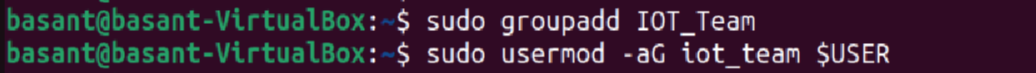
**Name:** Basant Tarik Salah   
**Instructor:** Eng. Mohamed Abo-Khalil

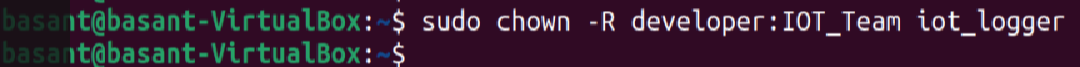
SIC7\_Task.Phase3

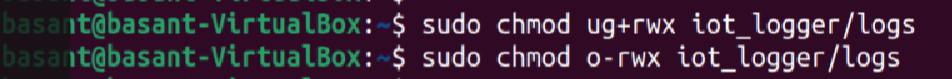
Part1: Tasks

1. Create a new group iot\_team and add your user to it
2. Create a new developer user, add it to the group.



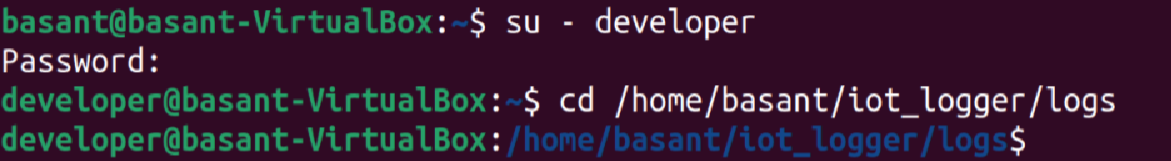
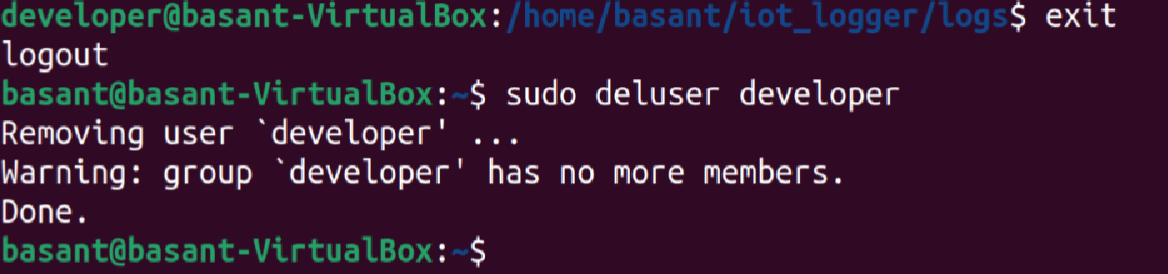
1. Change ownership of iot\_logger to the developer + group.



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

Or



1. Test access as new user, then remove test user.

Part 2: Open Ended Questions

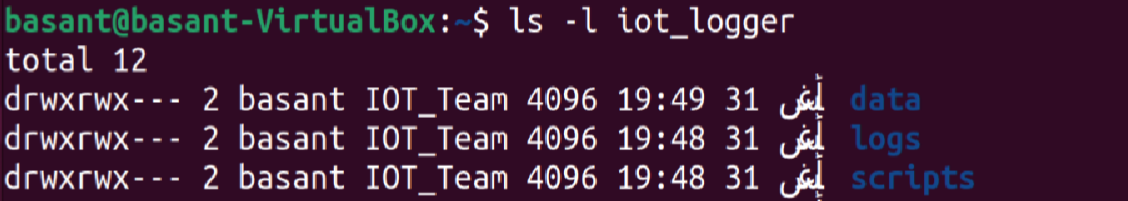
1. **How do Linux file permissions (r, w, x) work for files vs directories? Give an example using ls -l.**

Ans.

Linux uses 3 letters to control what users can do:

* **r = read**
  + For a **file**: enables opening file and seeing its content.
  + For a **directory:** enables listing its inside files.
* **w = write**
  + For a **file:** enables editing or deleting it.
  + For a **directory:** enables creating new files or deleting the existing files.
* **x = execute**
  + For a **file:** enables running it
  + For a **directory:** enables entering it

For Example:



**rwx:** the owner can do can do everything (read, write, execute).  
**rwx:** the group members can also do everything.  
**---:** others have no permission at all. They can’t even look inside

1. **Explain octal notation for permissions and what the umask command does. Give one calculation example.**

Ans.

Permissions can be written in octal notation as **numbers** instead of letters.  
Each permission has a number:

* **r “read” = 4**
* **w “write” = 2**
* **x “execute” = 1**

They are added together:

* **rwx**  **= 7** (full access)
* **rw-**  **= 6** (read + write)
* **r-x**  **= 5** (read + execute)
* **---**  **=** **0** (no permissions)

For Example **chmod 770** means:

* Owner = 7 (rwx)
* Group = 7 (rwx)
* Others = 0 (no access)

**umask** sets the default permissions when new files are created. It subtracts its value from the default file value.

For Example:

Default file mode = 666 (rw-rw-rw-)

If umask = 022: new file = 644 (rw-r--r--)

* Owner can read/write.
* Group and others can only read.

1. **What is the difference between the root user and a normal user? Why root is considered dangerous?**

Ans.

* **Normal users**

They have a limited power. As they can only change files in their home folder and can’t affect system files directly.

* **Root users (super users)**

They have unlimited power. As they can do anything without restrictions. They can install or remove software, delete any file, manage users and change system settings.

The command: **sudo** gives temporary root power.

It is dangerous because mistakes can break the whole system.