx---`, grant a second group (`sales`) `r-x` permissions.



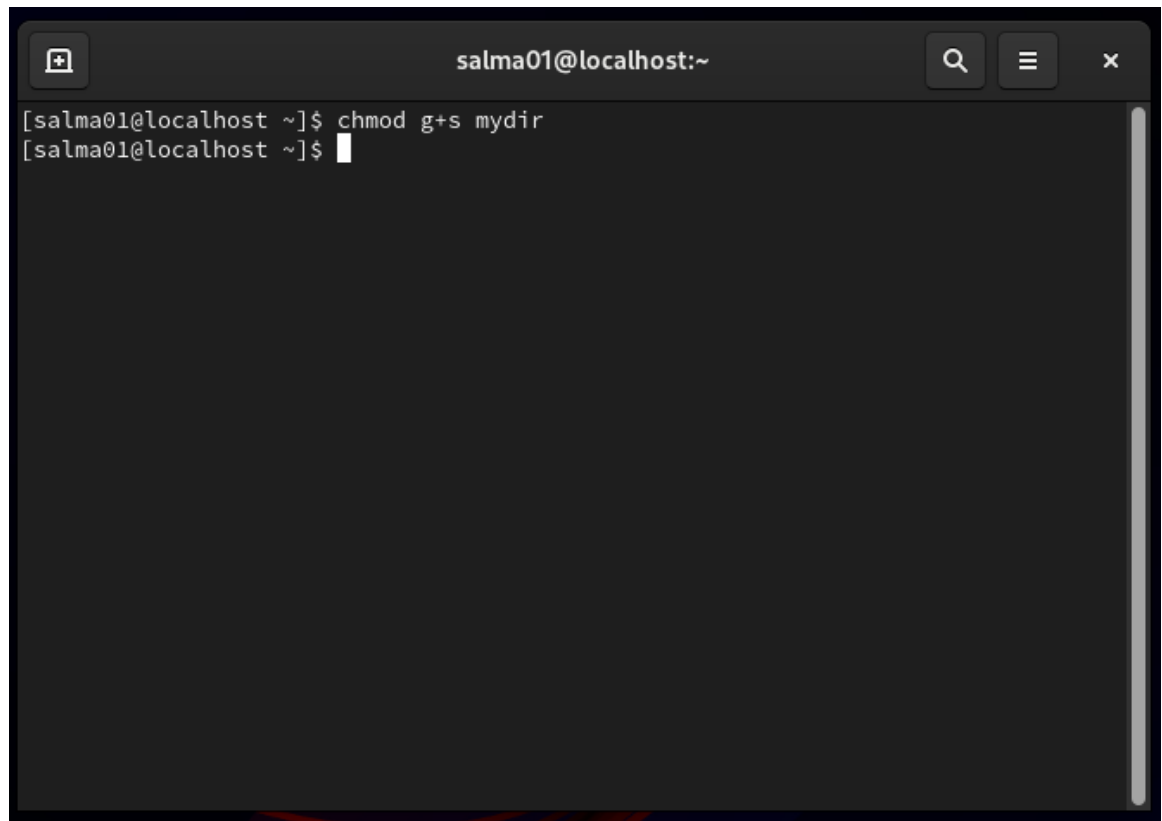
```
salma01@localhost:~  
[salma01@localhost ~]$ mkdir mydir  
[salma01@localhost ~]$ chmod 770 mydir  
[salma01@localhost ~]$ chmod  
chmod: missing operand  
Try 'chmod --help' for more information.  
You have new mail in /var/spool/mail/salma01  
[salma01@localhost ~]$ chown :sales mydir  
chown: changing group of 'mydir': Operation not permitted  
You have new mail in /var/spool/mail/salma01  
[salma01@localhost ~]$ sudo chown :sales mydir  
[sudo] password for salma01:  
[salma01@localhost ~]$ chmod g+rx mydir  
You have new mail in /var/spool/mail/salma01  
[salma01@localhost ~]$ ls -ld mydir  
drwxrwx---. 2 salma01 sales 6 Dec 13 15:36 mydir  
[salma01@localhost ~]$
```

14. create a file on that directory and grant read and write to a second group (sales).



```
salma01@localhost:~  
[salma01@localhost ~]$ touch mydir/myfile  
[salma01@localhost ~]$ sudo chown :sales mydir/myfile  
[sudo] password for salma01:  
[salma01@localhost ~]$ chmod g+rw mydir/myfile  
[salma01@localhost ~]$ ls -ld mydir/myfile  
-rw-rw-r--. 1 salma01 sales 0 Dec 13 18:16 mydir/myfile  
[salma01@localhost ~]$
```

15. set the owning group as the owning group of any newly created file in that directory.

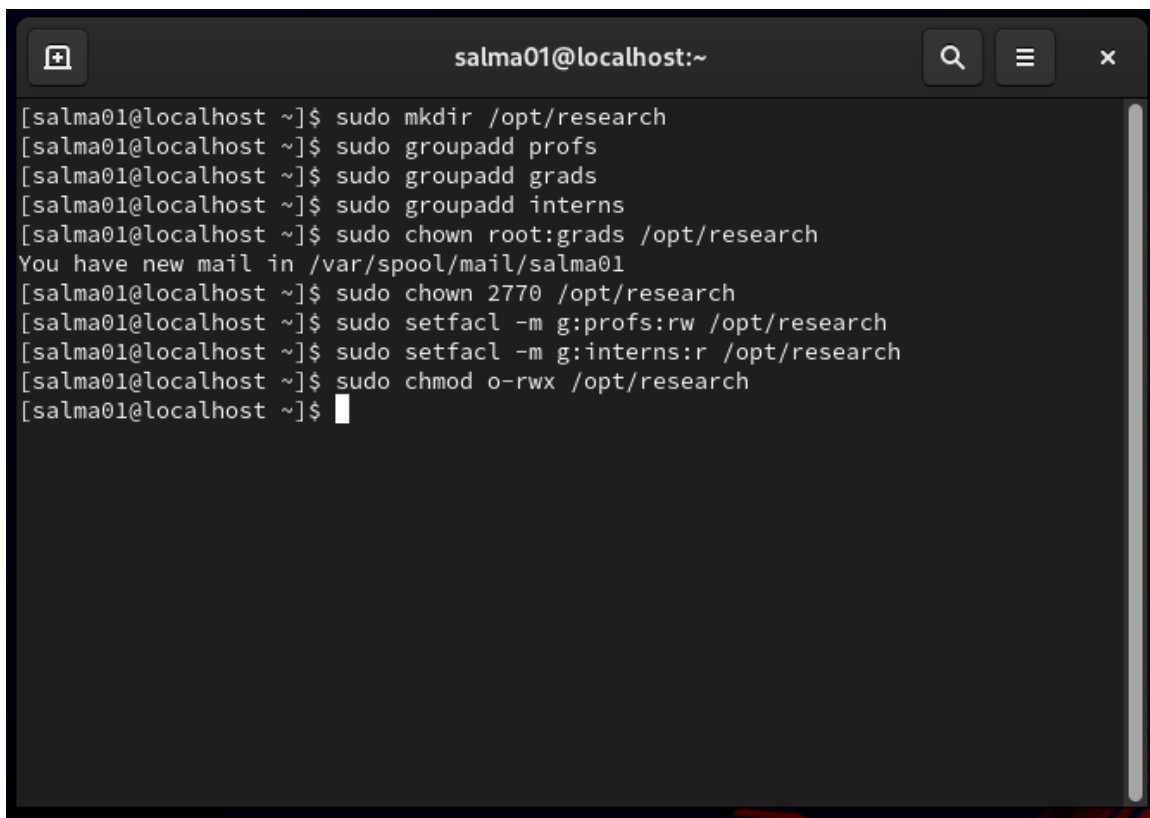


```
salma01@localhost:~  
[salma01@localhost ~]$ chmod g+s mydir  
[salma01@localhost ~]$
```

A terminal window with a dark background. The title bar shows 'salma01@localhost:~' and standard window controls. The terminal content shows the user 'salma01' at 'localhost' in the home directory '~'. They have entered the command 'chmod g+s mydir' and the prompt has moved to the next line.

