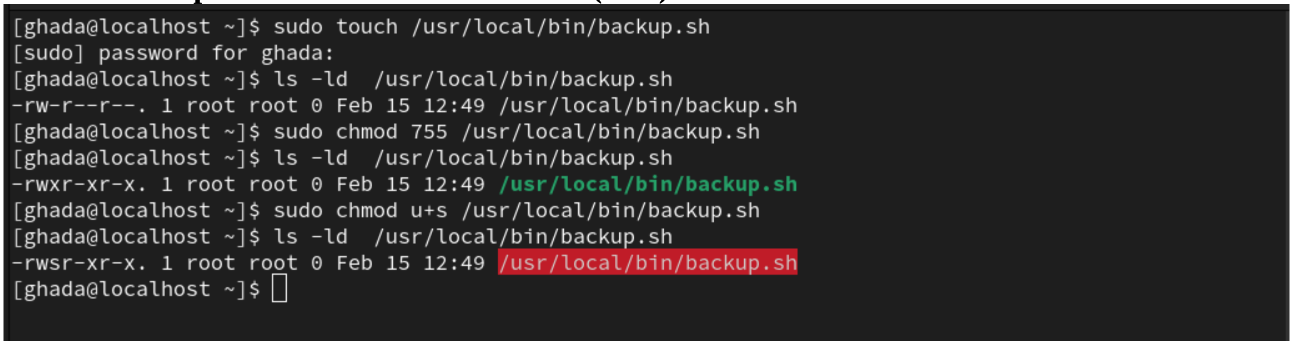
# Lab2 (redhat2) :
# Name : ghada emad bhoul OS

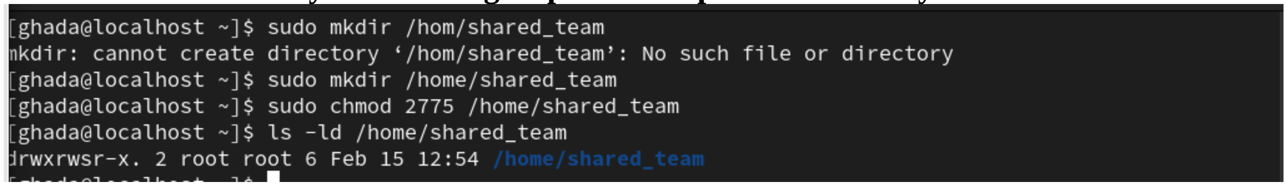
**1. Create a script named backup.sh in /usr/local/bin and Set the SUID bit so that the script runs with the permissions of the file owner (root).**



```
[ghada@localhost ~]$ sudo touch /usr/local/bin/backup.sh
[sudo] password for ghada:
[ghada@localhost ~]$ ls -ld  /usr/local/bin/backup.sh
-rw-r--r--. 1 root root 0 Feb 15 12:49 /usr/local/bin/backup.sh
[ghada@localhost ~]$ sudo chmod 755 /usr/local/bin/backup.sh
[ghada@localhost ~]$ ls -ld  /usr/local/bin/backup.sh
-rwxr-xr-x. 1 root root 0 Feb 15 12:49 /usr/local/bin/backup.sh
[ghada@localhost ~]$ sudo chmod u+s /usr/local/bin/backup.sh
[ghada@localhost ~]$ ls -ld  /usr/local/bin/backup.sh
-rwsr-xr-x. 1 root root 0 Feb 15 12:49 /usr/local/bin/backup.sh
[ghada@localhost ~]$
```

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

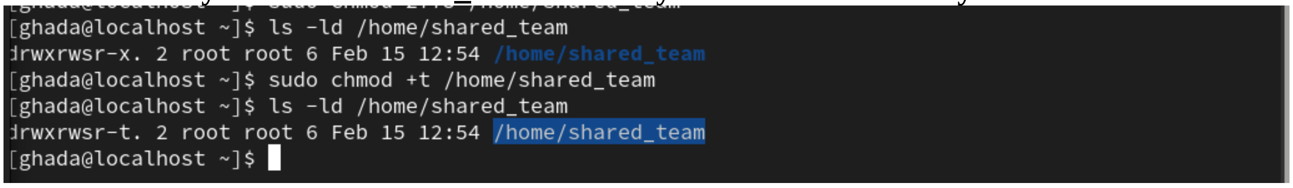
**2. Create a directory named shared_team in /home and Set the SGID bit so that any files created in this directory inherit the group ownership of the directory.**



```
[ghada@localhost ~]$ sudo mkdir /hom/shared_team
mkdir: cannot create directory ‘/hom/shared_team’: No such file or directory
[ghada@localhost ~]$ sudo mkdir /home/shared_team
[ghada@localhost ~]$ sudo chmod 2775 /home/shared_team
[ghada@localhost ~]$ ls -ld /home/shared_team
drwxrwsr-x. 2 root root 6 Feb 15 12:54 /home/shared_team
```

