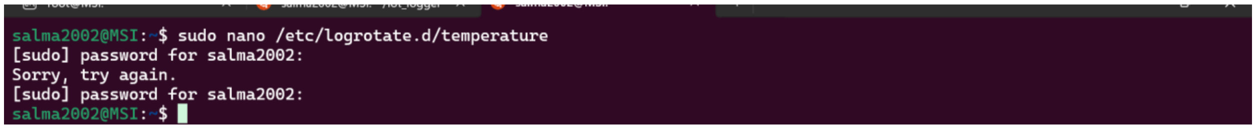
Day 6 – Phase 6: Log Rotation, Scheduling, Archiving

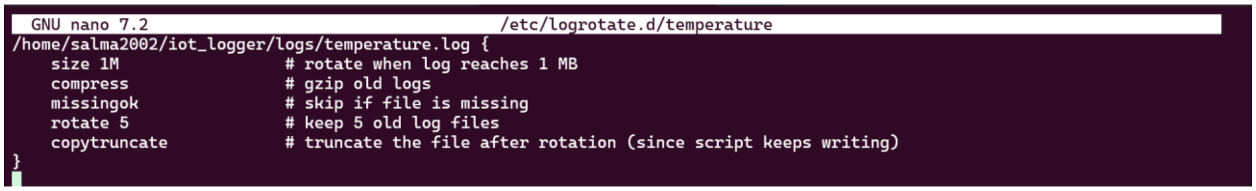
Boss's Request: Prepare the system for production use.

Tasks:

• Configure log rotation for temperature.log (rotate at 1 MB, compress).

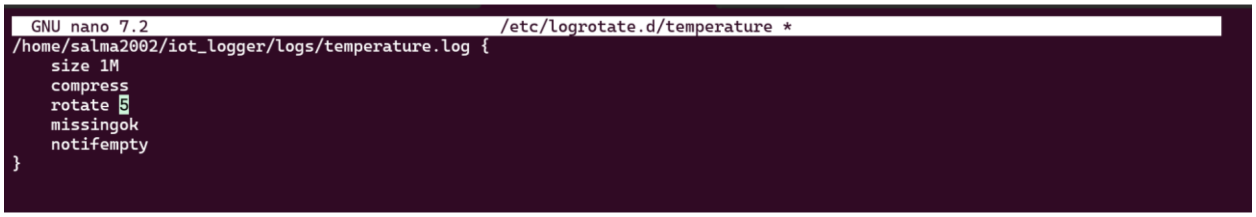


```
salma2002@MSI:~$ sudo nano /etc/logrotate.d/temperature
[sudo] password for salma2002:
Sorry, try again.
[sudo] password for salma2002:
salma2002@MSI:~$
```

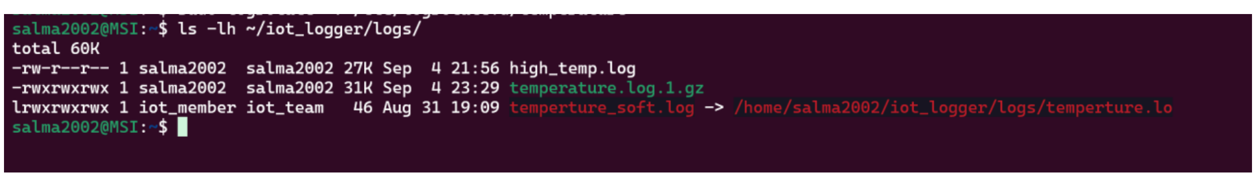


```
  GNU nano 7.2                                  /etc/logrotate.d/temperature
/home/salma2002/iot_logger/logs/temperature.log {
    size 1M                 # rotate when log reaches 1 MB
    compress                # gzip old logs
    missingok               # skip if file is missing
    rotate 5                # keep 5 old log files
    copytruncate            # truncate the file after rotation (since script keeps writing)
}
```

• Test by forcing a rotation.