16. Grant your colleagues a collective directory called `/opt/research`, where they can store generated research results. Only members of group `profs` and `grads` should be able to create new files in the directory, and new file should have the following properties:

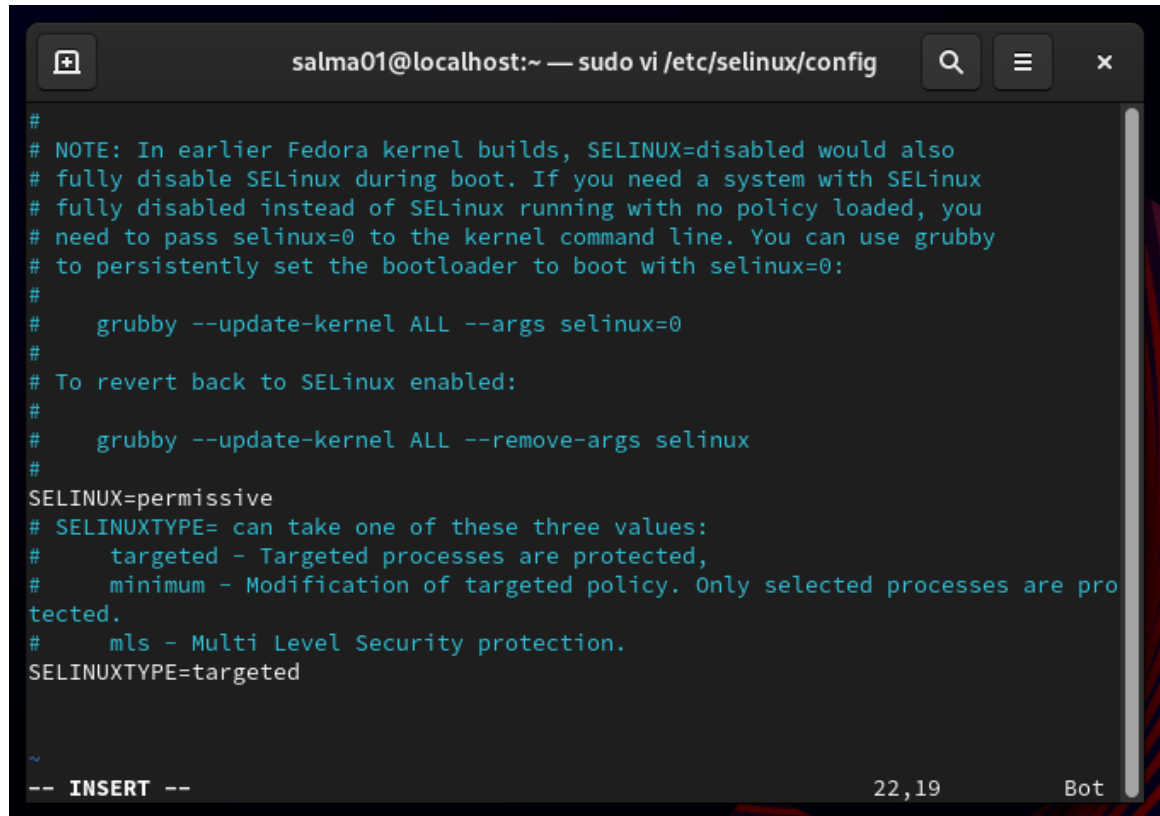
- the directory should be owned by `root`
- new files should be group owned by group `grads`
- group `profs` should automatically have read/write access to new files
- group `interns` should automatically have read only access to new files
- other users should not be able to access the directory and its contents at all.



```
salma01@localhost:~  
[salma01@localhost ~]$ sudo mkdir /opt/research  
[salma01@localhost ~]$ sudo groupadd profs  
[salma01@localhost ~]$ sudo groupadd grads  
[salma01@localhost ~]$ sudo groupadd interns  
[salma01@localhost ~]$ sudo chown root:grads /opt/research  
You have new mail in /var/spool/mail/salma01  
[salma01@localhost ~]$ sudo chown 2770 /opt/research  
[salma01@localhost ~]$ sudo setfacl -m g:profs:rw /opt/research  
[salma01@localhost ~]$ sudo setfacl -m g:interns:r /opt/research  
[salma01@localhost ~]$ sudo chmod o-rwx /opt/research  
[salma01@localhost ~]$
```


17. Change your default SELinux mode to permissive and reboot.

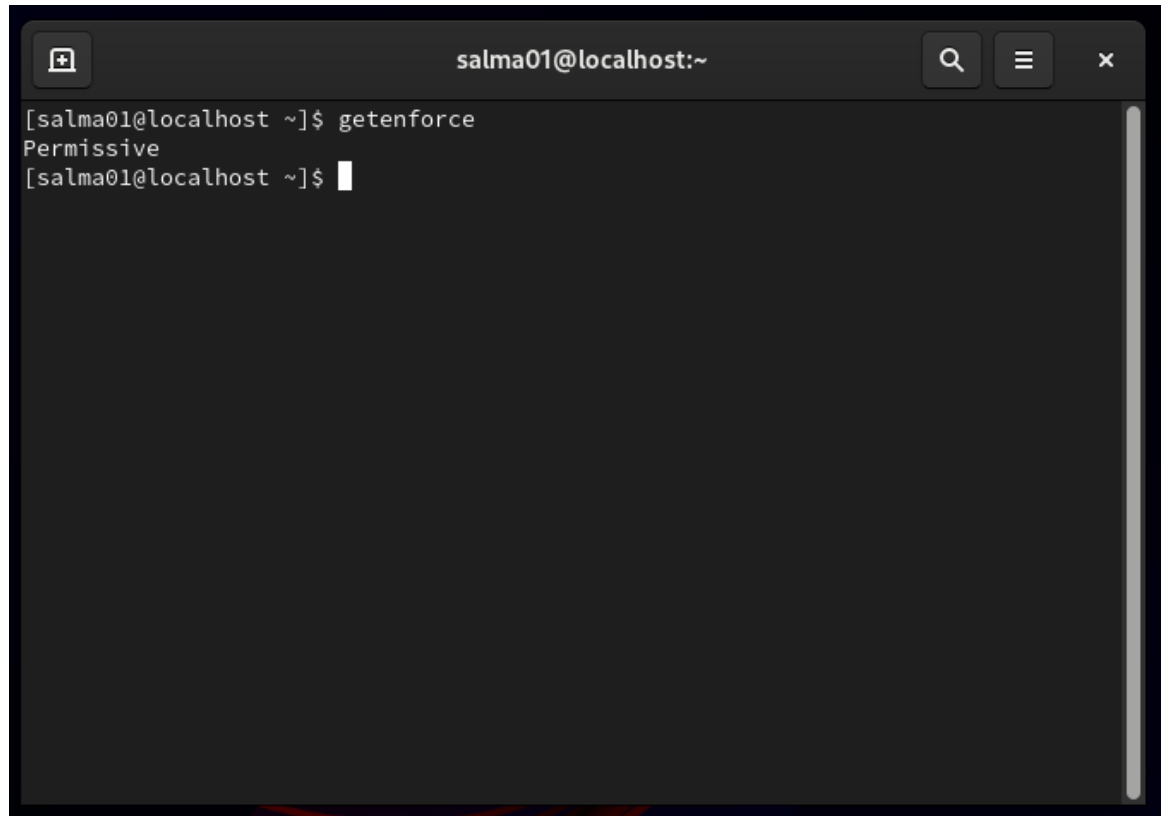
```
sudo nano /etc/selinux/config
```



```
#
# NOTE: In earlier Fedora kernel builds, SELINUX=disabled would also
# fully disable SELinux during boot. If you need a system with SELinux
# fully disabled instead of SELinux running with no policy loaded, you
# need to pass selinux=0 to the kernel command line. You can use grubby
# to persistently set the bootloader to boot with selinux=0:
#
#   grubby --update-kernel ALL --args selinux=0
#
# To revert back to SELinux enabled:
#
#   grubby --update-kernel ALL --remove-args selinux
#
SELINUX=permissive
# SELINUXTYPE= can take one of these three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are pro
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted

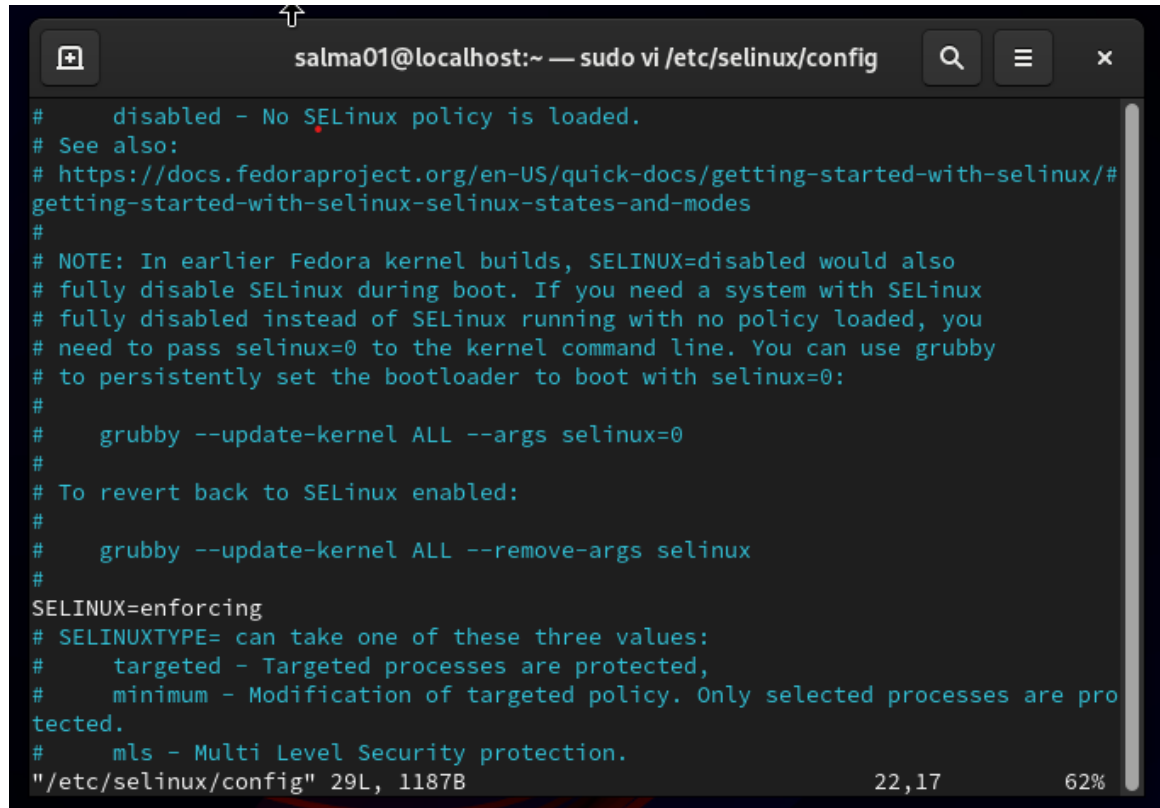
~
-- INSERT --                                22,19                                Bot
```