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

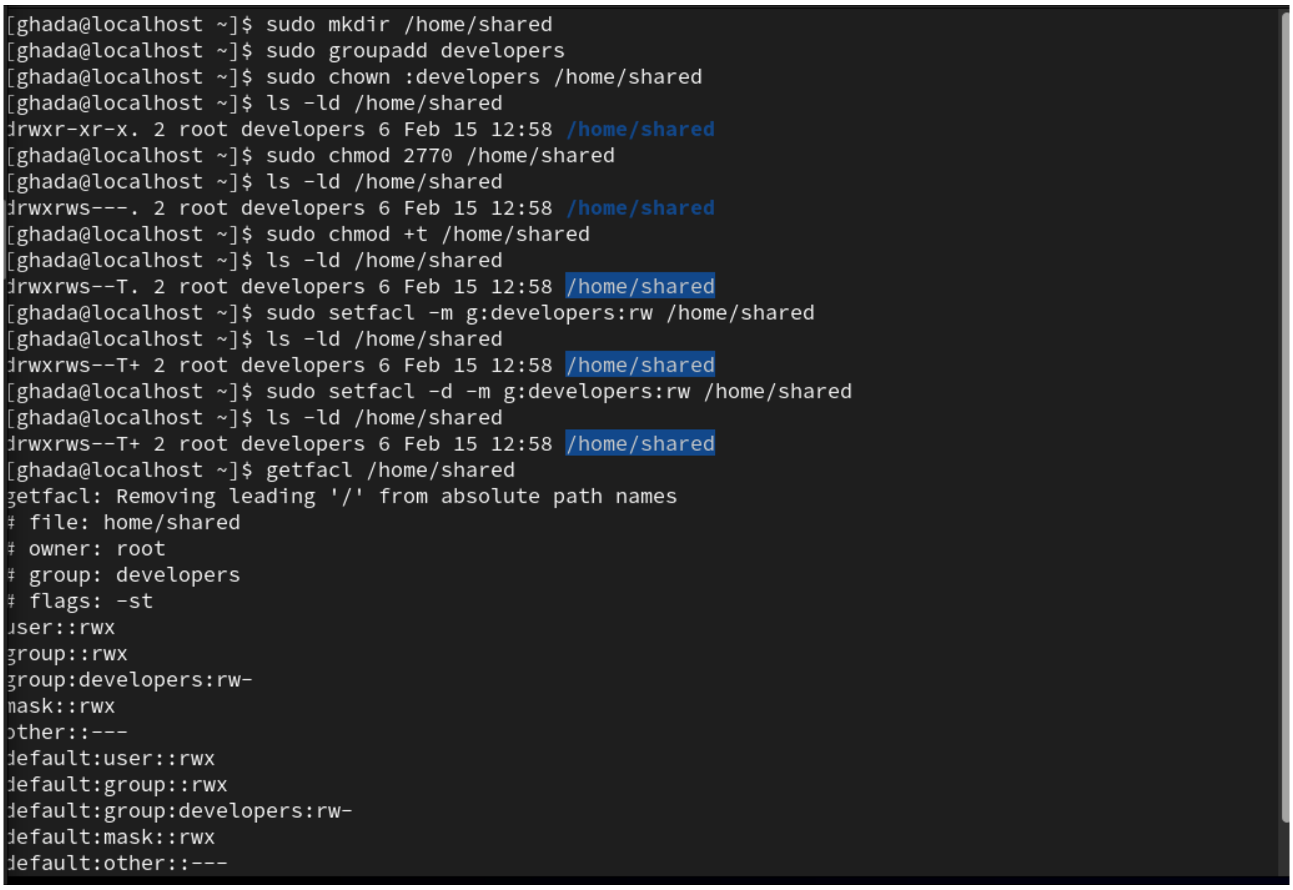
**3. Set the sticky bit on the shared_team directory so that users can only delete their own files.**



```
[ghada@localhost ~]$ ls -ld /home/shared_team
drwxrwsr-x. 2 root root 6 Feb 15 12:54 /home/shared_team
[ghada@localhost ~]$ sudo chmod +t /home/shared_team
[ghada@localhost ~]$ ls -ld /home/shared_team
drwxrwsr-t. 2 root root 6 Feb 15 12:54 /home/shared_team
[ghada@localhost ~]$
```

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

**4.Create a shared directory named shared where:**
**add read and write permissions for a group named developers using ACL .**
**All new files and subdirectories inherit the group developers permissions(use the setgid permession).**
**Only the owner of a file can delete it (use the sticky bit).**



```
[ghada@localhost ~]$ sudo mkdir /home/shared
[ghada@localhost ~]$ sudo groupadd developers
[ghada@localhost ~]$ sudo chown :developers /home/shared
[ghada@localhost ~]$ ls -ld /home/shared
rwxr-xr-x. 2 root developers 6 Feb 15 12:58 /home/shared
[ghada@localhost ~]$ sudo chmod 2770 /home/shared
[ghada@localhost ~]$ ls -ld /home/shared
rwxrws---. 2 root developers 6 Feb 15 12:58 /home/shared
[ghada@localhost ~]$ sudo chmod +t /home/shared
[ghada@localhost ~]$ ls -ld /home/shared
rwxrws--T. 2 root developers 6 Feb 15 12:58 /home/shared
[ghada@localhost ~]$ sudo setfacl -m g:developers:rw /home/shared
[ghada@localhost ~]$ ls -ld /home/shared
rwxrws--T+ 2 root developers 6 Feb 15 12:58 /home/shared
[ghada@localhost ~]$ sudo setfacl -d -m g:developers:rw /home/shared
[ghada@localhost ~]$ ls -ld /home/shared
rwxrws--T+ 2 root developers 6 Feb 15 12:58 /home/shared
[ghada@localhost ~]$ getfacl /home/shared
etfacl: Removing leading '/' from absolute path names
# file: home/shared
# owner: root
# group: developers
# flags: -st
ser::rwx
roup::rwx
roup:developers:rw-
ask::rwx
ther::---
efault:user::rwx
efault:group::rwx
efault:group:developers:rw-
efault:mask::rwx
efault:other::---
```

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

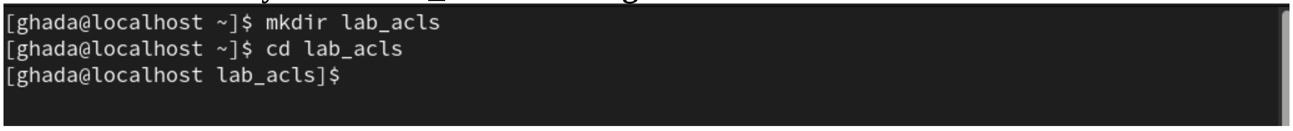
**5.What is the difference between traditional Linux permissions and ACLs?**

Traditional Linux permissions (rwx) only allow setting permissions for owner, group, and others.

ACLs (Access Control Lists) allow more granular control, letting you assign permissions to specific users or groups.

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

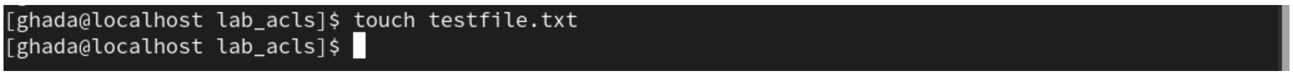
**6.Create a directory named lab_acls and navigate into it:**



```
[ghada@localhost ~]$ mkdir lab_acls
[ghada@localhost ~]$ cd lab_acls
[ghada@localhost lab_acls]$
```

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

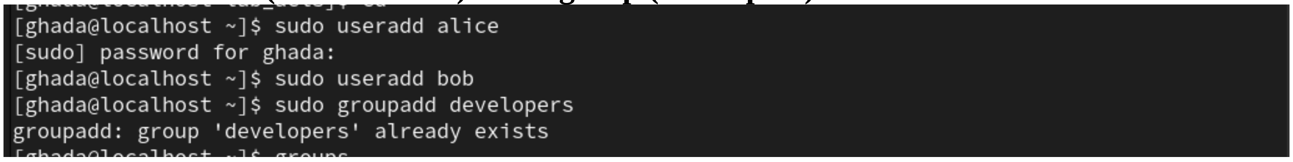
**7.Create a file named testfile.txt:**



```
[ghada@localhost lab_acls]$ touch testfile.txt
[ghada@localhost lab_acls]$ 
```