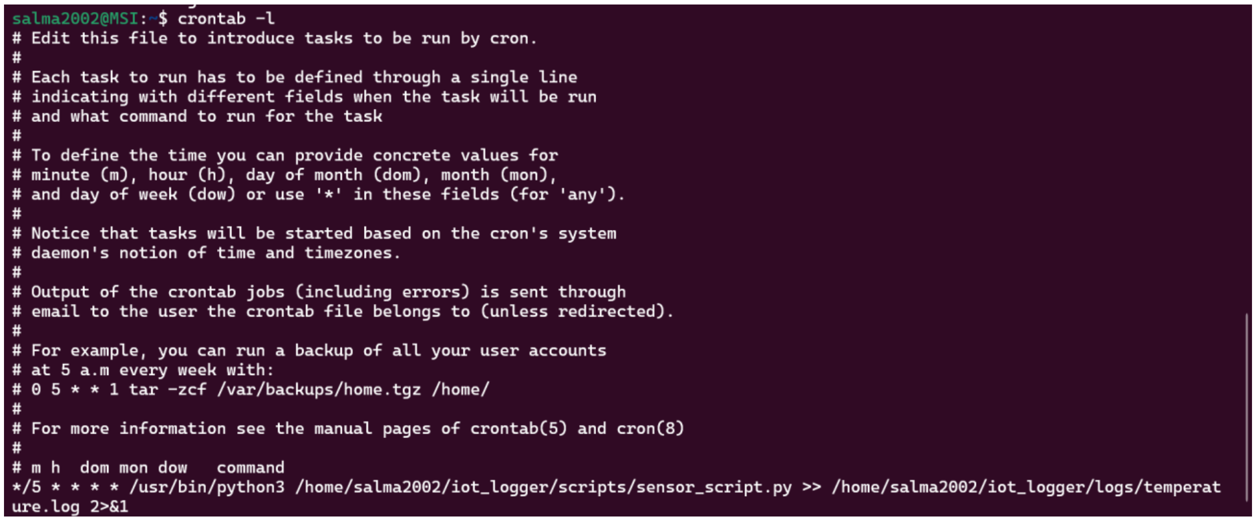


```
  GNU nano 7.2                                  /etc/logrotate.d/temperature *
/home/salma2002/iot_logger/logs/temperature.log {
    size 1M
    compress
    rotate 5
    missingok
    notifempty
}
```



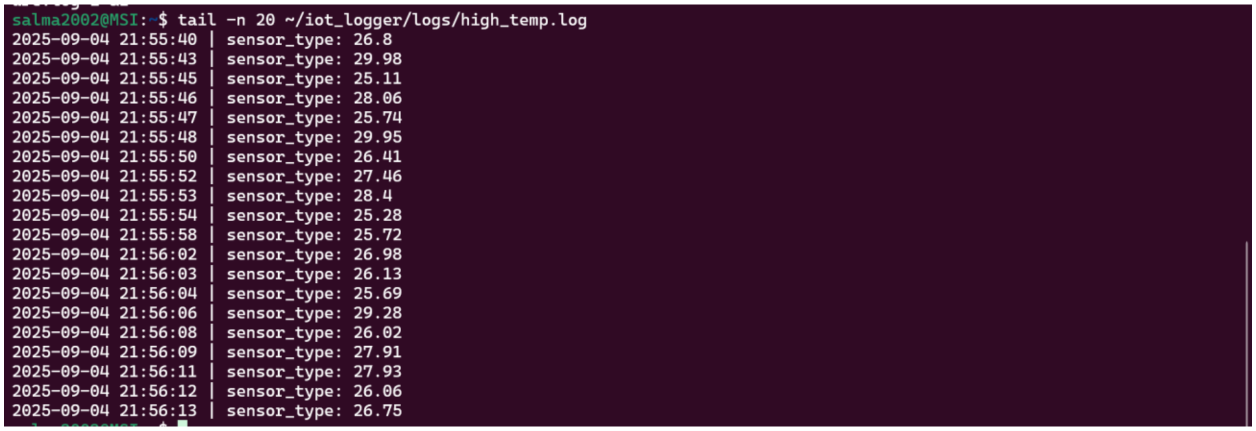
```
salma2002@MSI:~$ ls -lh ~/iot_logger/logs/
total 60K
-rw-r--r-- 1 salma2002  salma2002 27K Sep  4 21:56 high_temp.log
-rwxrwxrwx 1 salma2002  salma2002 31K Sep  4 23:29 temperature.log.1.gz
lrwxrwxrwx 1 iot_member iot_team   46 Aug 31 19:09 temperture_soft.log -> /home/salma2002/iot_logger/logs/temperture.lo
salma2002@MSI:~$
```

