

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

- Cron is a Linux daemon used to schedule tasks at specific times or intervals.
- Users define jobs in a crontab file, which has 5 fields for time followed by the command:

* * * * * command

```
| | | | |
| | | | └─ Day of week (0-6, Sunday=0)
| | | └─── Month (1-12)
| | └───── Day of month (1-31)
| └──────── Hour (0-23)
└────────── Minute (0-59)
```

- Example: Run a script every 5 minutes

```
* /5 * * * * /home/iot/scripts/collect_temp.sh
```

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

- Problem: Logs grow indefinitely → take up disk space.
- Solution: Log rotation automatically archives, compresses, or deletes old logs to keep system healthy.
- Example logrotate config for temperature.log:

```
"/etc/logrotate.d/temperature
/home/iot/logs/temperature.log {
    daily      # rotate logs daily
    rotate 7   # keep 7 days of logs
    compress   # compress old logs with gzip
    missingok  # ignore if file is missing
    notifempty # don't rotate empty logs
    create 640 iotuser iotgroup # set permissions for new log file
}
```

- This ensures temperature logs don't overwhelm storage in an IoT device.

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

- Virtual Machine (VM): Runs a full operating system (with its own kernel) on virtualized hardware, Heavyweight, more resource usage. Example: Running Ubuntu VM on Windows via VirtualBox.
- Container: Shares the host OS kernel, but isolates applications with namespaces and cgroups, Lightweight, fast startup, efficient resource use. Example: Docker container running Python IoT app on Linux.
- Yes, containers must use the same kernel type as the host.
 - Example: Linux host → containers must use Linux kernel.
- But userspace can differ:
 - You can run Ubuntu container on a Debian host (both use Linux kernel).
 - You cannot run a Windows container directly on Linux without virtualization.

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

- Throughout this project, many tasks combined Linux concepts:
 Redirection + File Management: We redirected simulated sensor data (>> temperature.log) into files, then managed those files with permissions, ownership, and links.
 Processes + Job Control: We ran scripts in the background (&), monitored them with ps and jobs, switched between foreground/background, and killed processes when needed.
 Scheduling + Logging: Using cron to automate data collection combined process management with file redirection.
 Archiving + Security: Compressing logs and simulating transfers used file management, compression tools, and networking.
- Real IoT systems do exactly this: They collect and store sensor data in logs, manage processes so devices run continuously, schedule tasks for efficiency (cron, timers), archive and send data to servers for analysis. In production, combining these Linux basics ensures IoT devices are reliable, automated, and secure.