18. After reboot, verify the system is in permissive mode.

A terminal window with a dark background and light gray text. The window title bar shows 'salma01@localhost:~' and standard window controls (search, menu, close). The terminal content shows the command 'getenforce' being executed, resulting in the output 'Permissive'. The prompt '[salma01@localhost ~]\$' is visible at the end of the line.

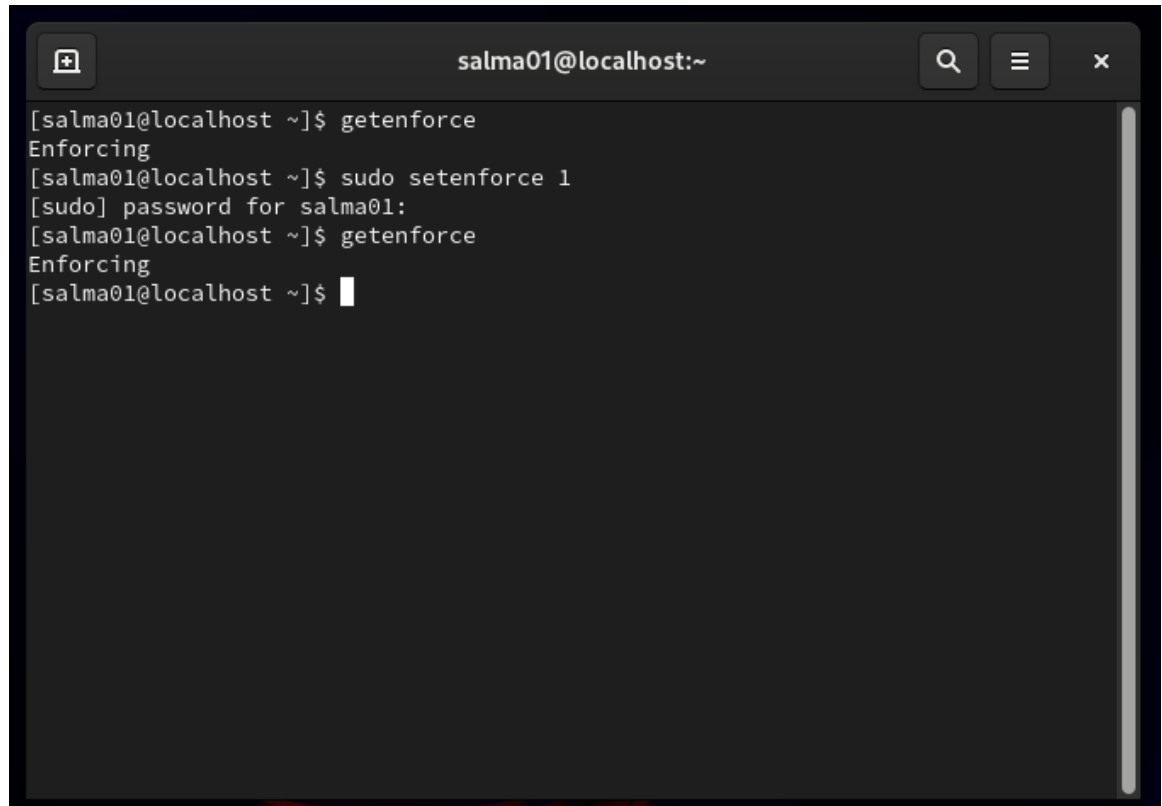
```
[salma01@localhost ~]$ getenforce
Permissive
[salma01@localhost ~]$
```

