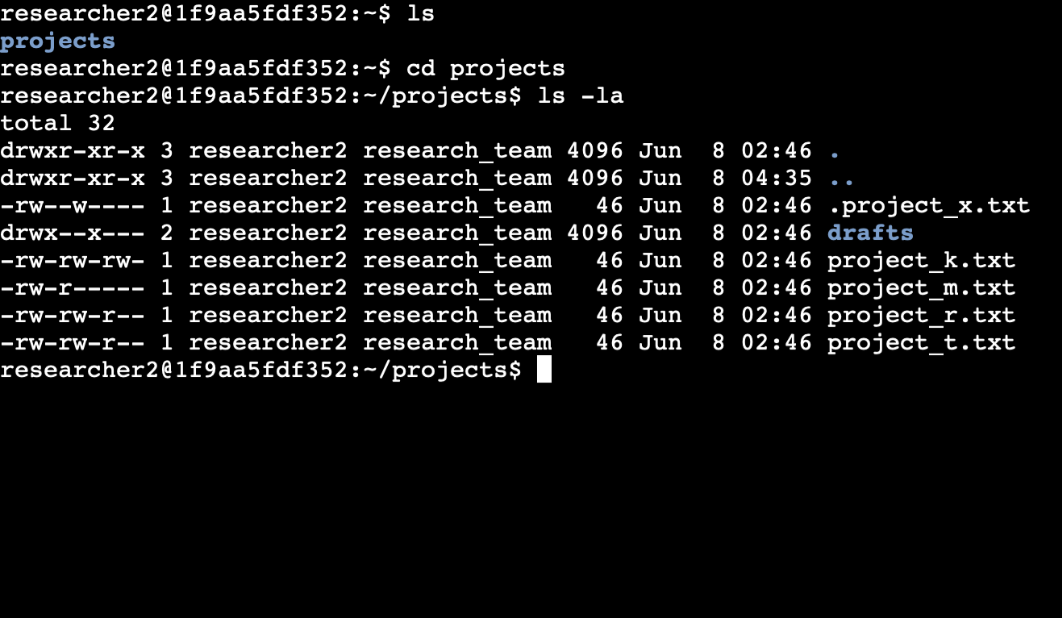
# File permissions in Linux

## Project description

The research team at my organization needs to update the file permissions for certain files and directories within the projects directory. The permissions do not currently reflect the level of authorization that should be given. Checking and updating these permissions will help keep their system secure. To complete this task, I performed the following tasks:

## Check file and directory details

The following code demonstrates how I used Linux commands to determine the existing permissions set for a specific directory in the file system.



The first line of the screenshot displays the command I entered, ls -la command to check permissions for files and subdirectories in the project directory, even hidden files. The output of my command indicates that there is one directory named drafts, one hidden file .project\_x.txt and five other project files.

## Describe the permissions string

The 10-character string shows the different types of owners(user, group, and others) and the permissions that they have(read, write, and execute). For example if you look at project\_m.txt it reads -rw-r-----it is broken down into three different owner types.