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

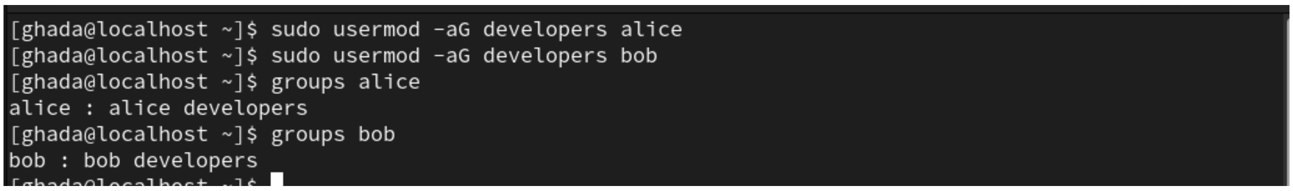
**8.Create two users (alice and bob) and a group (developers):**



```
[ghada@localhost ~]$ sudo useradd alice
[sudo] password for ghada:
[ghada@localhost ~]$ sudo useradd bob
[ghada@localhost ~]$ sudo groupadd developers
groupadd: group 'developers' already exists
```

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

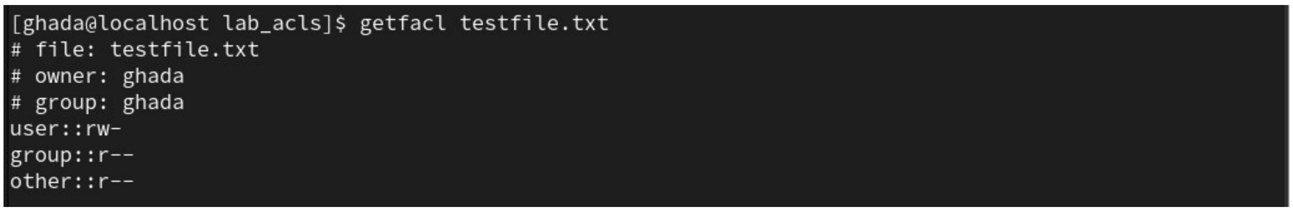
**9.Add alice and bob to the developers group:**



```
[ghada@localhost ~]$ sudo usermod -aG developers alice
[ghada@localhost ~]$ sudo usermod -aG developers bob
[ghada@localhost ~]$ groups alice
alice : alice developers
[ghada@localhost ~]$ groups bob
bob : bob developers
```

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

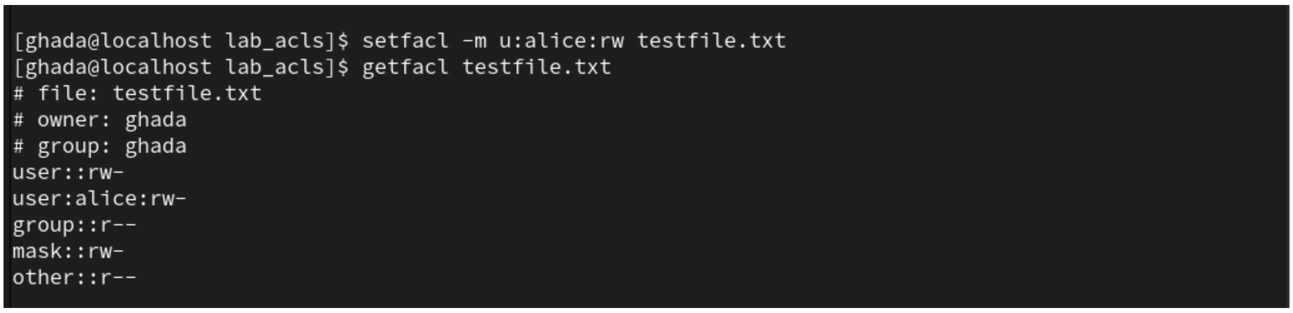
**10.View and List the ACL of a file named testfile.txt.**



```
[ghada@localhost lab_acls]$ getfacl testfile.txt
# file: testfile.txt
# owner: ghada
# group: ghada
user::rw-
group::r--
other::r--
```

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

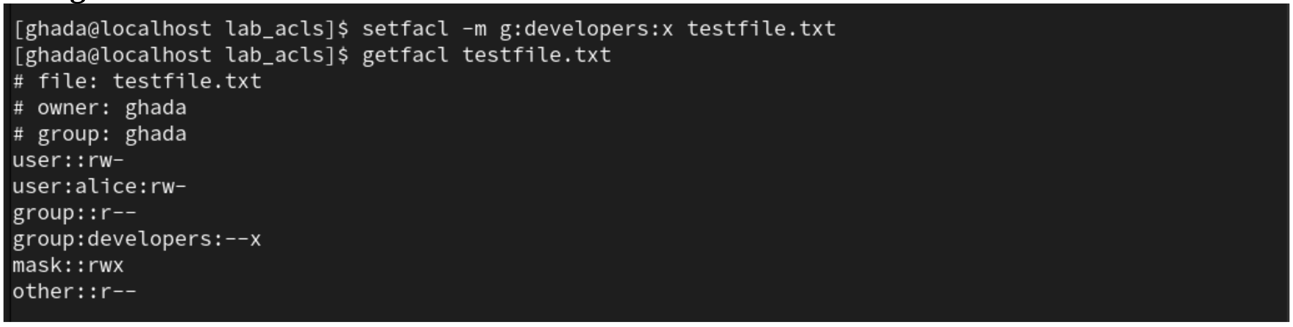
**11.read and write permissions for a user named alice to the file testfile.txt and Verify the changes.**



```
[ghada@localhost lab_acls]$ setfacl -m u:alice:rw testfile.txt
[ghada@localhost lab_acls]$ getfacl testfile.txt
# file: testfile.txt
# owner: ghada
# group: ghada
user::rw-
user:alice:rw-
group::r--
mask::rw-
other::r--
```

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

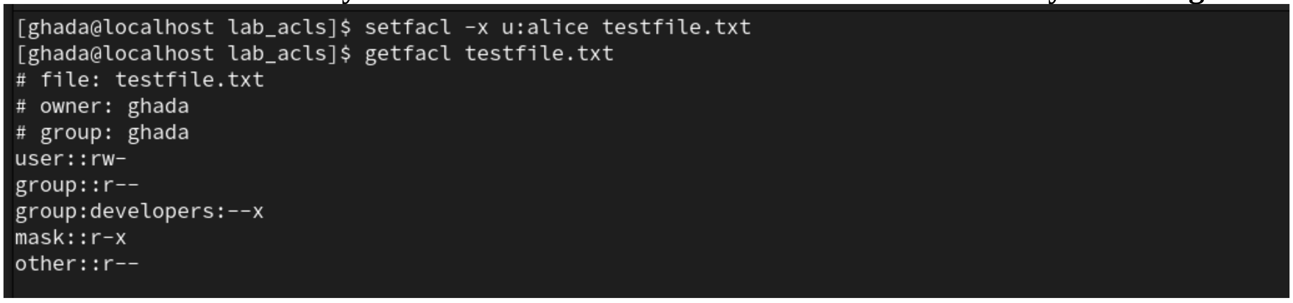
**12.Add execute permission for a group named developers to the file testfile.txt and Verify the changes.**



```
[ghada@localhost lab_acls]$ setfacl -m g:developers:x testfile.txt
[ghada@localhost lab_acls]$ getfacl testfile.txt
# file: testfile.txt
# owner: ghada
# group: ghada
user::rw-
user:alice:rw-
group::r--
group:developers:--x
mask::rwx
other::r--
```