19. Change the default SELinux mode to enforcing.



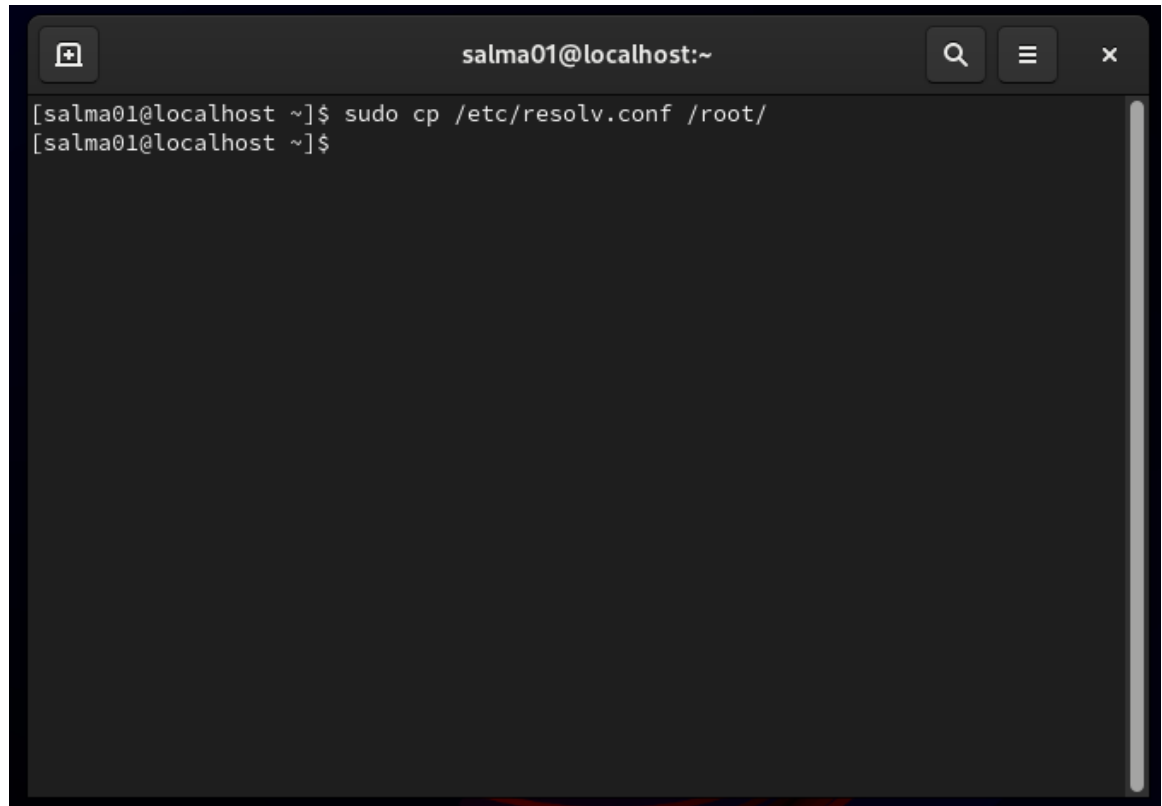
```
salma01@localhost:~ — sudo vi /etc/selinux/config
#      disabled - No SELinux policy is loaded.
# See also:
# https://docs.fedoraproject.org/en-US/quick-docs/getting-started-with-selinux/#
# getting-started-with-selinux-selinux-states-and-modes
#
# NOTE: In earlier Fedora kernel builds, SELINUX=disabled would also
# fully disable SELinux during boot. If you need a system with SELinux
# fully disabled instead of SELinux running with no policy loaded, you
# need to pass selinux=0 to the kernel command line. You can use grubby
# to persistently set the bootloader to boot with selinux=0:
#
#      grubby --update-kernel ALL --args selinux=0
#
# To revert back to SELinux enabled:
#
#      grubby --update-kernel ALL --remove-args selinux
#
SELINUX=enforcing
# SELINUXTYPE= can take one of these three values:
#      targeted - Targeted processes are protected,
#      minimum - Modification of targeted policy. Only selected processes are pro
#      tected.
#      mls - Multi Level Security protection.
"/etc/selinux/config" 29L, 1187B                               22,17                               62%
```

20. Change the current SELinux mode to enforcing.

A terminal window titled 'salma01@localhost:~' with search, menu, and close buttons. It shows the execution of 'getenforce' (output: Enforcing), 'sudo setenforce 1' (prompting for password), and a second 'getenforce' (output: Enforcing).

```
[salma01@localhost ~]$ getenforce
Enforcing
[salma01@localhost ~]$ sudo setenforce 1
[sudo] password for salma01:
[salma01@localhost ~]$ getenforce
Enforcing
[salma01@localhost ~]$
```

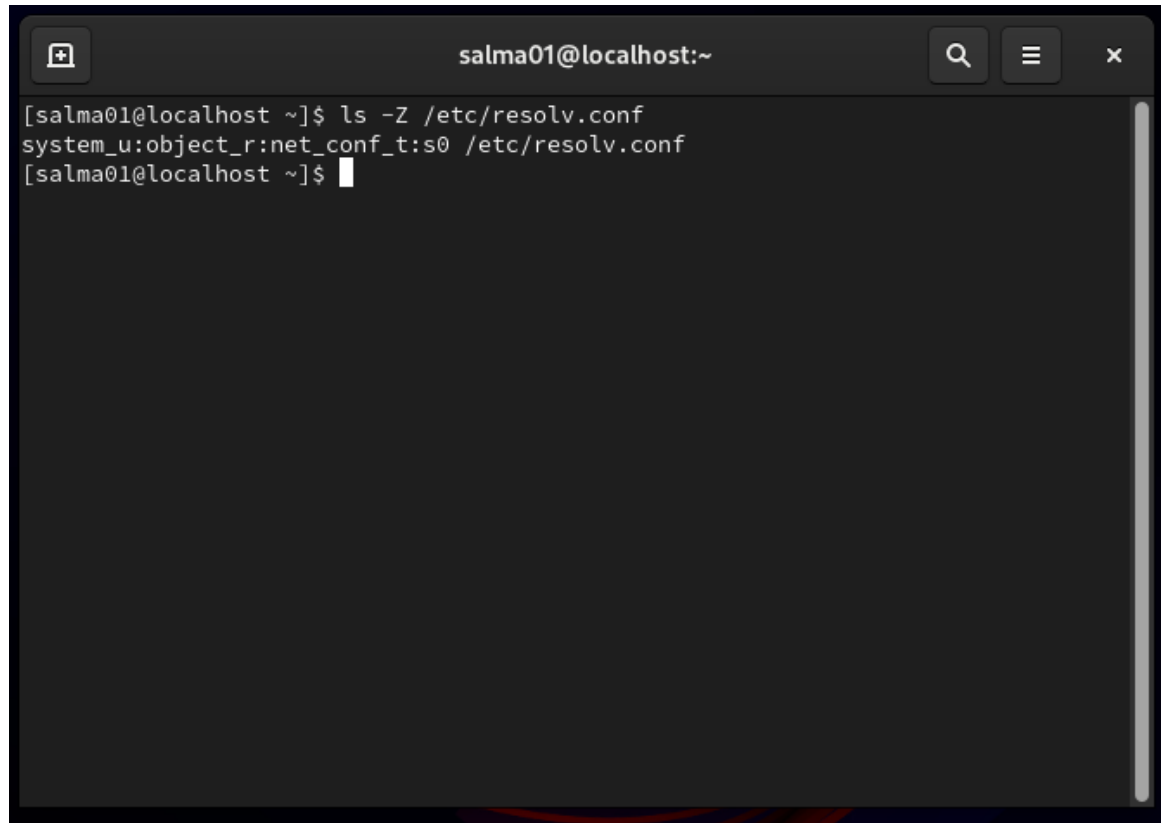
21. Copy /etc/resolv.conf file to root's home directory.



```
salma01@localhost:~  
[salma01@localhost ~]$ sudo cp /etc/resolv.conf /root/  
[salma01@localhost ~]$
```

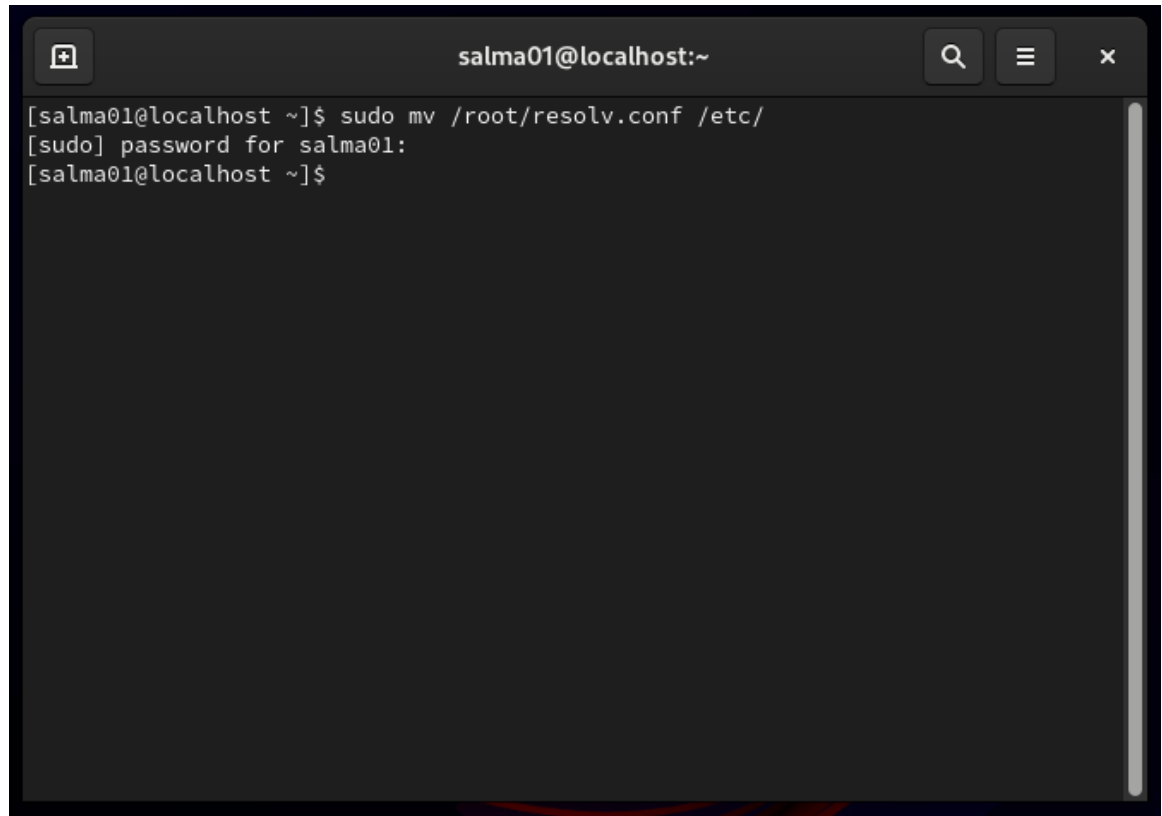
A terminal window with a dark background. The title bar shows 'salma01@localhost:~' and standard window controls (search, menu, close). The terminal content shows a user prompt '[salma01@localhost ~]\$', the command 'sudo cp /etc/resolv.conf /root/', and a subsequent prompt '[salma01@localhost ~]\$'.

