Linux Group Management Hands-on Lab

Introduction:

This comprehensive hands-on lab focuses on Linux **group management**, a critical aspect of user access control and permission delegation in a multi-user environment. Participants will learn how to create groups with default and custom GIDs, assign users to specific groups, rename groups, and delete them when no longer needed. The lab also includes a practical exercise on setting up an "admin" group with sudo privileges — demonstrating the power and flexibility of group-based permission control. Participants will gain experience using commands like `groupadd`, `groupmod`, `groupdel`, `usermod`, and `visudo`, reinforcing essential system administration skills.

Creating a default group:

Let’s create the first group `designers` with default settings. This will help you understand how Linux creates groups with default GIDs.

```
student@cloudkida:~$ sudo groupadd designers
#The system will ask for your password to confirm permission.
[sudo] password for student:
```

Let’s verify the group was created:

```
student@cloudkida:~$ cat /etc/group | grep “designers”
designers:x:1005:
```

Creating a group with a specific GID:

Now, create a group called `marketing` and assign it a custom Group ID (GID), e.g., 2000. This is especially useful in environments where GIDs need to remain consistent across multiple systems.

```
student@cloudkida:~$ sudo groupadd -g 2000 marketing
```

# Verify the group:

```
student@cloudkida:~$ sudo /etc/group | grep marketing
marketing:x:2000:
```

This confirms that the `marketing` group has GID `2000`.

Creating a user, adding to 'admingroup' group, and giving sudo access to the group:

In this task, we will:

1. Create a group `admingroup`
2. Create a user `adminuser`
3. Add `adminuser` to the `admingroup` group
4. Provide the group sudo privileges

```
student@cloudkida:~$ sudo groupadd admingroup
student@cloudkida:~$ sudo useradd -m -G admingroup adminuser
student@cloudkida:~$ sudo passwd adminuser
```

Now grant sudo privileges to the `admingroup` group. Create the file with command below:

```
student@cloudkida:~$ sudo vim /etc/sudoers.d/admin_group
```

Add this line in the file:

```
%admingroup ALL=(ALL:ALL) ALL
```

Save and exit.

To confirm the user is in the group:

```
student@cloudkida:~$ groups adminuser
adminuser : adminuser admingroup
```

Now `adminuser` can use `sudo` if logged in.

Renaming an existing group:

Let’s first create the group `developer` with default settings.

```
student@cloudkida:~$ sudo groupadd developer
```

Let’s rename the existing group `developer` to `uiux`.

```
student@cloudkida:~$ sudo groupmod -n uiux developer
```

Check if the group name changed:

```
student@cloudkida:~$ getent group uiux
uiux:x:1005:
```

Deleting a group and observing changes:

Sometimes, groups become obsolete or redundant and need to be removed. In this exercise, we will create a group, assign it as the group owner of a file, then delete the group and observe the effect.

Let’s start by creating a group named `tempgroup`.

```
student@cloudkida:~$ sudo groupadd tempgroup
```

Now, create a file named example.txt.

```
student@cloudkida:~$ touch example.txt
```

Change the group ownership of this file to tempgroup:

```
student@cloudkida:~$ sudo chgrp tempgroup example.txt
```

Check the group ownership of the file.

```
student@cloudkida:~$ ls -l example.txt
-rw-r--r-- 1 student tempgroup 0 Apr 30 10:05 example.txt
```

Check the group ownership of the file.

```
student@cloudkida:~$ sudo groupdel tempgroup
```

After deletion, check the file again.

```
student@cloudkida:~$ ls -l example.txt
-rw-r--r-- 1 student 1010 0 Apr 30 10:05 example.txt
```

Here, there is GID of the deleted group tempgroup. Since the system can no longer resolve the group name, it shows the raw GID — confirming the group was successfully deleted.

Conclusion and Key Takeaways:

Command	Description
`groupadd`	Creates a new group
`groupadd -g`	Assigns a specific GID to a group
`groupmod -n`	Renames a group
`groupdel`	Deletes an existing group
`usermod -aG`	Adds a user to a supplementary group
`visudo`	Securely edits the sudoers file for group-based access

Group-based management is critical for secure, scalable permission handling on Linux systems. Mastering these commands ensures efficient system administration practices.