• Schedule the Python script to run every 5 minutes with cron.



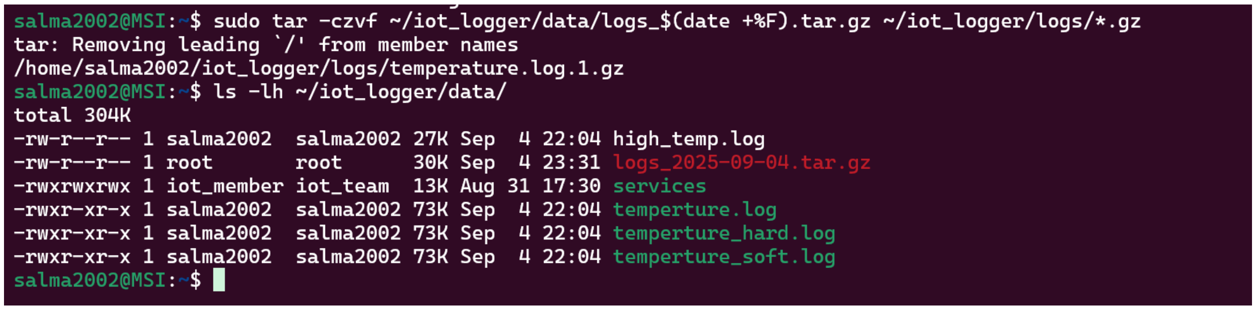
```
salma2002@MSI:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/5 * * * * /usr/bin/python3 /home/salma2002/iot_logger/scripts/sensor_script.py >> /home/salma2002/iot_logger/logs/temperat
ure.log 2>&1
```

• **Verify log growth over time.**



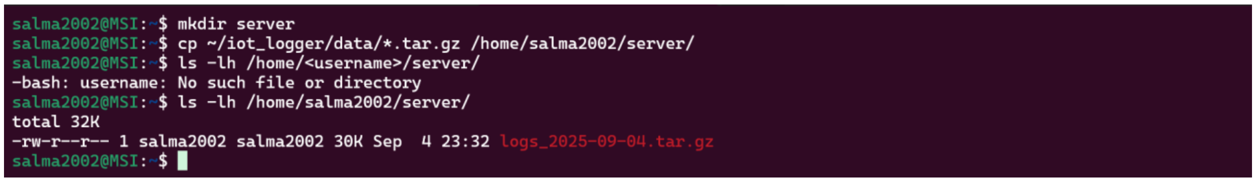
```
salma2002@MSI:~$ tail -n 20 ~/iot_logger/logs/high_temp.log
2025-09-04 21:55:40 | sensor_type: 26.8
2025-09-04 21:55:43 | sensor_type: 29.98
2025-09-04 21:55:45 | sensor_type: 25.11
2025-09-04 21:55:46 | sensor_type: 28.06
2025-09-04 21:55:47 | sensor_type: 25.74
2025-09-04 21:55:48 | sensor_type: 29.95
2025-09-04 21:55:50 | sensor_type: 26.41
2025-09-04 21:55:52 | sensor_type: 27.46
2025-09-04 21:55:53 | sensor_type: 28.4
2025-09-04 21:55:54 | sensor_type: 25.28
2025-09-04 21:55:58 | sensor_type: 25.72
2025-09-04 21:56:02 | sensor_type: 26.98
2025-09-04 21:56:03 | sensor_type: 26.13
2025-09-04 21:56:04 | sensor_type: 25.69
2025-09-04 21:56:06 | sensor_type: 29.28
2025-09-04 21:56:08 | sensor_type: 26.02
2025-09-04 21:56:09 | sensor_type: 27.91
2025-09-04 21:56:11 | sensor_type: 27.93
2025-09-04 21:56:12 | sensor_type: 26.06
2025-09-04 21:56:13 | sensor_type: 26.75
```

