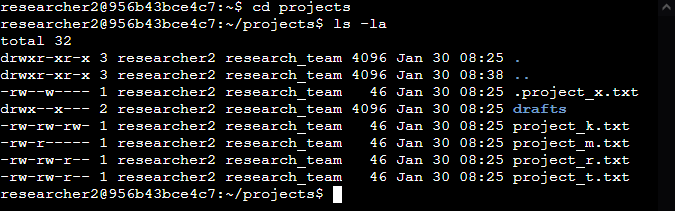
# File permissions in Linux

## Project description

Identifying and changing file permissions can strongly contribute to the confidentiality, integrity and availability of information stored on a computer. The Linux command ls -la can be used to check the file permissions in a folder, while the command chmod can be used change the file permissions on files and directories. A combination of these commands and their modifiers can be used to ensure that only authorized users have access to files even on a shared computer.

## Check file and directory details

First we enter the projects folder using the cd command and list all the files and their permissions using the ls -la command.

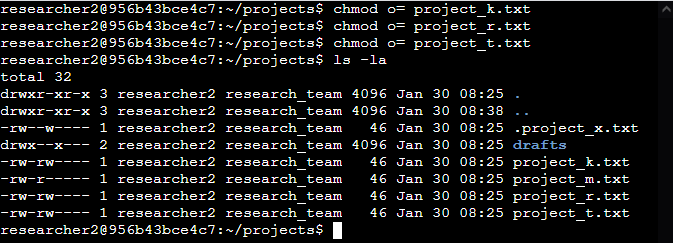


## Describe the permissions string

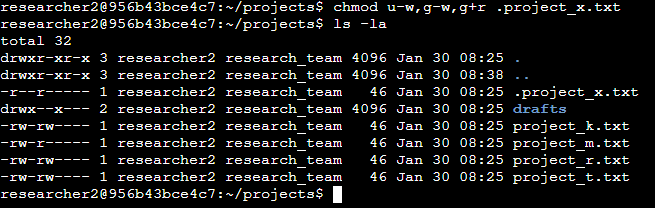
Using the file .project\_x.txt as our example, here is what the 10 character file permissions descriptors -rw--w---- mean:

1. -: The first character describes the file type. - mean its a normal file, d means its a directory
2. r: Characters 2 – 4 describe permission relation to the user. The second character describes the read permissions for the user, researcher2 in our case, r means user has read permissions, - would be they do not,
3. w: The third character describes the write permissions for the user. w means the user has write permissions, - would mean they do not.
4. -: The fourth character describes the execution permissions for the user. x would mean the user has execution permissions on the file and – would mean they do not.
5. -: Characters 5 – 7 describe the group permissions. The fifth character describes the read permissions for the group, in this case research\_team. - means the group has no read permission on this file, r would mean they do.
6. w: This character describes the write permission for the group. w means they have write permission – would mean they do not.
7. -: This character describes the execution permissions for the group, - means they do not have execution permission x would me they do.
8. -: Characters 8 – 10 describe permissions for other users. The eighth character describes the read permissions for other users. r would mean other have read permission, - would mean other users do not have read permission,
9. -: This character describes the write permissions for other users. - means other users do not have write permission, w would mean they do.
10. -: The tenth character describes execution permissions for other users. - means they do not have execution permission, x would mean they do.

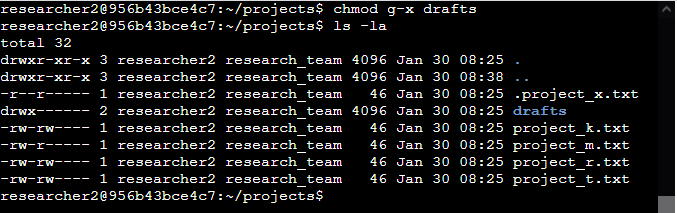
## Change file permissions

We changed the file permissions to project\_k.txt, project\_t.txt and project\_r.txt so that other users have no access permissions. The chmod o= sets the file permissions for other user for each file to none.

## Change file permissions on a hidden file

We changed the file permissions to hidden file .project\_x.txt using the chmod command to give read-only permissions to both group and user. Modifiers u-w, g-w remove write permissions for both the user and the group, this command can also be written as ug-w. Modifier g+r gives the group read permissions. Upon inspection using the ls -la command we can see that the file is now read-only for both user and group.

## Change directory permissions

The Linux command chmod g-x drafts removes execution permissions for the group from the drafts folder. ls -la shows that after this command is executed now only the user has access to the drafts folder.

## Summary

In this exercise we identified the permissions on files and directories in a folder, including hidden files using the ls -la Linux command. We then changed these permissions to ensure least privilege and other access modifications using the chmod Linux command. Overall our goal was to ensure the confidentiality, integrity and availability of the files stored on our computer.