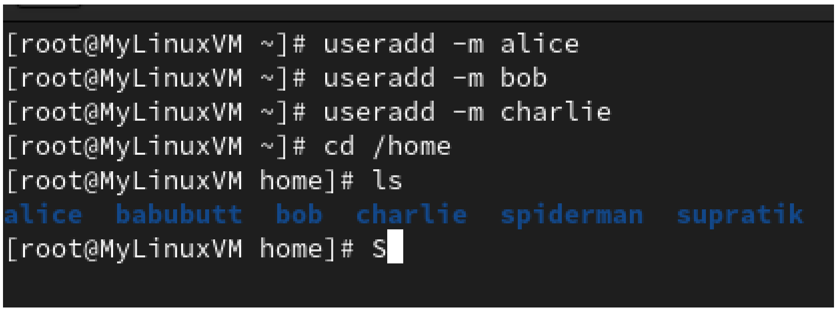
Assignment 3

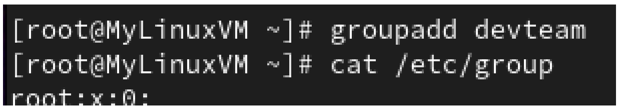
Part 1

Q1) Create the following users- alice, bob, charlie



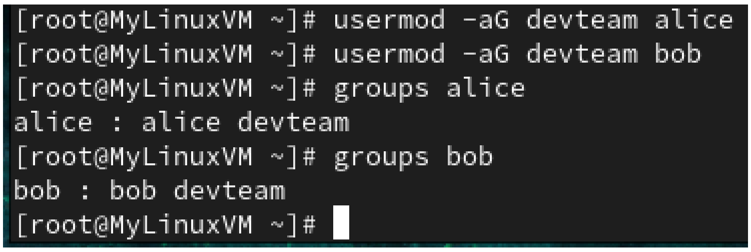
```
[root@MyLinuxVM ~]# useradd -m alice
[root@MyLinuxVM ~]# useradd -m bob
[root@MyLinuxVM ~]# useradd -m charlie
[root@MyLinuxVM ~]# cd /home
[root@MyLinuxVM home]# ls
alice  babubutt  bob  charlie  spiderman  supratik
[root@MyLinuxVM home]# S
```

Q2) Create a group called “devteam”



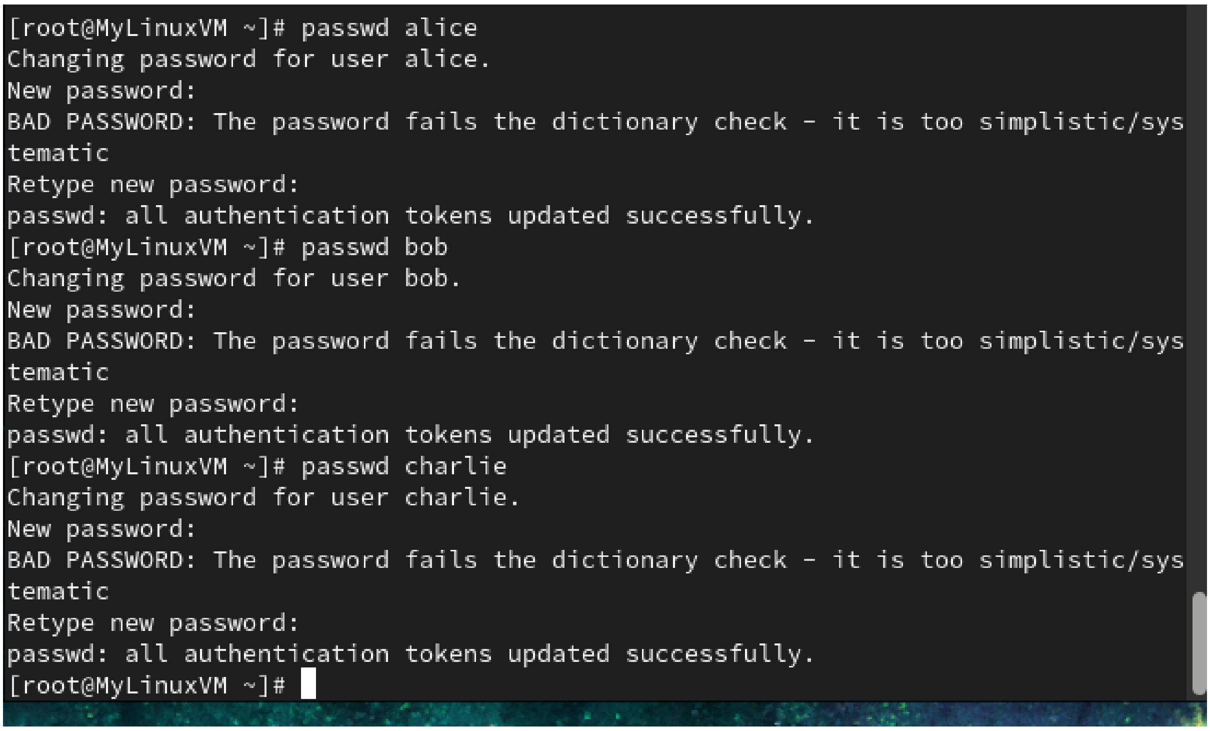
```
[root@MyLinuxVM ~]# groupadd devteam
[root@MyLinuxVM ~]# cat /etc/group
root:x:0:
```