• **Compress old logs into .tar.gz in data/.**

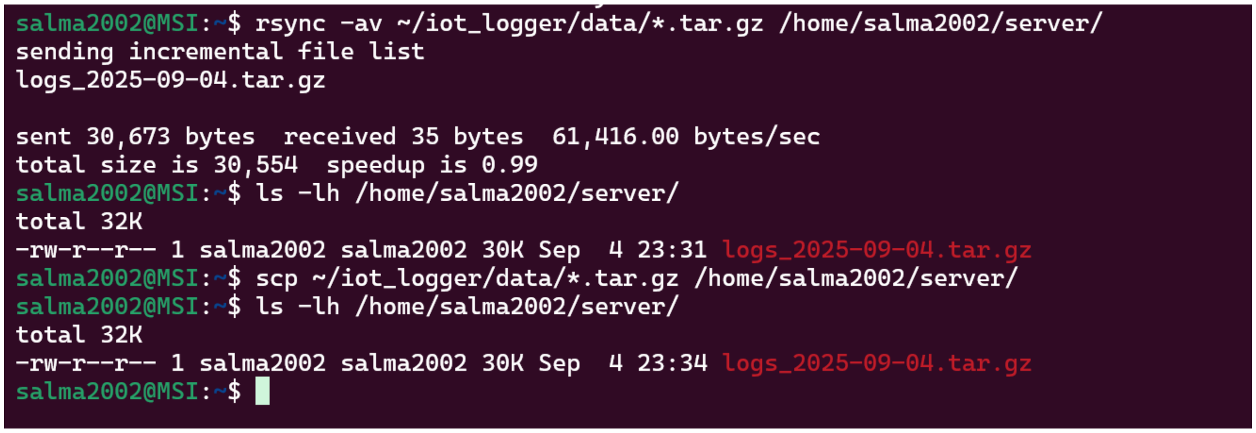


```
salma2002@MSI:~$ sudo tar -czvf ~/iot_logger/data/logs_$(date +%F).tar.gz ~/iot_logger/logs/*.gz
tar: Removing leading `/' from member names
/home/salma2002/iot_logger/logs/temperature.log.1.gz
salma2002@MSI:~$ ls -lh ~/iot_logger/data/
total 304K
-rw-r--r-- 1 salma2002  salma2002 27K Sep  4 22:04 high_temp.log
-rw-r--r-- 1 root       root      30K Sep  4 23:31 logs_2025-09-04.tar.gz
-rwxrwxrwx 1 iot_member iot_team  13K Aug 31 17:30 services
-rwxr-xr-x 1 salma2002  salma2002 73K Sep  4 22:04 temperture.log
-rwxr-xr-x 1 salma2002  salma2002 73K Sep  4 22:04 temperture_hard.log
-rwxr-xr-x 1 salma2002  salma2002 73K Sep  4 22:04 temperture_soft.log
salma2002@MSI:~$
```