22. Observe the SELinux context of the initial /etc/resolv.conf.

A terminal window titled 'salma01@localhost:~' with search, menu, and close buttons. The command '[salma01@localhost ~]\$ ls -Z /etc/resolv.conf' has been executed, resulting in the output 'system_u:object_r:net_conf_t:s0 /etc/resolv.conf'. The prompt '[salma01@localhost ~]\$' is followed by a cursor.

```
[salma01@localhost ~]$ ls -Z /etc/resolv.conf
system_u:object_r:net_conf_t:s0 /etc/resolv.conf
[salma01@localhost ~]$
```

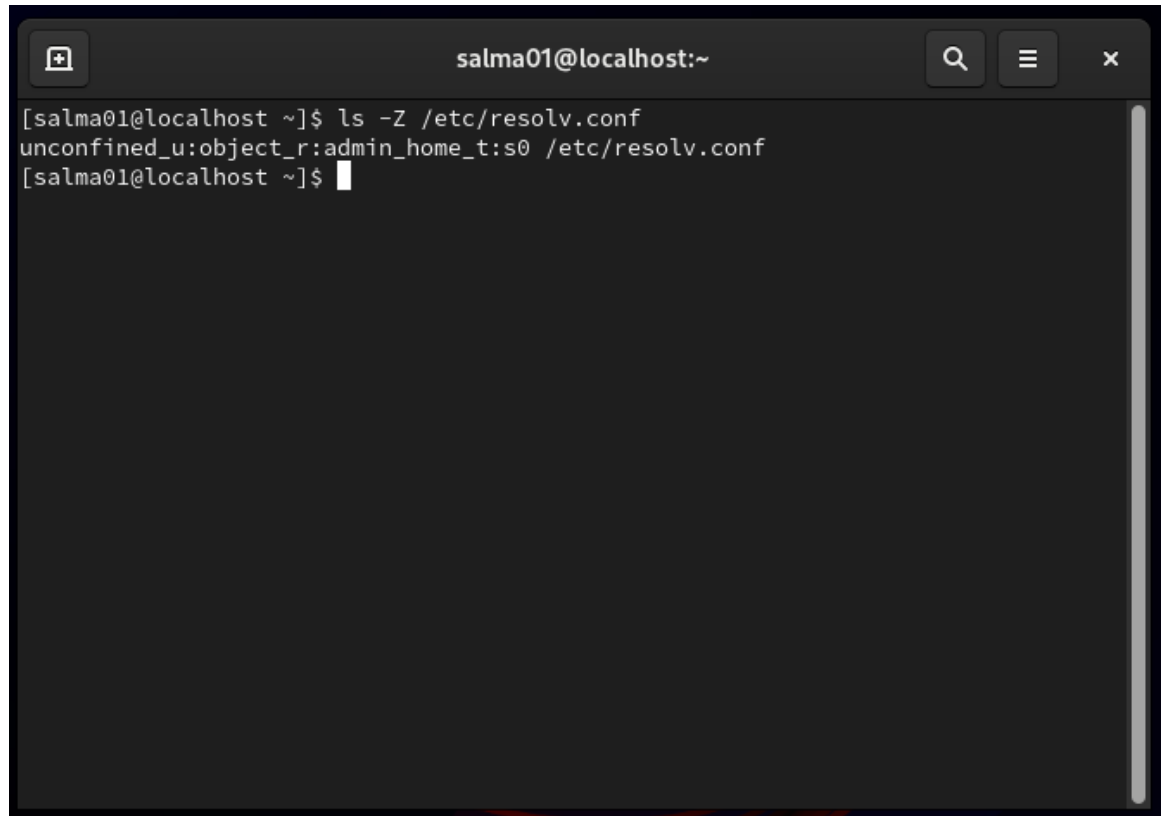
23. Move resolv.conf from root's home directory to /etc/resolv.conf.



```
salma01@localhost:~  
[salma01@localhost ~]$ sudo mv /root/resolv.conf /etc/  
[sudo] password for salma01:  
[salma01@localhost ~]$
```

A terminal window with a dark background. The title bar shows 'salma01@localhost:~' and standard window controls (search, menu, close). The terminal content shows a user running a command to move a file from /root to /etc, followed by a password prompt and a confirmation message.

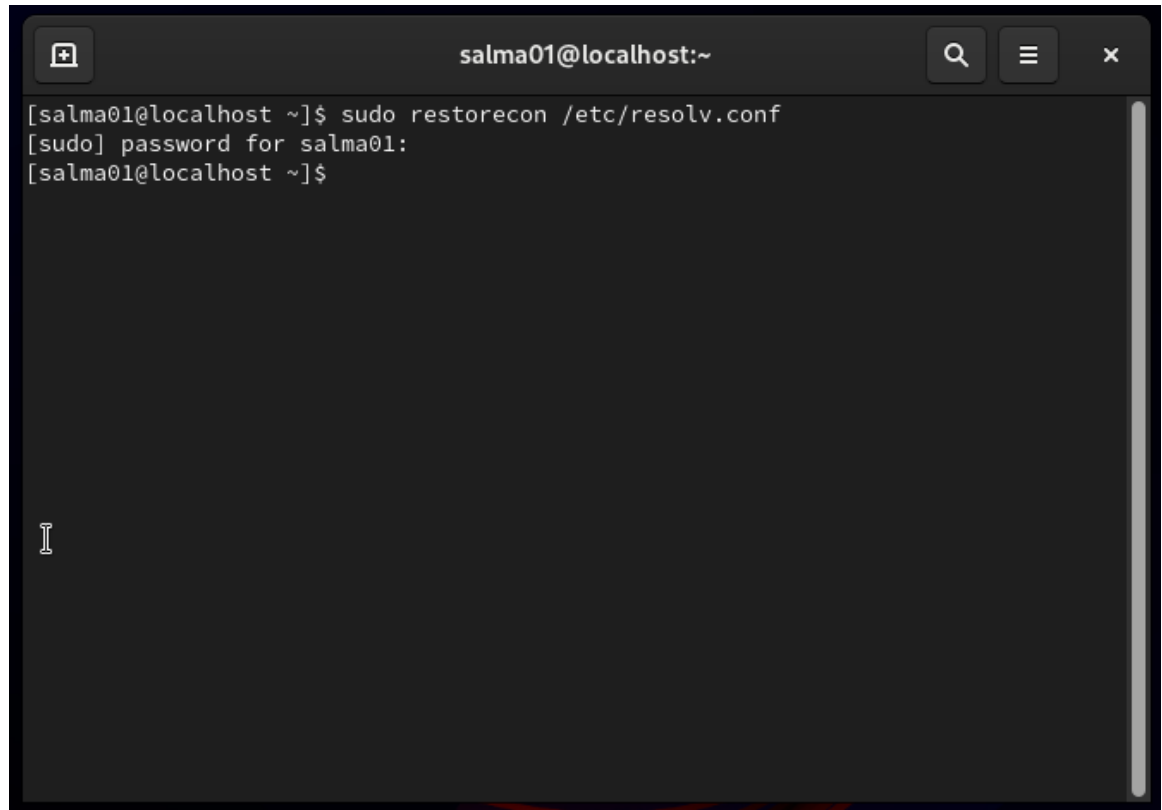
24. Observe the SELinux of the newly copied `/etc/resolv.conf`.