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

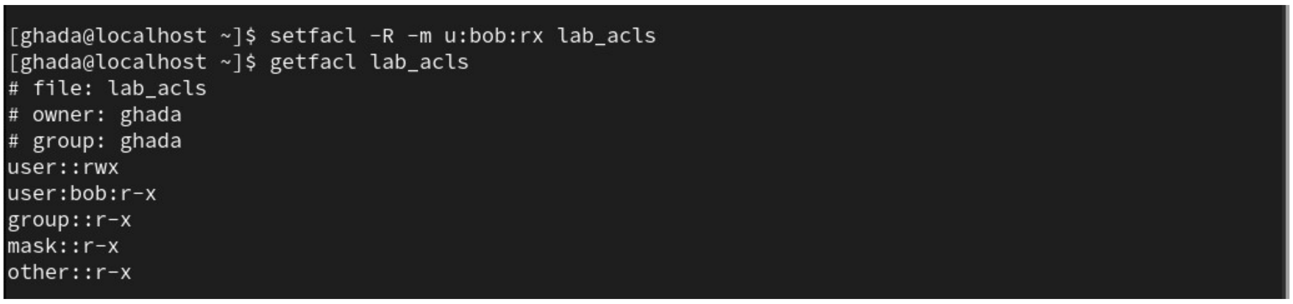
**13.Remove the ACL entry for the user alice from the file testfile.txt and Verify the changes.**



```
[ghada@localhost lab_acls]$ setfacl -x u:alice testfile.txt
[ghada@localhost lab_acls]$ getfacl testfile.txt
# file: testfile.txt
# owner: ghada
# group: ghada
user::rw-
group::r--
group:developers:--x
mask::r-x
other::r--
```

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

**14.Set read and execute permissions for bob on all files and subdirectories inside lab_acls.**



```
[ghada@localhost ~]$ setfacl -R -m u:bob:rx lab_acls
[ghada@localhost ~]$ getfacl lab_acls
# file: lab_acls
# owner: ghada
# group: ghada
user::rwx
user:bob:r-x
group::r-x
mask::r-x
other::r-x
```

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

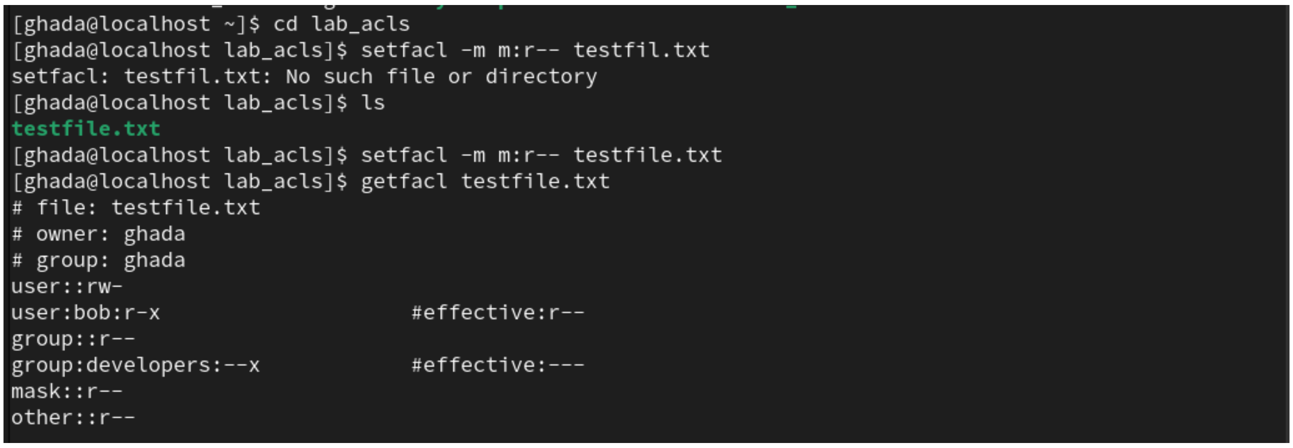
**15.How does the mask affect the effective permissions of named users and groups?**

The mask determines the maximum permissions allowed for users and groups in ACL.

If a named user/group has `rwx` but the mask is `r--`, the effective permission will be `r--`.

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

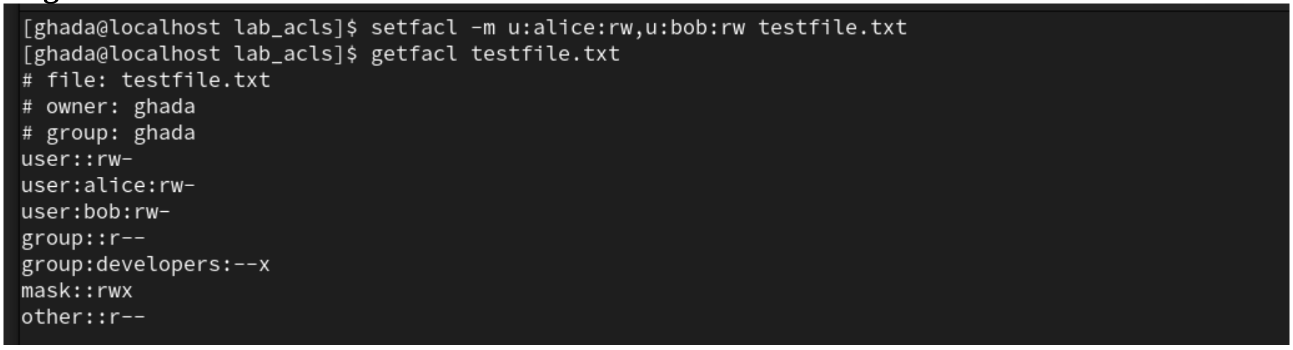
**16.Set the mask for the file testfile.txt to r-- and observe how it affects the effective permissions of named users and groups.**



```
[ghada@localhost ~]$ cd lab_acls
[ghada@localhost lab_acls]$ setfacl -m m:r-- testfil.txt
setfacl: testfil.txt: No such file or directory
[ghada@localhost lab_acls]$ ls
testfile.txt
[ghada@localhost lab_acls]$ setfacl -m m:r-- testfile.txt
[ghada@localhost lab_acls]$ getfacl testfile.txt
# file: testfile.txt
# owner: ghada
# group: ghada
user::rw-
user:bob:r-x                    #effective:r--
group::r--
group:developers:--x            #effective:---
mask::r--
other::r--
```

