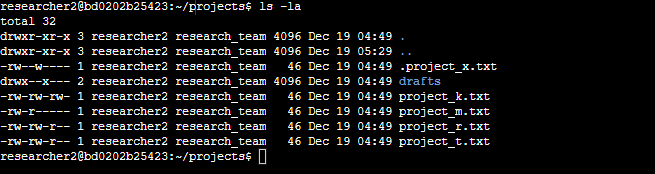
**File permissions in Linux**

Project description

In this scenario, the research team tasked me to modify the permissions for files and directories within the project directory. The operating system is Linux, indicating that the tasks require a command-line interface (Linux Bash shell) approach via Linux Terminal.

Check files and directory details

To begin with, I wrote the command ls to display what directories are available. As the result goes, the project is the only directory listed. Then, the command ls with the -la displays file contents as well as the hidden files within the directory of the project. The result shows there is one hidden file within the project directory. The hidden file naming conventions start with a period (.), followed by its name. In this case, ”.project\_x.txt ” is the hidden file. Other findings include four project files (ends with .txt) and one directory.



Describe the permissions string

The 10-character string determines the authorization of accessing the file and their specific permissions. The characters and what they represent are as follows:  
We’ll take the rst row from the picture above:

**drwxr-xr-x**

● **1st character**: This character is either a d or hyphen (-) and indicates the file type. Character d shows that it is a directory and drafts is the example. A hyphen (-) shows that it is a regular file.

●  **2nd-4th characters**: These characters indicate the read (r), write (w), and execute (x) permissions for the **user**. When one of these characters is a hyphen (-) instead, it indicates that this permission is not granted to the **user**.

●  **5th-7th characters:** These characters indicate the read (r), write (w), and execute (x) permissions for the **group**. When one of these characters is a hyphen (-) instead, it indicates that this permission is not granted for the **group**.

●  **8th-10th characters:** These characters indicate the read (r), write (w), and execute (x) permissions for **others**. It includes all other users on the system that are not **users** and the **group**. When one of these characters is a hyphen (-) instead, that indicates that this permission is not granted for **others**.

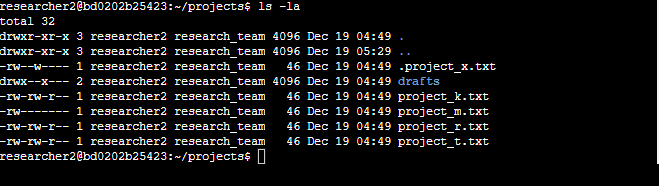
Change file permissions

The command chmod allows us to change the le permissions. To do this, there are some important notes:

* 1. The command chmod u+(r/w/x) project file name allows us to add the file permissions for the **users**.
  2. The command chmod u-(r/w/x) project file name allows us to remove the file permissions for the **users**.
  3. The command chmod g+(r/w/x) project file name allows us to add the file permissions for the **groups**.
  4. The command chmod g-(r/w/x) project file name allows us to remove the file permissions for the **group**.
  5. The command chmod o+(r/w/x) project file name allows us to add the le permissions for **others**.
  6. The command chmod o-(r/w/x) project file name allows us to remove the le permissions for **others**.

**Changes that I made:**

* I wrote the command chmod o-w project\_k.txt to remove write permissions from the file.
* I wrote the command chmod g-r project\_m.txt to remove read permissions from the file.

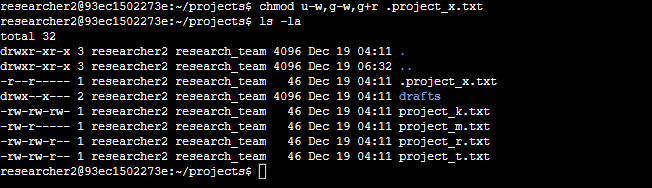


Change file permissions on a hidden file

The command **chmod** also allows us to change the le permissions for the hidden files. As for  
**‘’.project\_x.txt‘’**, I would like to remove the write permissions for the users and the group while maintaining read permissions for the group. The following code is able to make it happen

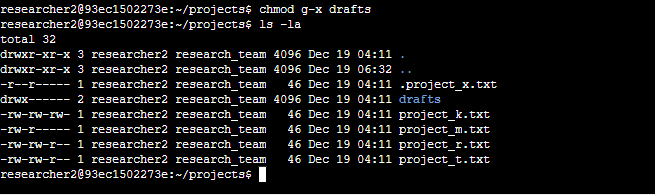
in a single line of code:

Chmod u-w,g-w,g+r .project\_x.txt



Change directory permissions

The command chmod g-x drafts will authorize only researcher2 to gain access to drafts directory.



Summary

This scenario demonstrates my capability to match the level of authorization my organization set for les and directories in the project directory. The command ls -la displays all the les in the directory while allows you to change permissions and directories.