EN

Quick Reference Guide

Automate Linux Infrastructure

Crons, bash scripts and alerts

Mars 2026

Table of Contents

1. Crontab Syntax

```
# min  hour  day-month  month  day-week  command

# Examples:
# 0 3 * * *     = Every day at 3:00 AM
# 0 */6 * * *   = Every 6 hours
# 30 8 * * 1    = Monday at 8:30 AM
# */5 * * * *   = Every 5 minutes

crontab -e       # Edit
crontab -l       # List
```

2. Docker Backup Script

```
#!/bin/bash
BACKUP_DIR="/backup/docker/$(date +%Y-%m-%d)"
mkdir -p "$BACKUP_DIR"

# Ghost Blog
tar czf "${BACKUP_DIR}/ghost.tar.gz" \
    /mnt/docker-data/ghost-blog/content/data/

# Vaultwarden (brief stop)
docker stop vaultwarden
cp ~/docker/vaultwarden/data/db.sqlite3 \
    "${BACKUP_DIR}/vaultwarden.db"
docker start vaultwarden

# Cleanup +7 days
find /backup/docker -maxdepth 1 -type d -mtime +7 -exec rm -rf {} +
```

3. Telegram Alerts

```
#!/bin/bash
# /usr/local/bin/notify.sh
TELEGRAM_TOKEN="YOUR_TOKEN"
TELEGRAM_CHAT="YOUR_CHAT_ID"

notify() {
    local level="$1" message="$2"
    curl -s -X POST \
        "https://api.telegram.org/bot${TELEGRAM_TOKEN}/sendMessage" \
        -d "chat_id=${TELEGRAM_CHAT}" \
        -d "text=[$(hostname)] ${level}: ${message}" > /dev/null
}
export -f notify
```

4. Docker Auto-Update

```
#!/bin/bash
source /usr/local/bin/notify.sh
DIRS=("$HOME/docker/zoraxy" "$HOME/docker/vaultwarden")
UPDATED=0
for dir in "${DIRS[@]}"; do
    cd "$dir"
    if docker compose pull 2>&1 | grep -q "Downloaded newer"; then
        docker compose up -d
        UPDATED=$((UPDATED+1))
    fi
done
docker image prune -f
notify "OK" "Docker update: ${UPDATED} updated"
```

5. Logrotate + Docker Logs

```
# /etc/logrotate.d/docker-logs
/var/lib/docker/containers/*/*.log {
    daily
    rotate 7
    compress
    copytruncate
}

# Limit Docker log size
# /etc/docker/daemon.json:
{
  "log-driver": "json-file",
  "log-opts": { "max-size": "10m", "max-file": "3" }
}
sudo systemctl restart docker
```

6. Systemd Timers

```
# /etc/systemd/system/docker-backup.service
[Unit]
Description=Docker Backup
[Service]
Type=oneshot
User=ubuntu
ExecStart=/usr/local/bin/docker-backup.sh

# /etc/systemd/system/docker-backup.timer
[Unit]
Description=Docker Backup Timer
[Timer]
OnCalendar=*-*-* 03:00:00
Persistent=true
[Install]
WantedBy=timers.target

sudo systemctl daemon-reload
sudo systemctl enable --now docker-backup.timer
systemctl list-timers
```

7. Disk Monitoring

```
#!/bin/bash
source /usr/local/bin/notify.sh
THRESHOLD=85
while IFS= read -r line; do
    usage=$(echo "$line" | awk '{print $5}' | tr -d '%')
    mount=$(echo "$line" | awk '{print $6}')
    if (( usage >= THRESHOLD )); then
        notify "WARNING" "Disk ${mount}: ${usage}% used"
    fi
done < <(df -h | grep -E '^/dev' | grep -v tmpfs)
```

8. Full Crontab

```
# Daily backup 3:00 AM
0 3 * * * /usr/local/bin/docker-backup.sh >> /var/log/backup.log 2>&1

# Docker update Sunday 4:00 AM
0 4 * * 0 /usr/local/bin/docker-update.sh >> /var/log/update.log 2>&1

# Disk check every hour
0 * * * * /usr/local/bin/check-disk.sh

# Clean old logs Monday 5:00 AM
0 5 * * 1 find /var/log -name "*.log.gz" -mtime +30 -delete
```

Full article: botum.ca/automate-linux-infrastructure/

Website: www.botum.ca -

- Canada