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

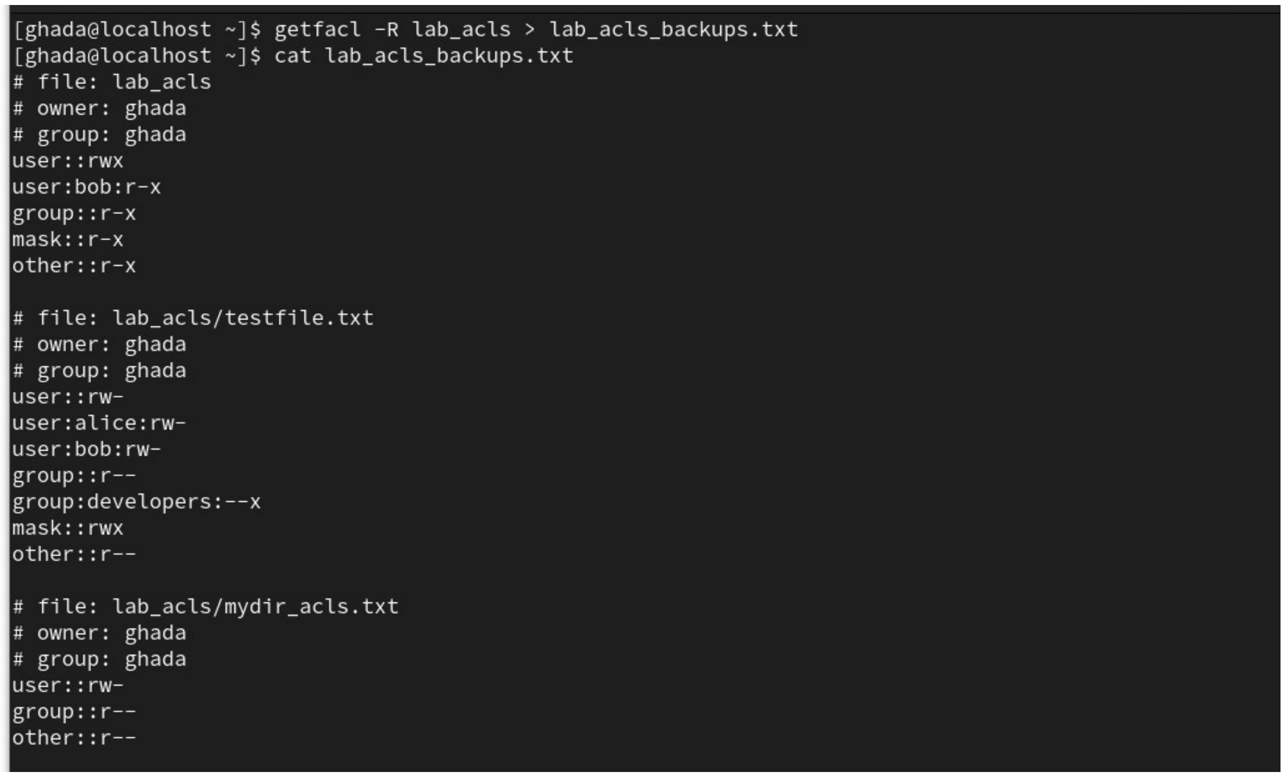
**17.Add read and write permissions for two users, alice and bob, to the file testfile.txt in a single command.**



```
[ghada@localhost lab_acls]$ setfacl -m u:alice:rw,u:bob:rw testfile.txt
[ghada@localhost lab_acls]$ getfacl testfile.txt
# file: testfile.txt
# owner: ghada
# group: ghada
user::rw-
user:alice:rw-
user:bob:rw-
group::r--
group:developers:--x
mask::rwx
other::r--
```

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

**18.Backup the ACLs of the directory mydir to a file named mydir_acls.txt.**



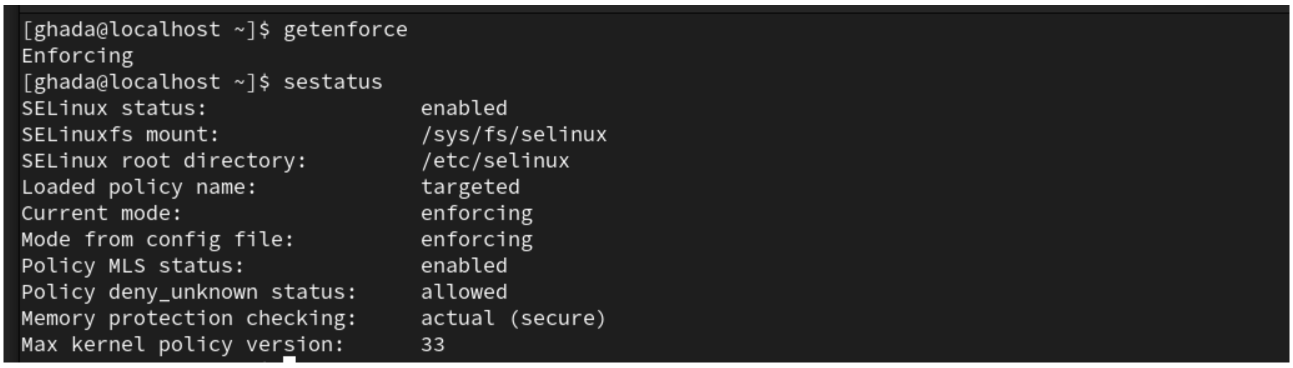
```
[ghada@localhost ~]$ getfacl -R lab_acls > lab_acls_backups.txt
[ghada@localhost ~]$ cat lab_acls_backups.txt
# file: lab_acls
# owner: ghada
# group: ghada
user::rwx
user:bob:r-x
group::r-x
mask::r-x
other::r-x

# file: lab_acls/testfile.txt
# owner: ghada
# group: ghada
user::rw-
user:alice:rw-
user:bob:rw-
group::r--
group:developers:--x
mask::rwx
other::r--

# file: lab_acls/mydir_acls.txt
# owner: ghada
# group: ghada
user::rw-
group::r--
other::r--
```