A terminal window titled "salma01@localhost:~" with search, menu, and close buttons. The terminal shows the command `ls -Z /etc/resolv.conf` and its output: `unconfined_u:object_r:admin_home_t:s0 /etc/resolv.conf`. The prompt is `[salma01@localhost ~]$` with a cursor.

```
[salma01@localhost ~]$ ls -Z /etc/resolv.conf
unconfined_u:object_r:admin_home_t:s0 /etc/resolv.conf
[salma01@localhost ~]$
```

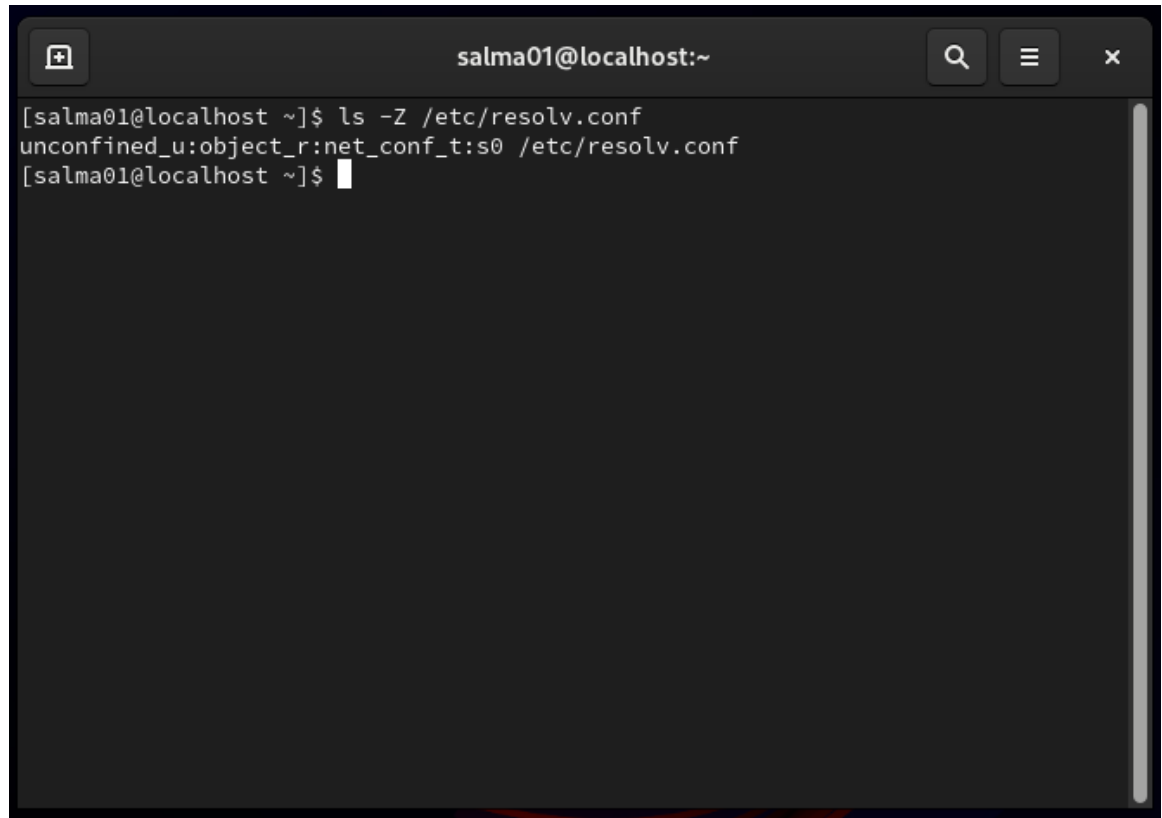

25. Restore the SELinux context of the newly positioned `/etc/resolv.conf`.



```
salma01@localhost:~  
[salma01@localhost ~]$ sudo restorecon /etc/resolv.conf  
[sudo] password for salma01:  
[salma01@localhost ~]$
```

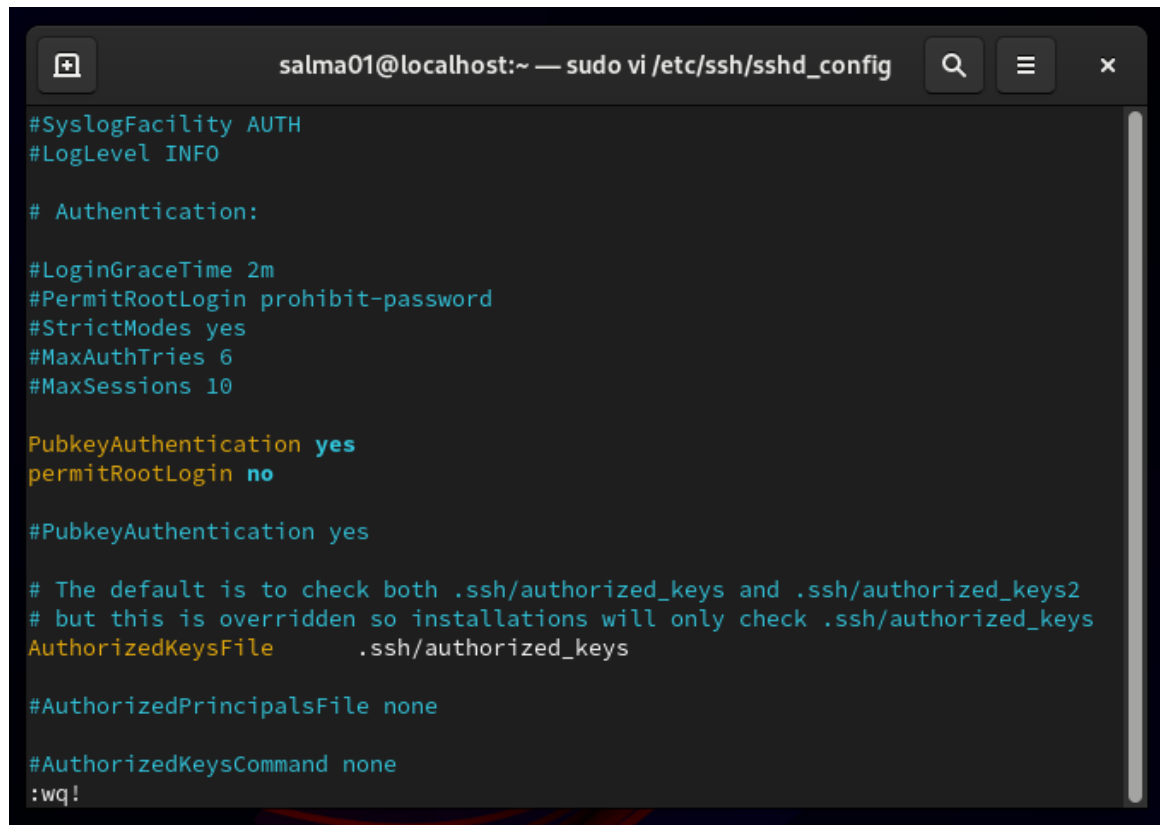
A terminal window with a dark background and light gray text. The window title bar shows 'salma01@localhost:~' and standard window controls (search, menu, close). The terminal content shows a user prompt, a command to restore SELinux context for /etc/resolv.conf, a password prompt, and the command execution completion.