• **Simulate sending archives to /home//server/ using cp, scp, or rsync. (hint: use can use scp and copy to destination directory in another path on the same machine just for simulation).**



```
salma2002@MSI:~$ mkdir server
salma2002@MSI:~$ cp ~/iot_logger/data/*.tar.gz /home/salma2002/server/
salma2002@MSI:~$ ls -lh /home/<username>/server/
-bash: username: No such file or directory
salma2002@MSI:~$ ls -lh /home/salma2002/server/
total 32K
-rw-r--r-- 1 salma2002 salma2002 30K Sep  4 23:32 logs_2025-09-04.tar.gz
salma2002@MSI:~$
```



```
salma2002@MSI:~$ rsync -av ~/iot_logger/data/*.tar.gz /home/salma2002/server/
sending incremental file list
logs_2025-09-04.tar.gz

sent 30,673 bytes  received 35 bytes  61,416.00 bytes/sec
total size is 30,554  speedup is 0.99
salma2002@MSI:~$ ls -lh /home/salma2002/server/
total 32K
-rw-r--r-- 1 salma2002 salma2002 30K Sep  4 23:31 logs_2025-09-04.tar.gz
salma2002@MSI:~$ scp ~/iot_logger/data/*.tar.gz /home/salma2002/server/
salma2002@MSI:~$ ls -lh /home/salma2002/server/
total 32K
-rw-r--r-- 1 salma2002 salma2002 30K Sep  4 23:34 logs_2025-09-04.tar.gz
salma2002@MSI:~$
```

Open-Ended Questions:

• How does cron scheduling work? Show a crontab entry to run a script every 5 minutes.

Cron is a time-based job scheduler in Unix-like operating systems. It works by using a daemon called crond that runs in the background and constantly checks for scheduled tasks. The schedule is defined in a file called a crontab (cron table), which contains commands to be executed at a specific time.

The crontab entry has five fields for specifying the time, followed by the command to be executed. The fields are:

- Minute (0-59)
- Hour (0-23)
- Day of the month (1-31)
- Month (1-12)
- Day of the week (0-6, where 0 and 7 are Sunday)

An asterisk * acts as a wildcard, matching all possible values for that field.

A crontab entry to run a script every 5 minutes would look like this:

*/5 * * * * /path/to/your/script.sh

In this example, */5 means "every 5th minute," and the remaining *s mean "every hour, every day of the month, every month, and every day of the week."

• **Why do we need log rotation? Show an example logrotate config for temperature.log.**

We need log rotation to manage disk space and improve system performance. Log files, especially on busy systems, can grow indefinitely, consuming all available disk space. This can lead to system instability and even crashes. Log rotation automates the process of archiving, compressing, and deleting old log files, ensuring that logs don't take up too much space.

Another important reason is to make log analysis easier. Instead of sifting through one gigantic log file, a series of smaller, date-stamped log files are easier to manage and analyze. This also prevents log files from becoming corrupted due to their large size.

• **Explain the difference between a Virtual Machine and a Container. Must containers use the same OS as the host? Why or why not?**

- **VM**:
  - Runs a full **guest OS** on top of a hypervisor
  - Heavy (GBs of memory, minutes to start)
  - Stronger isolation (separate kernel)
- **Container**:
  - Shares the **host OS kernel**
  - Lightweight (MBs, starts in seconds)
  - Uses namespaces + cgroups for isolation

Must containers use the same OS as the host?

- **Yes, same kernel.**
  - Example: Docker on Linux can only run Linux containers.
- But you can run different **distributions** (e.g., Ubuntu container on Fedora host).
- Windows/macOS use **VM layers** (like WSL2 or HyperKit) to run Linux containers.

• Reflection: Which actions in this project combined multiple Linux concepts (e.g., redirection + process monitoring)? How does this apply to real IoT systems?

- **Redirection (>>, 2>&1)** : saving sensor data to log files
- **Process monitoring (cron)** : scheduling periodic execution
- **Compression (gzip, tar)** → archiving logs
- **Permissions (chmod, groups)** : controlling file access

How this applies to real IoT systems:

- IoT devices continuously generate data : must **log & rotate** to prevent storage overflow.
- Logs are often **compressed & archived** for analysis in the cloud.
- Automation via **cron/systemd timers** ensures reliability without manual work.
- Using **lightweight containers** instead of VMs saves resources, critical for edge/IoT devices.