Contents

[**ACL: Access Control List** 2](#_Toc481333731)

[ **Need of ACL:** 2](#_Toc481333732)

[ To set ACL for a file 2](#_Toc481333733)

[ To see status of ACL of file 2](#_Toc481333734)

[ **Sudoers:** 3](#_Toc481333735)

[ **Explanation about users’ access in /etc/sudoers file:** 3](#_Toc481333736)

[ **Enable file system with ACL option:** 4](#_Toc481333737)

[ **ACL with more options:** 5](#_Toc481333738)

[1. Remove all ACL rules from specific file/directory 5](#_Toc481333739)

[2. Set ACL rules recursively 5](#_Toc481333740)

[3. Set same ACL rules on newly created files within specified directory 5](#_Toc481333741)

[4. Remove specific ACL rules from specific file/directory 5](#_Toc481333742)

# **ACL: Access Control List**

* ACL is an extended set of permissions.

## **Need of ACL:**

* Every Linux file has owner, group and others with read, write and execute permissions (POSIX permissions).
* If multiple users need access to same file without being in a same group and those users are from different groups, FACL (File Access Control List) will come into picture.
* It allows users to own read, write and execute permission for the same file individually as per need.
* To make sure if file has attached ACL,

Command: ls -l <filename>

If **+** sign at the end of permissions, it means file has an ACL attached to it.



* To set ACL for a file,

Command: setfacl -m u(for user)/g(for group):<username>:<permission> <filename/path>

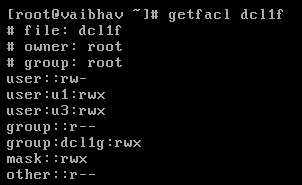
Example: setfacl -m u:u1:rwx dcl1f

setfacl -m g:dcl1g:rwx dcl1f

setfacl -m u:u1:rwx,g:dcl1g:rwx dcl1f

* To see status of ACL of file,

Command: getfacl <filename>



* When you try to access this file by specified users (u1 and u3), it will throw an error as user is not in sudoers file.
* To solve this problem, specified users need to be added in sudoers file.

## **Sudoers:**

* It is a security policy module that determines user’s sudo privileges.
* It is located in /etc/sudoers file. This file contains rules that users must follow when using sudo command.
* This file cannot be edited by normal text editors.
* It must be edited by **visudo** command.
* To solve that problem, add user’s information at the end of file.
* For an example, we want to allow two users who can access sudo.

u1 ALL=(ALL) ALL

u3 ALL=(ALL) ALL

### **Explanation about users’ access in /etc/sudoers file:**

ALL ALL= (ALL:ALL) ALL

1st 2nd 3rd 4th

If user don’t want password to be asked,

ALL ALL= (ALL:ALL) ALL NOPASSWD: ALL

* 1st: name of the user/group

If 1st is appended by percentage (%) then it is considered as group otherwise it is considered as user.

2nd: host/terminal/machine

3rd: act as specified user/group (user: group)

4th: specified commands to allow

* Example: root ALL=(ALL) ALL

It indicates that root user (1st) can run/execute all commands (4th) from all terminals/hosts/machines (2nd), acting as any user (3rd).

* Example: u1 ALL=(ALL) ALL

It indicates that user u1 (1st) can run/execute all commands (4th) from all terminals/hosts/machines (2nd), acting as any user (3rd).

* Example: %wheel ALL=(ALL) ALL

It indicates that users from group wheel (1st) can run/execute all commands (4th) from all terminals/hosts/machines (2nd), acting as any user (3rd).

* Example: linus ALL=(OP) ALL

It indicates that user linus (1st) can run/execute all commands (4th) from all terminals/hosts/machines (2nd), acting as users of OP group (3rd).

* Example: linus OFNET=(ALL) ALL

It indicates that user linus (1st) can run/execute all commands (4th) from any terminal/host/machine in OFNET network (2nd), acting as any user (3rd).

* File system must enable ACL in order to implement ACL in system.
* ACL can be enabled in file system by modifying /etc/fstab file.

## **Enable file system with ACL option:**

* Command: vi /etc/fstab
* Add acl in required file system as follows:
* After editing /etc/fstab file, remount relevant file system.

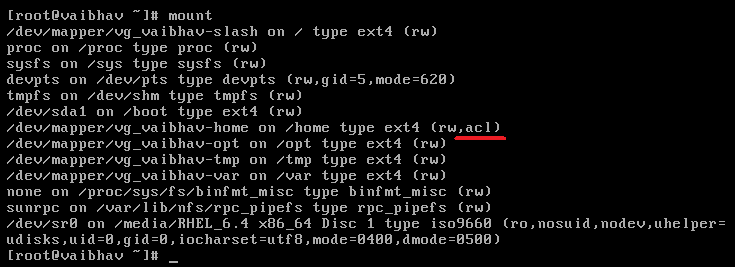
Command: mount -o remount <file system>

Example: mount -o remount /dev/mapper/vg\_vaibhav-home



* We can also verify whether file system has acl option enabled or not with mount command.

Command: mount



## **ACL with more options:**

* Say, we have two users u1 and directory vbhv. There are two files v1 and v2 in that directory.

### Remove all ACL rules from specific file/directory

Command: setfacl **-b** <file/directory>

Example: setfacl -**b** vbhv/

### Set ACL rules recursively

It means ACL rule will be applied on sub files within the directory as well

Command: setfacl -**R**m u(for user)/g(for group):<username>:<permission> <filename/path>

Example: setfacl -**R**m u:u1:rwx vbhv/

### Set same ACL rules on newly created files within specified directory

Command: setfacl -**d**m u(for user)/g(for group):<username>:<permission> <filename/path>

Example: setfacl -**d**m u:u1:rwx vbhv/

### Remove specific ACL rules from specific file/directory

Command: setfacl **-x** <file/directory>

Example: setfacl -**x** u:u1:rxv bhv/