Q3) Add users “alice” and “bob” to the devteam group



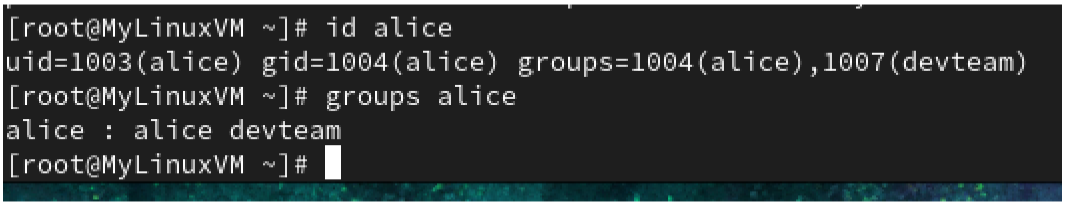
```
[root@MyLinuxVM ~]# usermod -aG devteam alice
[root@MyLinuxVM ~]# usermod -aG devteam bob
[root@MyLinuxVM ~]# groups alice
alice : alice devteam
[root@MyLinuxVM ~]# groups bob
bob : bob devteam
[root@MyLinuxVM ~]#
```

Q4) Set passwords for all users



```
[root@MyLinuxVM ~]# passwd alice
Changing password for user alice.
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/sys
tematic
Retype new password:
passwd: all authentication tokens updated successfully.
[root@MyLinuxVM ~]# passwd bob
Changing password for user bob.
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/sys
tematic
Retype new password:
passwd: all authentication tokens updated successfully.
[root@MyLinuxVM ~]# passwd charlie
Changing password for user charlie.
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/sys
tematic
Retype new password:
passwd: all authentication tokens updated successfully.
[root@MyLinuxVM ~]#
```

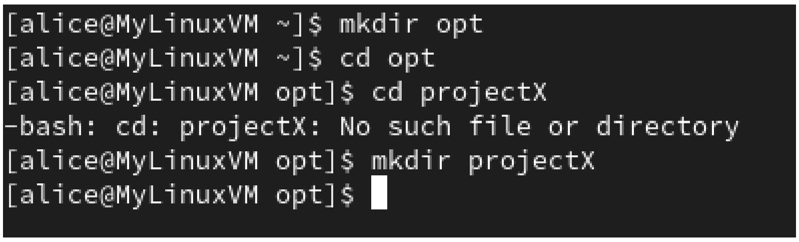
Q5) Verify user ID’s and group membership using id and group



```
[root@MyLinuxVM ~]# id alice
uid=1003(alice) gid=1004(alice) groups=1004(alice),1007(devteam)
[root@MyLinuxVM ~]# groups alice
alice : alice devteam
[root@MyLinuxVM ~]#
```

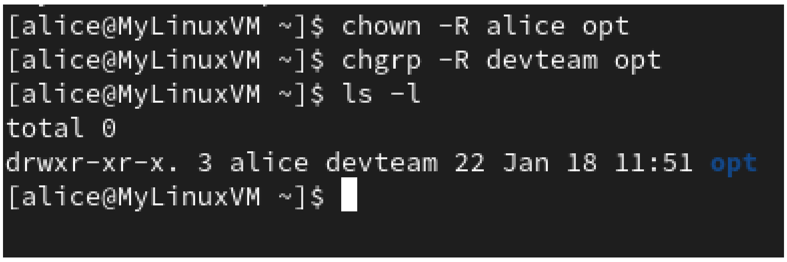
Part 2

Q1) Create a directory /opt/project



```
[alice@MyLinuxVM ~]$ mkdir opt
[alice@MyLinuxVM ~]$ cd opt
[alice@MyLinuxVM opt]$ cd projectX
-bash: cd: projectX: No such file or directory
[alice@MyLinuxVM opt]$ mkdir projectX
[alice@MyLinuxVM opt]$
```