**-------------------------------------------------------------------------------------------**

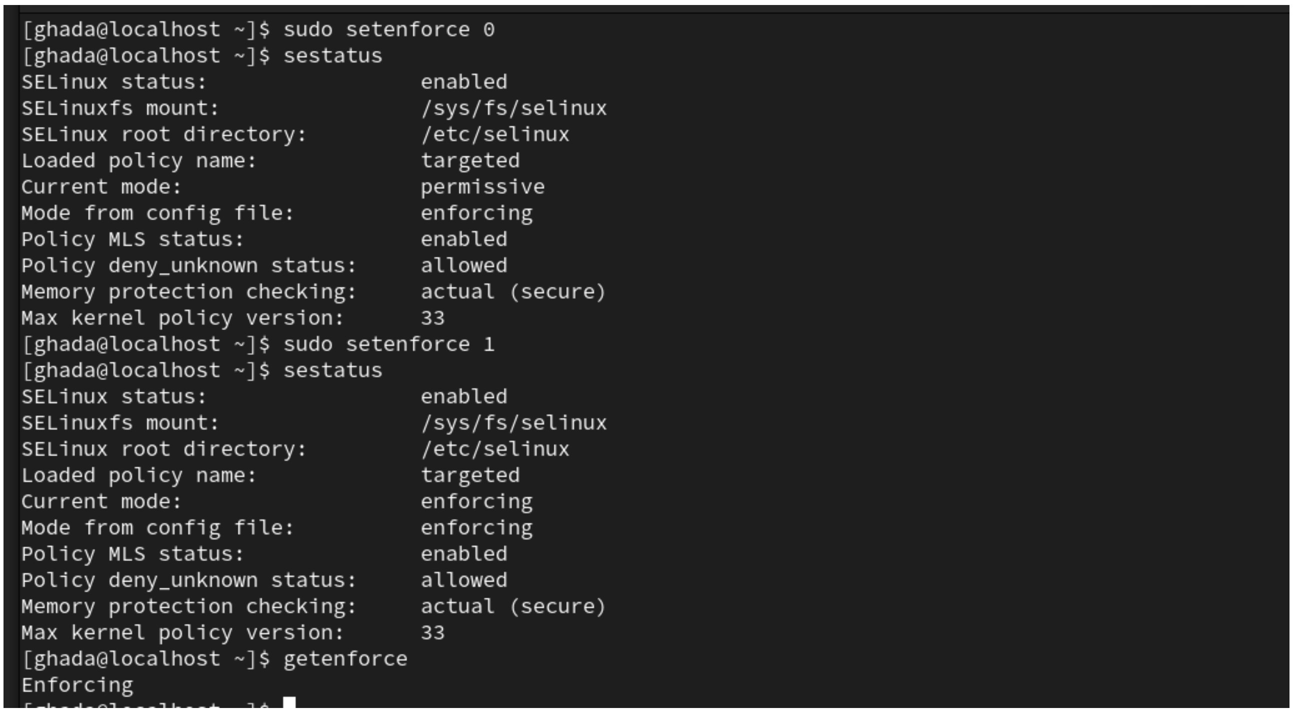
**19.Check the Current SELinux Mode.**



```
[ghada@localhost ~]$ getenforce
Enforcing
[ghada@localhost ~]$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

**-------------------------------------------------------------------------------------------**

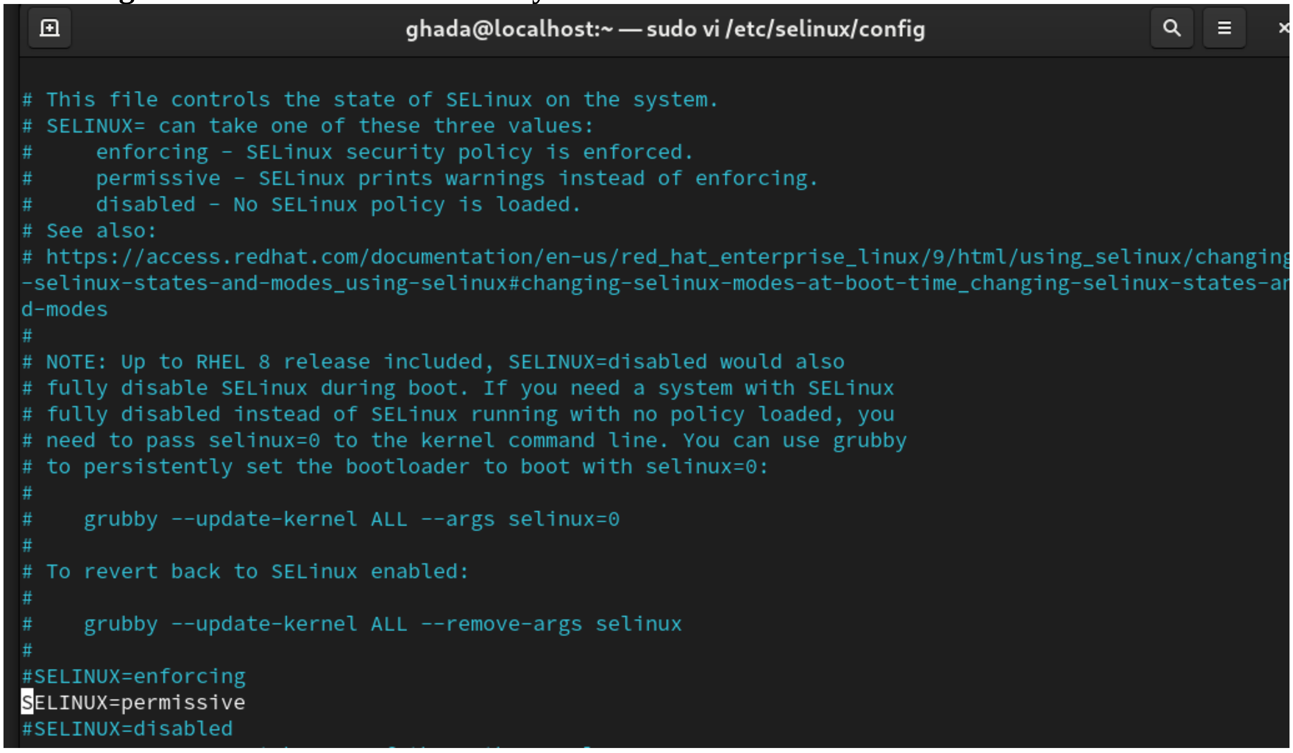
**20.Change SELinux mode temporarily.**



