```
 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
1 * * * * /usr/bin/php /var/www/html/vesync/whatsapp-latest/artisan schedule:run >> /dev/null 2>&1
# 1 * * * * /usr/bin/curl -s "https://www.vesync.co.in/run-cron-schedule"
*/5 * * * * curl -s "https://www.vesync.co.in/run-cron-schedule" > /dev/null 2>&1
*/5 * * * * curl -s "https://www.vesync.co.in/requeue-sent-failed-messages" > /dev/null 2>&1
ubuntu@ip-172-31-15-5:~/system-monitor$ sudo crontab -e
```

i have saw multiple cron added but i am modifye only one please suggest

You're seeing **multiple cron jobs** because they belong to **different crontab scopes**, not because your edit added all of them. Let me explain clearly 👇

🔍 What's happening

You ran **two different commands**:

1️⃣ `crontab -l`

This shows the **current user's crontab**

👉 In your case: **ubuntu user**

Output:

bash Copy code

```bash
1 * * * * /usr/bin/php /var/www/html/vesync/whatsapp-latest/artisan schedule:run
*/5 * * * * curl -s "https://www.vesync.co.in/run-cron-schedule" > /dev/null 2>&1
*/5 * * * * curl -s "https://www.vesync.co.in/requeue-sent-failed-messages" > /de
```

✅ These **already existed**

❌ You did NOT add all of them now

2️⃣ `sudo crontab -e`

This opens the **root user's crontab**

👉 **Different cron file**

👉 Runs as **root**

👉 Completely separate from `ubuntu` user cron

So now you have:

Command	Cron owner
`crontab -l`	ubuntu
`crontab -e`	ubuntu
`sudo crontab -l`	root
`sudo crontab -e`	root

✅ If you modified only ONE cron job

That is **normal and correct** 👍

Cron does **not overwrite other entries**

Each line is independent.

🔎 How to confirm exactly what you changed

Check ubuntu user cron

```bash
crontab -l
```

Check root cron

```bash
sudo crontab -l
```

👉 You'll clearly see where your change exists

🧠 Best practice (IMPORTANT)

❌ Avoid this unless required

```bash
sudo crontab -e
```

✅ Preferred

```bash
crontab -e
```

Why?

- Laravel / curl jobs usually **do not need root**
- Safer
- Easier to debug
- Avoid permission surprises

🧹 If you want to remove or edit ONLY ONE cron

1. Open correct crontab:

   bash Copy code

   ```bash
   crontab -e
   ```

2. Modify **only that specific line**
3. Save & exit

Other cron jobs remain untouched.

🧪 Optional: See which cron is executing

Add temporary logging:

bash Copy code

```bash
*/5 * * * * echo "cron ran at $(date)" >> /tmp/cron-test.log
```

✅ Summary

- ✔️ Multiple cron entries ≠ mistake
- ✔️ You edited only one line (correct)
- ✔️ `crontab -l ≠ sudo crontab -l`
- ✔️ Cron jobs are **user-specific**

If you want, I can:

- 🔍 Identify **which cron is critical**
- 🧹 Suggest **cleanup / consolidation**
- 📊 Help convert this into **Laravel-only scheduler**
- 🛑 Disable duplicate executions

Just tell me 👍