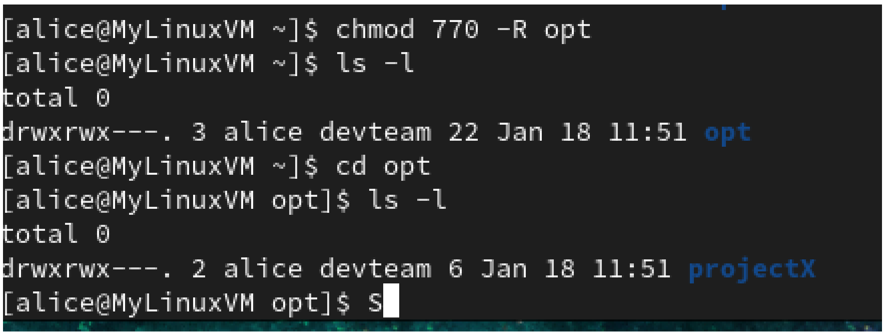
Q2) Change ownership – owner : alice, group: devteam



```
[alice@MyLinuxVM ~]$ chown -R alice opt
[alice@MyLinuxVM ~]$ chgrp -R devteam opt
[alice@MyLinuxVM ~]$ ls -l
total 0
drwxr-xr-x. 3 alice devteam 22 Jan 18 11:51 opt
[alice@MyLinuxVM ~]$
```

Q3) Apply permission such that: Owner and group -> full access and other -> no access

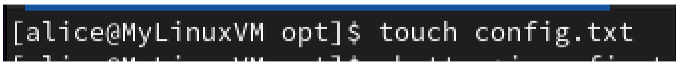
Q4) Verify permission using ls -l



```
[alice@MyLinuxVM ~]$ chmod 770 -R opt
[alice@MyLinuxVM ~]$ ls -l
total 0
drwxrwx---. 3 alice devteam 22 Jan 18 11:51 opt
[alice@MyLinuxVM ~]$ cd opt
[alice@MyLinuxVM opt]$ ls -l
total 0
drwxrwx---. 2 alice devteam 6 Jan 18 11:51 projectX
[alice@MyLinuxVM opt]$ $
```

Part 3

Q1) Create a file inside the directory: config.txt



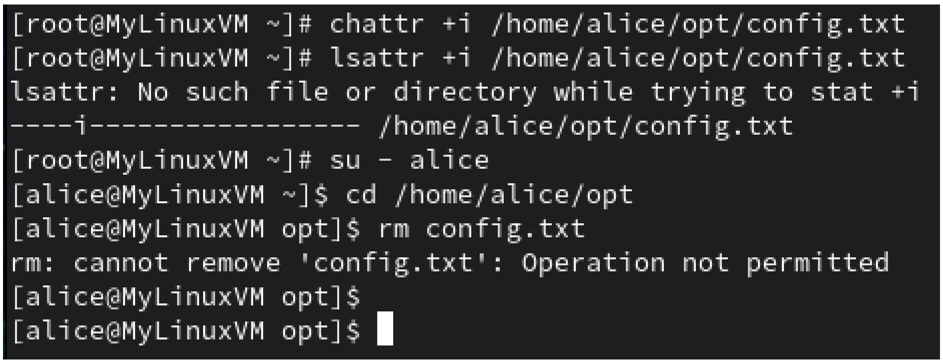
```
[alice@MyLinuxVM opt]$ touch config.txt
```

Q2) Make the file immutable so it cannot be deleted or modified accidentally

Q3) Verify attributes using lsattr

Q4) Attempt to delete or edit the file

(Answering above 3 questions together)



```
[root@MyLinuxVM ~]# chattr +i /home/alice/opt/config.txt
[root@MyLinuxVM ~]# lsattr +i /home/alice/opt/config.txt
lsattr: No such file or directory while trying to stat +i
----i----------------- /home/alice/opt/config.txt
[root@MyLinuxVM ~]# su - alice
[alice@MyLinuxVM ~]$ cd /home/alice/opt
[alice@MyLinuxVM opt]$ rm config.txt
rm: cannot remove 'config.txt': Operation not permitted
[alice@MyLinuxVM opt]$
[alice@MyLinuxVM opt]$
```

Conceptual Questions

Q1) Why use groups instead of giving permissions to individual users?

➔ Groups provide **efficient** management. By applying permissions to a group rather than individuals, members automatically inherit access. This ensures **consistency** and significantly reduces administrative overhead.

Q2) Why would an immutable file be useful in production?

- ➔ Immutable files prevent accidental or malicious changes to critical system files. They ensure **security** by protecting logs, configuration files from being modified or deleted, even by the root user.

Q3) Why is /etc/shadow readable only by root?

- ➔ The `/etc/shadow` file is restricted to root because it stores **encrypted password hashes**. Restricting access prevents regular users from performing **offline brute-force attacks** to discover other users' passwords.