```
[ghada@localhost ~]$ sudo setenforce 0
[ghada@localhost ~]$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   permissive
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
[ghada@localhost ~]$ sudo setenforce 1
[ghada@localhost ~]$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
[ghada@localhost ~]$ getenforce
Enforcing
```

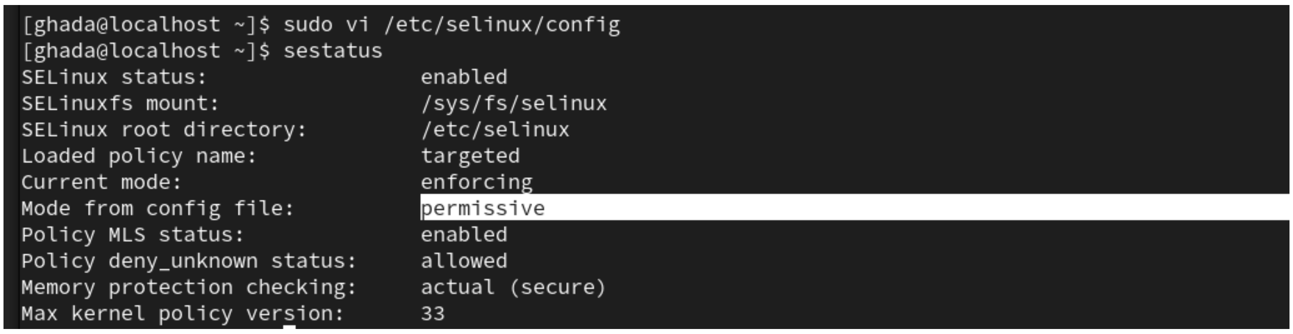
-------------------------------------------------------------------------------------

**21.Change SELinux Mode Permanently.**



```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
# See also:
# https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/using_selinux/changing
-selinux-states-and-modes_using-selinux#changing-selinux-modes-at-boot-time_changing-selinux-states-ar
d-modes
#
# NOTE: Up to RHEL 8 release included, 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
SELINUX=permissive
#SELINUX=disabled
```

**Enforcing → fully enabled**
**permissive → warnings only, no enforcement**
**disabled → turns Selinux off ( and require reboot).**



```
[ghada@localhost ~]$ sudo vi /etc/selinux/config
[ghada@localhost ~]$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          permissive
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

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

**22.what different between cp , mv ,cp -a (ContextSwitching).**

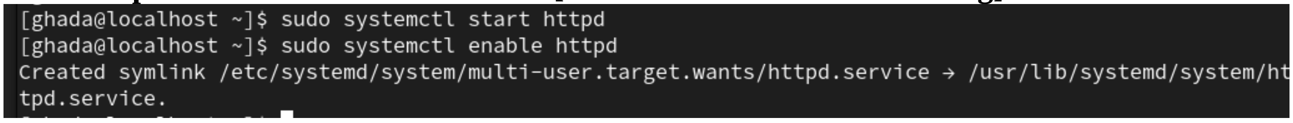
**Cp : copies files but doesn’t preserve selinux context and new file gets the default context of the destination**

**mv : moves files without changing selinux context. Keeps the orignial selinux context**

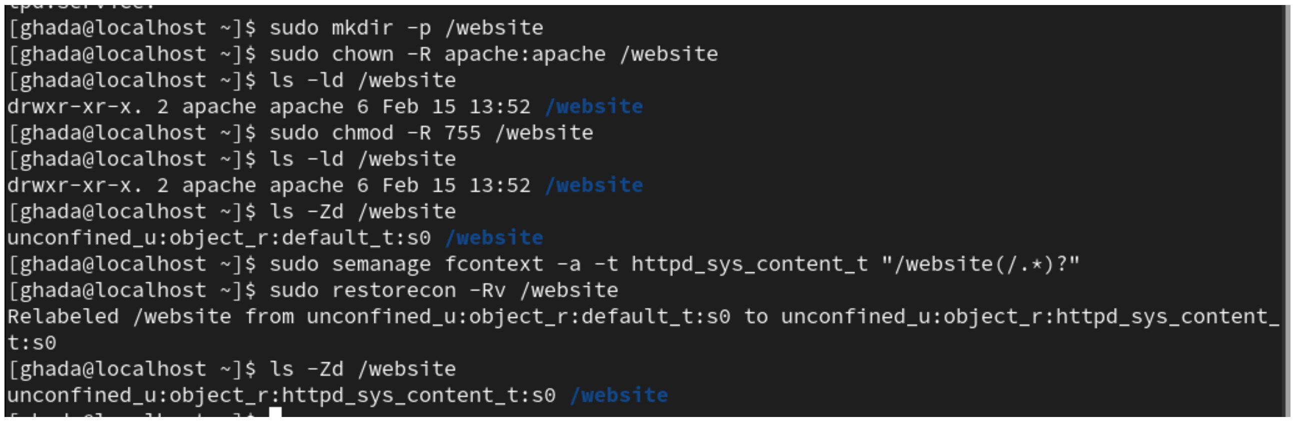
**cp -a : copies files and preserves selinux context , maintains the original selinux labels.**

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

**23.Run Apache web Server on /websites [must SELinuxMode=Enforcing]**

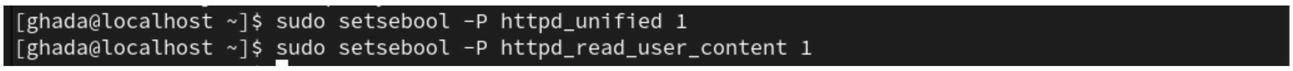


```
[ghada@localhost ~]$ sudo systemctl start httpd
[ghada@localhost ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/ht
tpd.service.
```



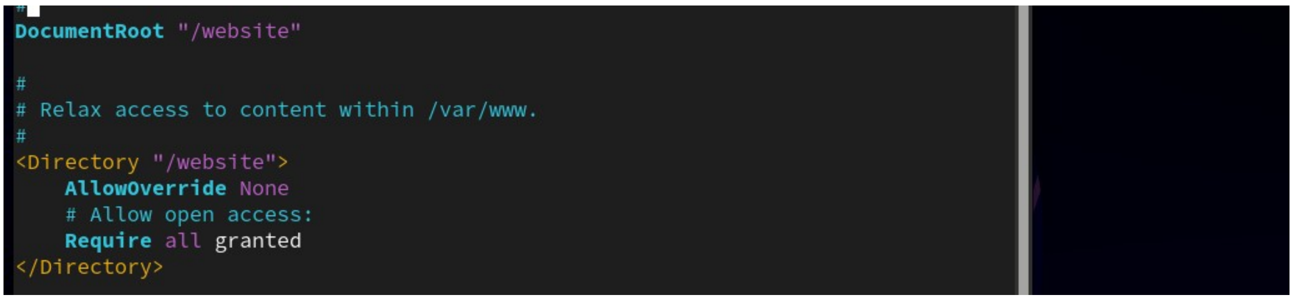
```
[ghada@localhost ~]$ sudo mkdir -p /website
[ghada@localhost ~]$ sudo chown -R apache:apache /website
[ghada@localhost ~]$ ls -ld /website
drwxr-xr-x. 2 apache apache 6 Feb 15 13:52 /website
[ghada@localhost ~]$ sudo chmod -R 755 /website
[ghada@localhost ~]$ ls -ld /website
drwxr-xr-x. 2 apache apache 6 Feb 15 13:52 /website
[ghada@localhost ~]$ ls -Zd /website
unconfined_u:object_r:default_t:s0 /website
[ghada@localhost ~]$ sudo semanage fcontext -a -t httpd_sys_content_t "/website(/.*)?"
[ghada@localhost ~]$ sudo restorecon -Rv /website
Relabeled /website from unconfined_u:object_r:default_t:s0 to unconfined_u:object_r:httpd_sys_content_
t:s0
[ghada@localhost ~]$ ls -Zd /website
unconfined_u:object_r:httpd_sys_content_t:s0 /website
```