26. Observe the SELinux context of the restored /etc/resolv.conf

A terminal window titled 'salma01@localhost:~' with search, menu, and close buttons. The terminal shows the command 'ls -Z /etc/resolv.conf' and its output 'unconfined_u:object_r:net_conf_t:s0 /etc/resolv.conf'.

```
[salma01@localhost ~]$ ls -Z /etc/resolv.conf
unconfined_u:object_r:net_conf_t:s0 /etc/resolv.conf
[salma01@localhost ~]$
```

27. Configure OpenSSH to allow public key-based login credentials



```
salma01@localhost:~ — sudo vi /etc/ssh/sshd_config

#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes
permitRootLogin no

#PubkeyAuthentication yes

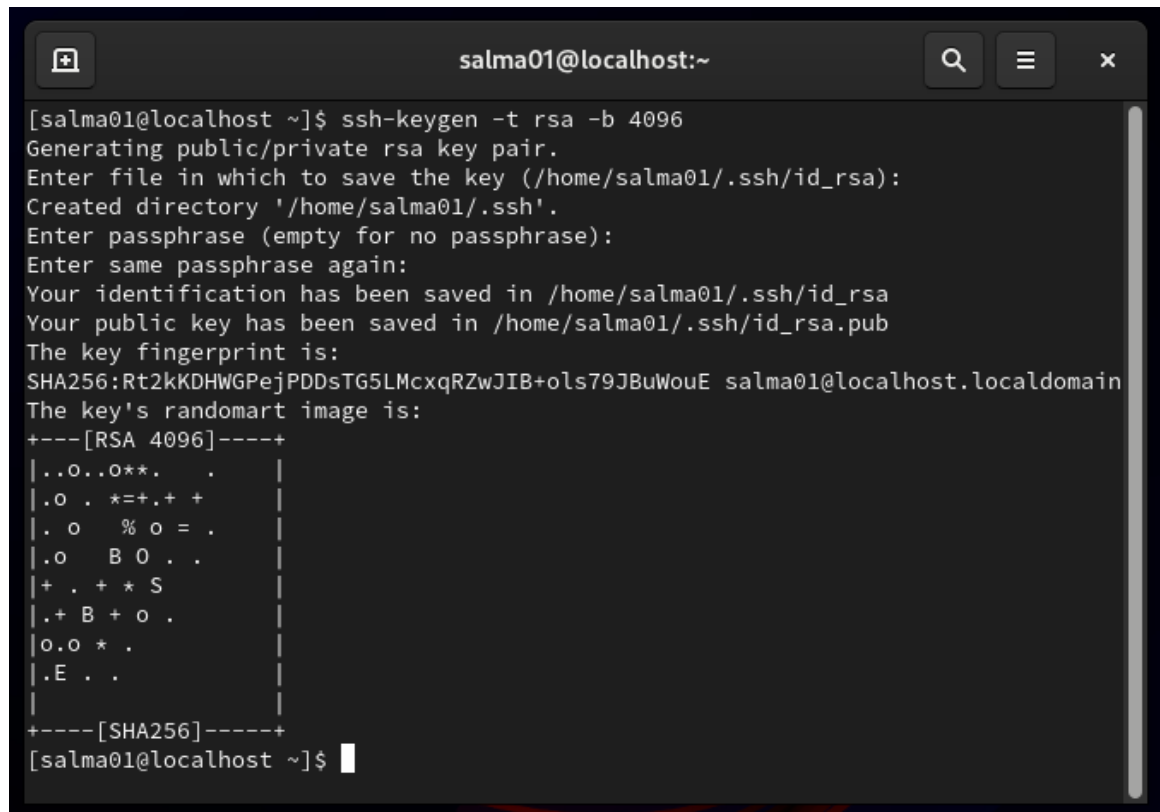
# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none

:wq!
```

28. Create an SSH key-pair.

A terminal window titled 'salma01@localhost:~' with search, menu, and close buttons in the title bar. The terminal shows the execution of 'ssh-keygen -t rsa -b 4096'. It prompts for a file name (defaulting to /home/salma01/.ssh/id_rsa), creates the directory, asks for a passphrase (left empty), and displays the key fingerprint (SHA256:Rt2kKDHWGPejPDDsTG5LMcxqRZwJIB+ols79JBuWouE) and a randomart image for the RSA 4096 key. The window ends with the prompt '[salma01@localhost ~]\$' and a cursor.

```
[salma01@localhost ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/salma01/.ssh/id_rsa):
Created directory '/home/salma01/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/salma01/.ssh/id_rsa
Your public key has been saved in /home/salma01/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Rt2kKDHWGPejPDDsTG5LMcxqRZwJIB+ols79JBuWouE salma01@localhost.localdomain
The key's randomart image is:
+----[RSA 4096]-----+
|..O..O**..      |
|.O . *=+..+ +   |
|. o   % o = .    |
|.o   B O . .     |
|+ . + * S       |
|. + B + o .      |
|o.o * .          |
|.E . .           |
|                 |
+----[SHA256]-----+
[salma01@localhost ~]$
```

29. Configure to login without the need of a password.

```
salma01@localhost:~ — ssh salma01@10.0.2.15
[salma01@localhost ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe38:e75a prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:38:e7:5a txqueuelen 1000 (Ethernet)
    RX packets 32357 bytes 48404412 (46.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12648 bytes 812422 (793.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 49 bytes 4009 (3.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 49 bytes 4009 (3.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[salma01@localhost ~]$ ssh-copy-id salma01@10.0.2.15
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
ED25519 key fingerprint is SHA256:yRBibSFy0fye7r6mH7iWf+lmcWqWltCiQjaI/GszpS4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

```
salma01@localhost:~ — ssh salma01@10.0.2.15
[salma01@localhost ~]$ ssh-copy-id salma01@10.0.2.15
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
ED25519 key fingerprint is SHA256:yRBibSFy0fye7r6mH7iWf+lmcWqWltCiQjaI/GszpS4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
salma01@10.0.2.15's password:

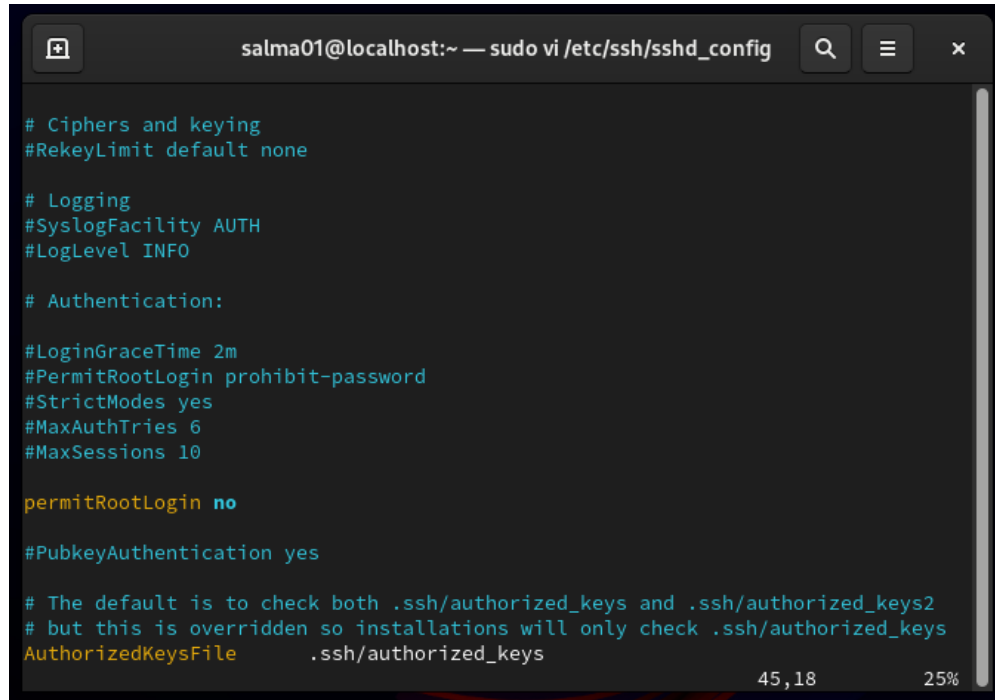
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'salma01@10.0.2.15'"
and check to make sure that only the key(s) you wanted were added.

[salma01@localhost ~]$ ssh salma01@10.0.2.15
Activate the web console with: systemctl enable --now cockpit.socket

Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Wed Dec 13 19:22:45 2023
Hello Salma, Today is Wed Dec 13 08:56:08 PM EET 2023
[salma01@localhost ~]$
```

30. Configure SSH to prevent root logins.