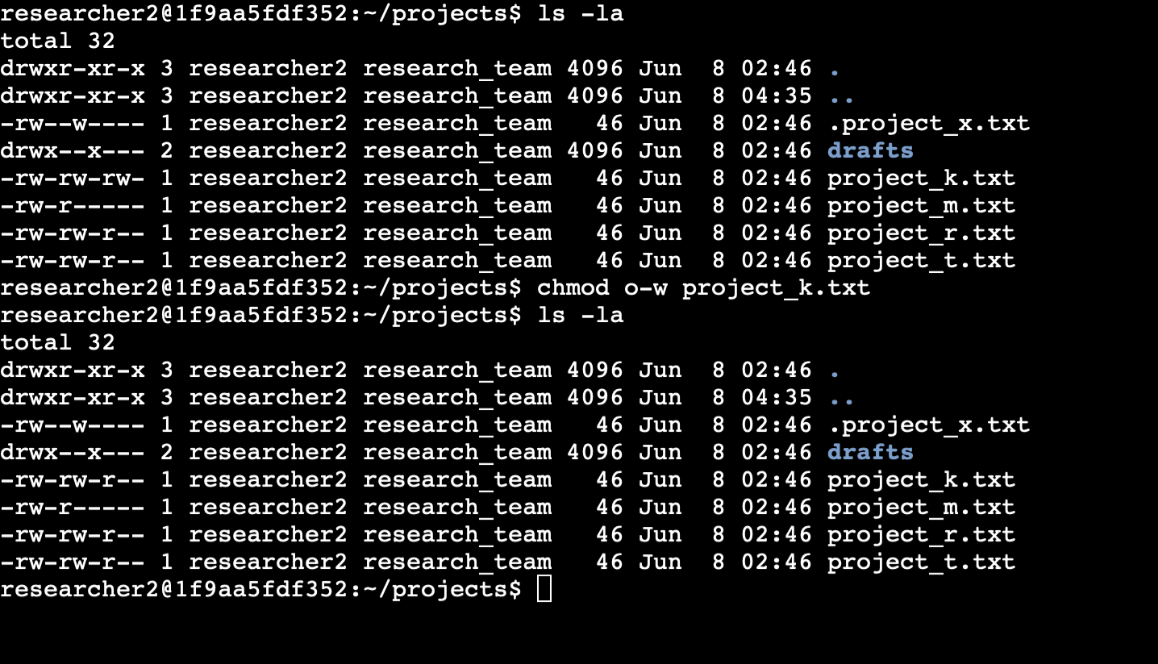
* 1st character: This character is either a d or hyphen, and indicates the file type. If it’s a d, it’s a directory. If it’s a hyphen, it’s a regular file.
* 2nd-4th character: These characters indicated the read, write, and execute (r,w,x) permissions for the user. If there is a hyphen (-) in place then it indicates that the user doesn’t have that permission.
* 5nd-7th character: These characters indicated the read, write, and execute (r,w,x) permissions for the group. If there is a hyphen (-) in place then it indicates that the group doesn’t have that permission.
* 8nd-10th character: These characters indicated the read, write, and execute (r,w,x) permissions for other. If there is a hyphen (-) in place then it indicates that other doesn’t have that permission.

Following the first hyphen, rw-, is the user group. It shows that they have reading permissions, writing permissions, but no executing permissions. Afterwards the next ownership is group, r--, which shows that they only have reading permissions. Lastly, ---, correlates to other; and it shows that they have no reading, writing or executing permissions.

