

# Day 3 – Phase 3: User, Group, and Permissions Management

**Boss's Request:** Secure the project and restrict access to authorized users only.

## Tasks:

- Create a new group `iot_team` and add your user to it.

```
sudo groupadd iot_team
sudo gpasswd -a $USER iot_team
```

```
mohamed@iot ~> sudo groupadd iot_team
mohamed@iot ~> tail -n 1 /etc/group
iot_team:x:1001:
mohamed@iot ~> sudo gpasswd -a $USER iot_team
Adding user mohamed to group iot_team
mohamed@iot ~> groups mohamed
mohamed : mohamed adm cdrom sudo dip plugdev lpadmin lxd sambashare iot_team
```

- Create a new developer user, add it to the group.

```
sudo adduser developer
sudo gpasswd -a developer iot_team
```

```
mohamed@iot ~> sudo adduser developer
Adding user 'developer' ...
Adding new group 'developer' (1002) ...
Adding new user 'developer' (1001) with group 'developer' ...
Creating home directory '/home/developer' ...
Copying files from '/etc/skel' ...
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: password updated successfully
Changing the user 'information' for developer
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
mohamed@iot ~> tail -n 1 /etc/passwd
developer:x:1001:1002::,/home/developer:/bin/bash
mohamed@iot ~> sudo gpasswd -a developer iot_team
Adding user developer to group iot_team
mohamed@iot ~> groups developer
developer : developer iot_team
```

- Change ownership of `iot_logger` to the developer + group.

```
sudo chown -R developer:iot_team ~/iot_logger
```

```
mohamed@iot ~> sudo chown -R developer:iot_team ~/iot_logger
mohamed@iot ~> ls -l
total 44
drwxr-xr-x 2 mohamed mohamed 4096 Aug 24 23:00 Desktop
drwxr-xr-x 2 mohamed mohamed 4096 Aug 24 23:00 Documents
drwxr-xr-x 2 mohamed mohamed 4096 Aug 24 23:00 Downloads
drwxrwxr-x 5 mohamed mohamed 4096 Aug 30 11:13 iot
drwxrwxr-x 5 developer iot_team 4096 Aug 31 21:50 iot_logger
drwxr-xr-x 2 mohamed mohamed 4096 Aug 24 23:00 Music
drwxr-xr-x 2 mohamed mohamed 4096 Aug 24 23:00 Pictures
drwxr-xr-x 2 mohamed mohamed 4096 Aug 24 23:00 Public
drwx----- 5 mohamed mohamed 4096 Aug 30 10:52 snap
drwxr-xr-x 2 mohamed mohamed 4096 Aug 24 23:00 Templates
drwxr-xr-x 2 mohamed mohamed 4096 Aug 24 23:00 Videos
mohamed@iot ~> ls -l iot_logger/
total 12
drwxrwxr-x 2 developer iot_team 4096 Aug 31 21:26 data
drwxrwxr-x 2 developer iot_team 4096 Aug 31 21:53 logs
drwxrwxr-x 2 developer iot_team 4096 Aug 31 21:23 scripts
```

- Set permissions: group can read/write logs, others blocked.

```
sudo chmod -R g=rw,o-rwx ~/iot_logger/logs
```

# or

```
sudo chmod -R 760 ~/iot_logger/logs # that may change owner permissions too
```

```
mohamed@iot ~> ls -l iot_logger/
total 12
drwxrwxr-x 2 developer iot_team 4096 Aug 31 21:26 data
drwxrwxr-x 2 developer iot_team 4096 Aug 31 21:53 logs
drwxrwxr-x 2 developer iot_team 4096 Aug 31 21:23 scripts
mohamed@iot ~> sudo chmod -R g=rw,o-rwx ~/iot_logger/logs
mohamed@iot ~> ls -l iot_logger/
total 12
drwxrwxr-x 2 developer iot_team 4096 Aug 31 21:26 data
drwxrw---- 2 developer iot_team 4096 Aug 31 21:53 logs
drwxrwxr-x 2 developer iot_team 4096 Aug 31 21:23 scripts
```

- Test access as new user, then remove test user.
  - **developer** is owner of `iot_logger` , so he can access `iot_logger` not `/home/mohamed/iot_logger`

```
cd iot_logger/  
# change to developer user at same path  
su developer
```

```
# to test access  
touch logs/test.txt
```

```
mohamed@iot ~> cd iot_logger/  
mohamed@iot ~/iot_logger> su developer  
Password:  
developer@iot:/home/mohamed/iot_logger$ touch logs/test.txt  
developer@iot:/home/mohamed/iot_logger$ tree .  
.  
├── data  
│   └── services  
├── logs  
│   ├── hard_temperature.log  
│   ├── symbolic_temperature.log -> temperature.log  
│   ├── temperature.log  
│   └── test.txt  
└── scripts  
    └── sensor_script.py  
  
3 directories, 6 files
```

- remove test user

```
sudo userdel -r developer
```

## Open-Ended Questions:

- How do Linux file permissions ( `r`, `w`, `x` ) work for files vs directories? Give an example using `ls -l`.

Permission	Files	Directories
<code>r</code>	read file contents	list directory contents
<code>w</code>	modify file contents	create/delete files inside
<code>x</code>	execute file	access (cd into) the directory

- Example:

```
drwxrwxr-x 5 mohamed mohamed 4096 Aug 31 21:50 iot_logger
```

- Owner & group can do everything, others can read+enter
  - Explain octal notation for permissions and what the `umask` command does. Give one calculation example.
    - Octal notation →  $r=4, w=2, x=1$
    - Example:
      - `chmod 775 iot_logger`
- ```
drwxrwxr-x 5 mohamed mohamed 4096 Aug 31 21:50 iot_logger
```
- Owner = `111` = 7, Group = `111` = 7, Other = `101` = 5
  - `umask` sets default permissions when new files are created.
    - Example: if `umask 022`, If default file permissions = `666`, actual = `644`.
  - What is the difference between the root user and a normal user? Why is root considered dangerous?
    - **Root user** = superuser, full control (can change system files, delete anything).
    - **Normal user** = limited, safer.
    - Root is dangerous because mistakes (like `rm -rf /`) can break the system.