



DevOps Cron Job Template with flock

Automated Job Scheduling with Overlap Protection and Logging

by Ekansh Gupta

Ekansh Gupta - DevOps Cron Job Template

Overview

This PDF provides a reusable, production-grade Bash script using `flock` to prevent overlapping cron job executions.

It is suitable for DevOps automation such as backups, cleanup, syncing, reporting, etc.

Consolidated Script Template

```
#!/bin/bash

# === Configurable Variables ===
LOCKFILE="/tmp/myjob.lock"
LOGFILE="/var/log/myjob.log"
JOB_NAME="My Scheduled Job"
NOTIFY_EMAIL="admin@example.com"    # (Optional)
SLACK_WEBHOOK_URL=""                # (Optional)

# === Setup File Descriptor for Lock ===
exec 200>"$LOCKFILE"
flock -n 200 || {
    echo "$(date '+%F %T') [$JOB_NAME] Another instance is already running. Exiting." >>
"$LOGFILE"
    exit 1
}

echo "$(date '+%F %T') [$JOB_NAME] Starting job..." >> "$LOGFILE"
START_TIME=$(date +%s)

# === Actual Job Logic Below ===
{
    # Simulate task
    echo "$(date '+%F %T') [$JOB_NAME] Performing main task..." >> "$LOGFILE"

    # Example: rsync, backup, or your actual job
    # /usr/bin/rsync -av /data /backup/ >> "$LOGFILE" 2>&1

    sleep 5 # simulate long task

    echo "$(date '+%F %T') [$JOB_NAME] Task completed successfully." >> "$LOGFILE"
} || {
    echo "$(date '+%F %T') [$JOB_NAME] Job failed." >> "$LOGFILE"
    # Optional alert:
    # echo "[$JOB_NAME] failed on $(hostname) at $(date)" | mail -s "$JOB_NAME Failed"
"$NOTIFY_EMAIL"
    exit 2
}

# === Job Completion Logging ===
```

Ekansh Gupta - DevOps Cron Job Template

```
END_TIME=$(date +%s)
DURATION=$((END_TIME - START_TIME))

echo "$(date '+%F %T') [$JOB_NAME] Completed in ${DURATION}s" >> "$LOGFILE"

# Optional Slack Alert (Uncomment if needed)
# if [ -n "$SLACK_WEBHOOK_URL" ]; then
#     curl -s -X POST -H 'Content-type: application/json' \
#     --data "{\"text\":\"[OK] $JOB_NAME completed on $(hostname) in ${DURATION}s.\"}" \
#     "$SLACK_WEBHOOK_URL"
# fi

exit 0
```

Usage Instructions

How to Use:

1. Save as `myjob.sh` and run `chmod +x myjob.sh`.
2. Add this to crontab: `*/15 * * * * /path/to/myjob.sh`
3. Logs will be written to `/var/log/myjob.log`

DevOps Best Practices

Key Best Practices Used:

- flock for safe job locking (prevents overlaps)
- exec 200> for persistent lock scope
- Structured logging with timestamps
- Exit codes for error visibility
- Optional Slack/email alerts
- Easily reusable config section