**allow apache to accesss directories**



```
[ghada@localhost ~]$ sudo setsebool -P httpd_unified 1
[ghada@localhost ~]$ sudo setsebool -P httpd_read_user_content 1
```

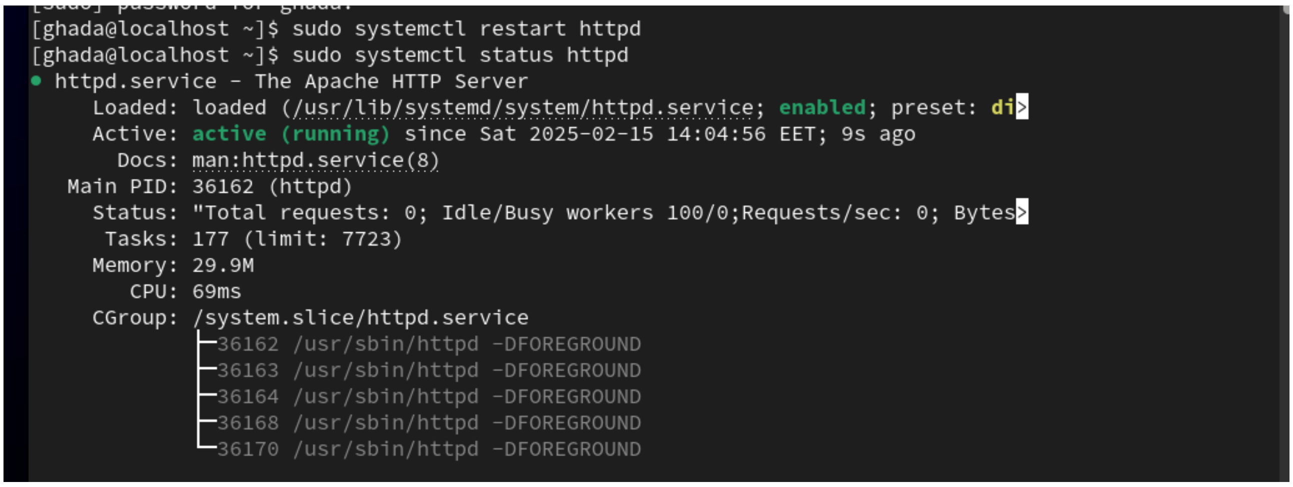
**configure apache to use /website directory**
**in /etc/httpd/conf/httpd.conf**



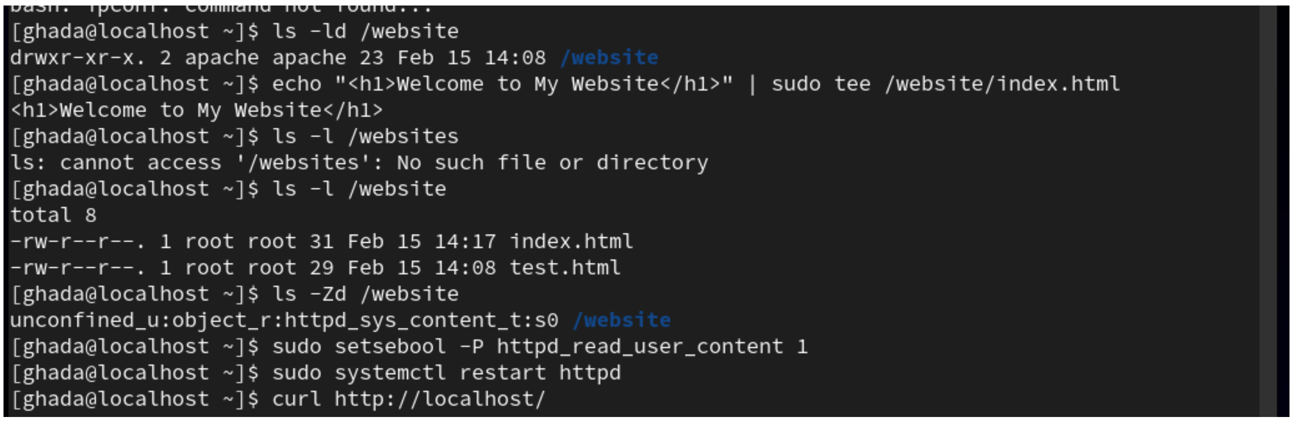
```
#
DocumentRoot "/website"

#
# Relax access to content within /var/www.
#
<Directory "/website">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>
```

## Restart Apache and Verify



```
[sudo] password for ghada:
[ghada@localhost ~]$ sudo systemctl restart httpd
[ghada@localhost ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: di>
     Active: active (running) since Sat 2025-02-15 14:04:56 EET; 9s ago
       Docs: man:httpd.service(8)
   Main PID: 36162 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes>
      Tasks: 177 (limit: 7723)
     Memory: 29.9M
        CPU: 69ms
     CGroup: /system.slice/httpd.service
             ├─36162 /usr/sbin/httpd -DFOREGROUND
             ├─36163 /usr/sbin/httpd -DFOREGROUND
             ├─36164 /usr/sbin/httpd -DFOREGROUND
             ├─36168 /usr/sbin/httpd -DFOREGROUND
             └─36170 /usr/sbin/httpd -DFOREGROUND
```



```
bash: ipconf: command not found...
[ghada@localhost ~]$ ls -ld /website
drwxr-xr-x. 2 apache apache 23 Feb 15 14:08 /website
[ghada@localhost ~]$ echo "<h1>Welcome to My Website</h1>" | sudo tee /website/index.html
<h1>Welcome to My Website</h1>
[ghada@localhost ~]$ ls -l /websites
ls: cannot access '/websites': No such file or directory
[ghada@localhost ~]$ ls -l /website
total 8
-rw-r--r--. 1 root root 31 Feb 15 14:17 index.html
-rw-r--r--. 1 root root 29 Feb 15 14:08 test.html
[ghada@localhost ~]$ ls -Zd /website
unconfined_u:object_r:httpd_sys_content_t:s0 /website
[ghada@localhost ~]$ sudo setsebool -P httpd_read_user_content 1
[ghada@localhost ~]$ sudo systemctl restart httpd
[ghada@localhost ~]$ curl http://localhost/
```