```
salma01@localhost:~ — sudo vi /etc/ssh/sshd_config

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

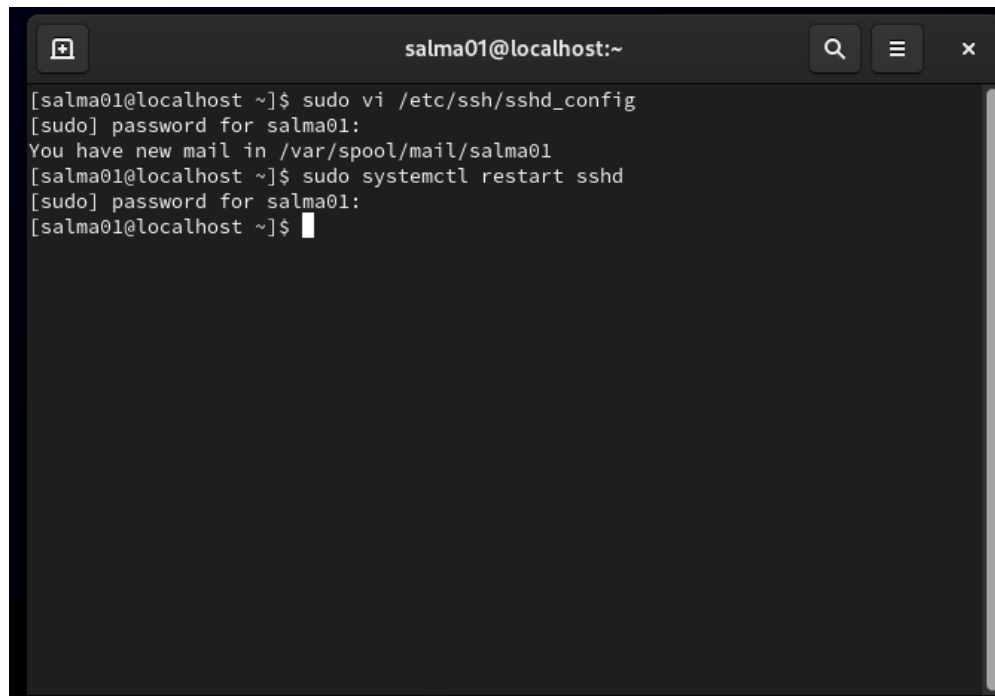
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

permitRootLogin no

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys

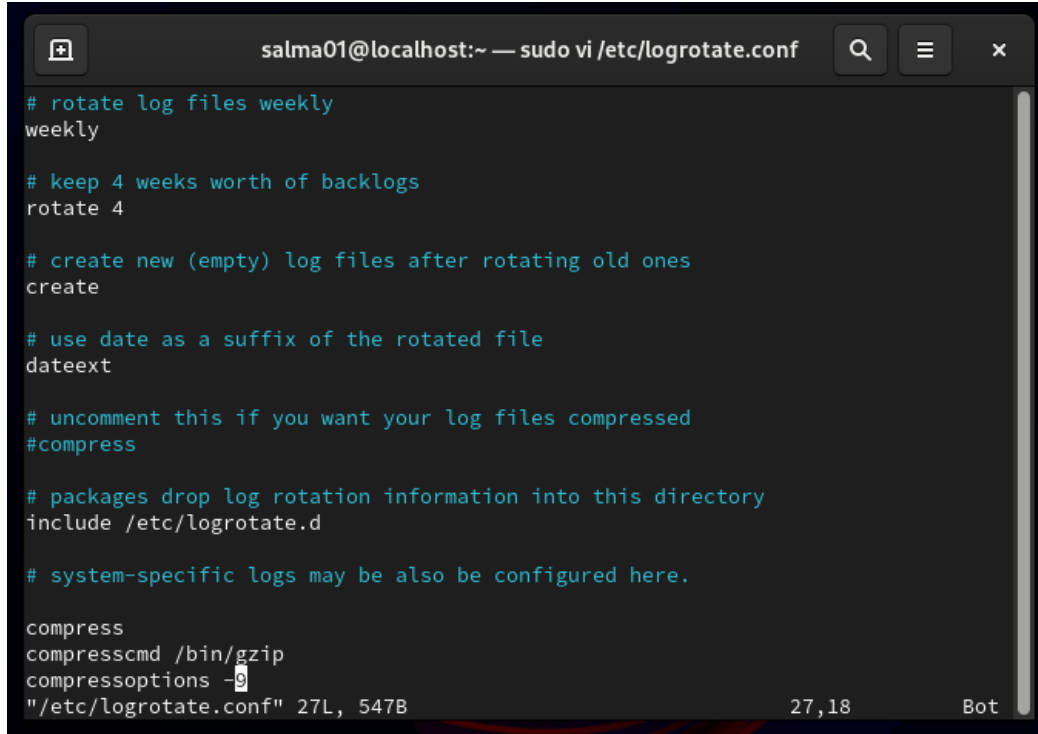
45,18 25%
```



```
salma01@localhost:~

[salma01@localhost ~]$ sudo vi /etc/ssh/sshd_config
[sudo] password for salma01:
You have new mail in /var/spool/mail/salma01
[salma01@localhost ~]$ sudo systemctl restart sshd
[sudo] password for salma01:
[salma01@localhost ~]$
```

31. Configure logrotate default setting to compress log files when they are rotated.



A terminal window titled "salma01@localhost:~ — sudo vi /etc/logrotate.conf". The window shows the contents of the /etc/logrotate.conf file. The configuration includes settings for rotating log files weekly, keeping 4 weeks of backlogs, creating new log files after rotation, and using the date as a suffix. The "compress" option is being edited, with "compresscmd /bin/gzip" and "compressoptions -9" being entered. The status bar at the bottom indicates the cursor is at line 27, column 18, and the file size is 547B.

```
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

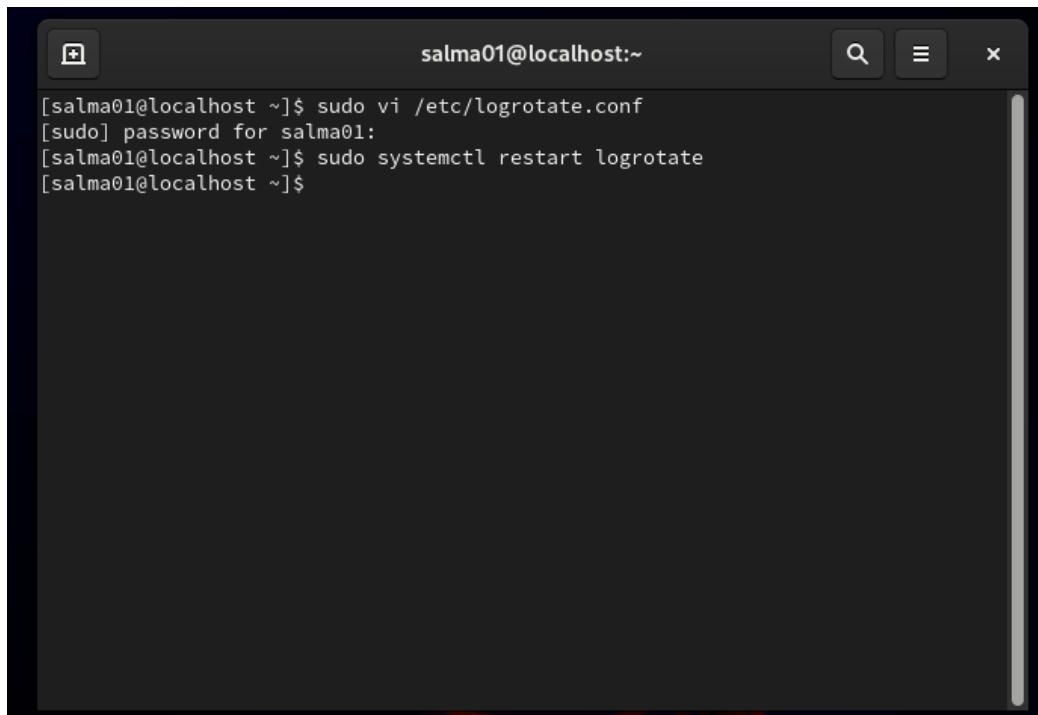
# use date as a suffix of the rotated file
dateext

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# system-specific logs may be also be configured here.

compress
compresscmd /bin/gzip
compressoptions -9
"/etc/logrotate.conf" 27,18 547B Bot
```



A terminal window titled "salma01@localhost:~". The window shows the execution of commands to restart logrotate. The user runs "sudo vi /etc/logrotate.conf", enters the password for "salma01", and then runs "sudo systemctl restart logrotate". The prompt returns to the user's shell.

```
[salma01@localhost ~]$ sudo vi /etc/logrotate.conf
[sudo] password for salma01:
[salma01@localhost ~]$ sudo systemctl restart logrotate
[salma01@localhost ~]$
```