Add and Manage Users with Linux Commands

Project description

This project focused on practicing **user management and file permission handling** in a Linux environment using Bash shell commands. The scenario simulated real-life tasks performed by a security analyst or system administrator, including adding new users, assigning them to groups, managing file ownership, and safely removing users and their associated groups from the system.

Project Objectives:

- ✅ Add a new user to the system.

- ✅ Assign the user to a **primary group.**

- ✅ Change file ownership to the new user.

- ✅ Add the user to a **secondary group.**

- ✅ Delete the user and clean up system groups.

- ✅ Verify each step using system queries.

What I Did :

1. Added a New User to the System:

```
sudo useradd researcher9
```

- ✅ **Command Explanation:**
  `useradd` creates a new user account named `researcher9`. The `sudo` prefix is required for administrative privileges.

2. Assigned a Primary Group:

```
sudo usermod -g research_team researcher9
```

- ✅ **Command Explanation:**
  `usermod` modifies an existing user.
  `-g` sets the **primary group** for the user.
  The user `researcher9` is now primarily associated with the `research_team`.

```
analyst@6baf3c2b897c:~$ sudo usermod -g research_team researcher9
analyst@6baf3c2b897c:~$ sudo chown project_r.txt_researcher9
```

3. Changed File Ownership:

```
sudo chown researcher9 /home/researcher2/projects/project_r.txt
```

✅ **Command Explanation:**
`chown` changes the owner of a file or directory.
This makes `researcher9` the owner of `project_r.txt`, giving them control over the file.

❌ **Note:**
I initially ran the wrong command:

```
sudo chown project_r.txt researcher9
```

This was incorrect because the **syntax was flipped** — the correct format is:

```
sudo chown new_owner file
```

```
analyst@6baf3c2b897c:~$ sudo chown project_r.txt researcher9
chown: invalid user: 'project_r.txt'
analyst@6baf3c2b897c:~$ sudo chown researcher9 /home/researcher2/projects
/project_r.txt
```

4. Added the User to a Secondary Group:

```
sudo usermod -aG sales_team researcher9
```

- ✅ **Command Explanation:**
  `-a` (append) ensures the user keeps existing group memberships.

-G adds the user to the `sales_team` as a **secondary group.**

Without -a, the user would have been removed from their other groups.

```
analyst@6baf3c2b897c:~$ sudo usermod -aG sales_team researcher9
analyst@6baf3c2b897c:~$ groups researcher9
researcher9 : research_team sales_team
```

---

5. Verified Group Membership:

`groups researcher9`

- ✅ **Command Explanation:**
  This lists all groups that the user belongs to.
  Output confirmed the user was in both `research_team` (primary) and `sales_team` (secondary).

```
analyst@6baf3c2b897c:~$ groups researcher9
researcher9 : research_team sales_team
```

6. Deleted the User:

`sudo userdel researcher9`

- ✅ **Command Explanation:**
  `userdel` removes the user account from the system.

⚙️ Linux returned:

`userdel: group researcher9 not removed because it is not the primary group of user researcher9.`

- ➜ This is expected. Linux creates a user-specific group by default, but won't delete it automatically if the user's **primary group** is something else.

```
analyst@6baf3c2b897c:~$ sudo userdel researcher9
userdel: group researcher9 not removed because it is not the primary grou
p of user researcher9.
```

●

7. Deleted the Leftover Group:

```
sudo groupdel researcher9
```

- ✅ **Command Explanation:**
  `groupdel` deletes the now-unused group named `researcher9`.

```
nalyst@6baf3c2b897c:~$ sudo groupdel researcher9
```

---

8. Verified User Deletion:

```
grep 'researcher9' /etc/passwd
```

- ✅ **Command Explanation:**
  Checked the system user file to confirm the user was fully removed.

- **Result:**
  No output → ✅ User successfully deleted.

```
analyst@6baf3c2b897c:~$ grep 'researcher9' /etc/passwd
analyst@6baf3c2b897c:~$ ▊
```

Summary

I efficiently added, modified, verified, and safely deleted a Linux user while managing group memberships and file permissions — all using essential system admin commands. Clean, controlled, and secure user management from start to finish.