## 

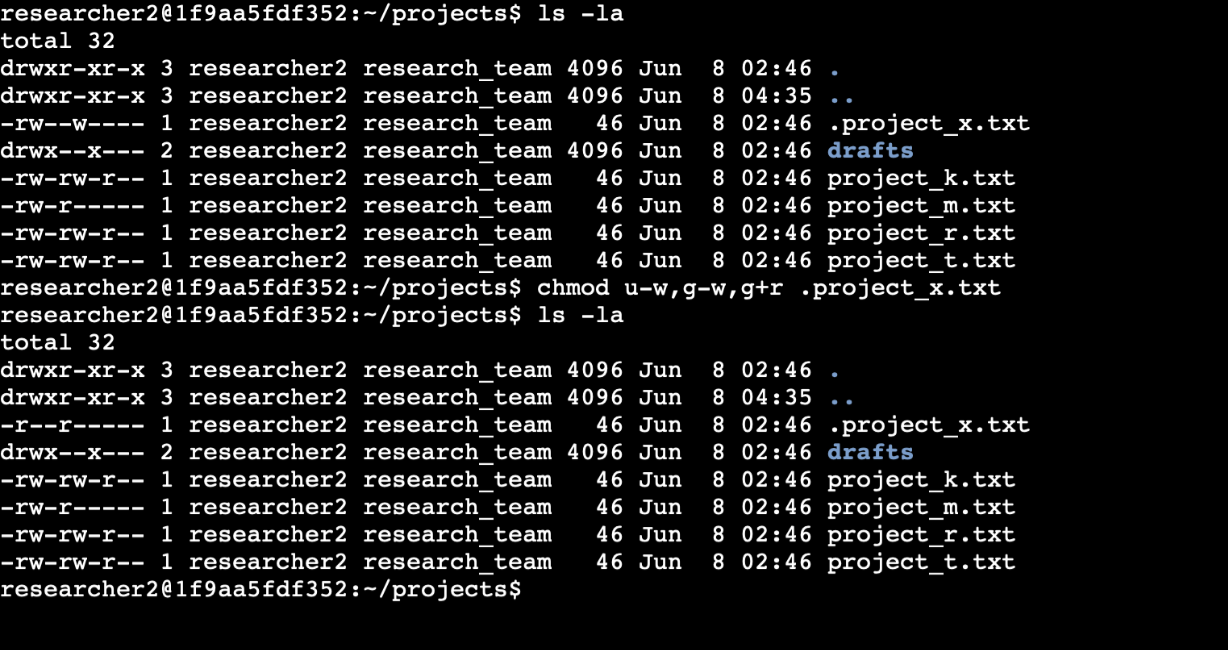
## Change file permissions

The organization does not allow other to have write access to any files. As you can see below project\_k.txt reads, -rw-rw-rw-, which means that the user, group, and other have reading and writing permissions. In order to change other permissions I used the chmod command, chmod o-w project\_k.txt, which removed other’s writing permissions.



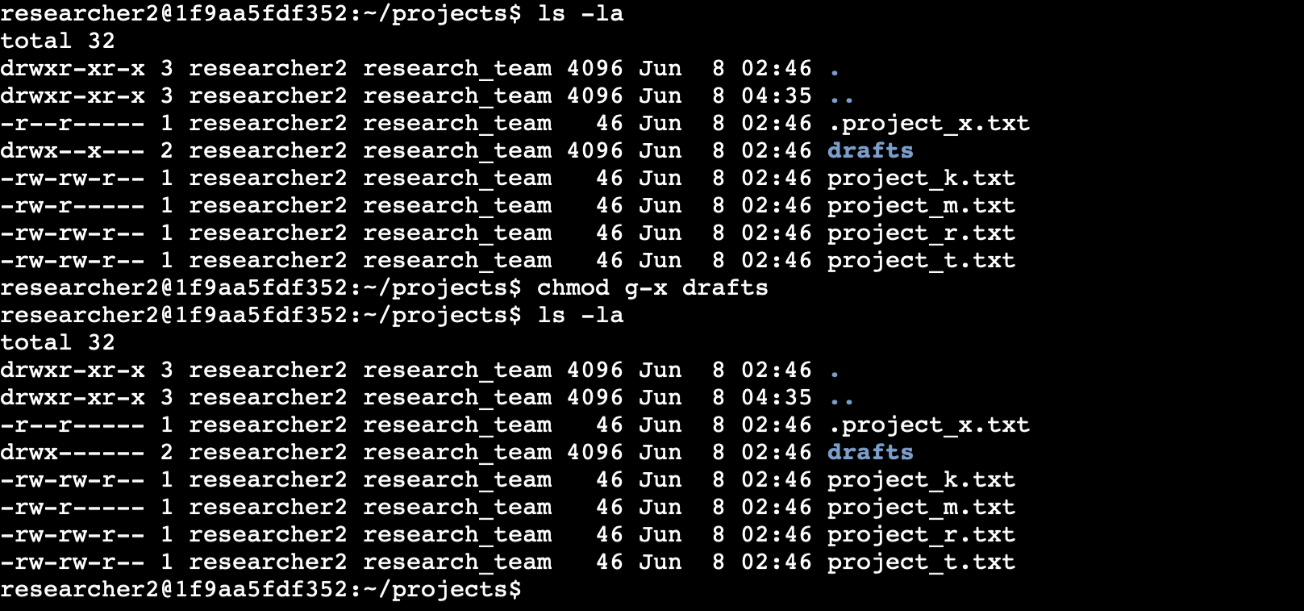
## Change file permissions on a hidden file

The hidden, or archived, file .project\_x.txt shouldn’t have any writing permissions for anyone. However, the user and group had permission to write. Furthermore, the user and the group should be able to read the hidden file. In order to change the permissions I typed, chmod u-w,g-w,g+r .project\_x.txt, which removed the user and group’s ability to write and added the ability for the group to read the hidden file.



## Change directory permissions

The organization only wants researcher2 user to have access to the drafts directory and its contents. However, group has execute permissions. I used the chmod command to remove the groups execute permissions, so that only the user would have access to drafts.



## Summary

I changed multiple permissions to match the level of authorization my organization wanted for files and directories in the projects directory. The first step in this was using ls -la to check the permissions for the directory. This informed my decisions in the following steps. I

then used the chmod command multiple times to change the permissions on files and

directories.