

# מדריך מלא להגדרת משימה אוטומטית (Cron Job)

## בלינוקס

במדריך זה נלמד איך להגדיר משימה אוטומטית (Cron Job) שתורץ בזמן קבוע, לדוגמה כל בוקר בשעה 10:00 להריץ סקריפט עדכון וניקוי מערכת.

### מה זה Cron?

**cron** הוא שירות בלינוקס שמאפשר להריץ פקודות או סקריפטים אוטומטית בזמנים קבועים מראש. ניתן להגדיר פעולות יומיות, שבועיות או חודשיות.

### איך זה עובד?

ההגדרות נשמרות בקובץ מיוחד שנקרא `crontab`. כל שורה בקובץ מייצגת משימה אחת.

### מבנה שורה ב-crontab

#	m	h	dom	mon	dow	command
	*	*	*	*	*	הפקודה להרצה

- m** - דקה (0–59)
- h** - שעה (0–23)
- dom** - יום בחודש (1–31)
- mon** - חודש (1–12)
- dow** - יום בשבוע (0–6, כאשר 0=ראשון)
- command** - הפקודה או הסקריפט להרצה

### דוגמה למשימה יומית בשעה 10:00

נניח שיש לנו סקריפט בשם `maintenance.sh` שמבצע עדכונים וניקוי:

```
~/scripts/maintenance.sh
```

אנחנו רוצים שהוא ירוץ כל בוקר בשעה 10 וישמור רק את הלוג האחרון. נוסיף את השורה הבאה ל-`crontab`:

```
0 10 * * * ~/scripts/maintenance.sh > ~/scripts/maintenance.log 2>&1
```

## הסבר על השורה:

- 0 → הדקה (המשימה תתחיל בדיוק בדקה 00)
- 10 → השעה (10 בבוקר)
- \* \* \* → כל יום, כל חודש, כל יום בשבוע
- ~/scripts/maintenance.sh → הנתיב לסקריפט שלך
- > → מחליף את קובץ הלוג הישן בחדש
- 2>&1 → כולל גם הודעות שגיאה בלוג

💡 **טיפ:** אם היית משתמש ב- >> במקום > , הלוג היה מתווסף לסוף הקובץ במקום להחליף אותו.

## הרשאות הרצה לסקריפט

כדי שהמערכת תוכל להריץ את הסקריפט שלך (באופן ידני או אוטומטי), חייבות להיות לו הרשאות הרצה. כדי להגדיר זאת, הרץ:

```
chmod +x ~/scripts/maintenance.sh
```

כעת תוכל לוודא שההרשאה נוספה:

```
ls -l ~/scripts/maintenance.sh
```

👏 יפה מאוד!

הפלט מראה שהכל תקין לגמרי:

```
-rwxrwxr-x 1 rafi053 rafi053 1081 Oct 19 18:30 /home/rafi053/scripts/maintenance.sh
```

♦ המשמעות:

- rwx הראשון – אתה (המשתמש rafi053) יכול לקרוא (r), לכתוב (w) ולהריץ (x) את הסקריפט.
- rwxr-x – גם קבוצה ואחרים יכולים להריץ, אבל לא לשנות.
- כלומר הקובץ ניתן להרצה, בדיוק כמו שצריך.

✅ עכשיו אתה יכול:

- להריץ אותו ידנית:

```
~/scripts/maintenance.sh
```

- או לתת ל-cron להריץ אותו אוטומטית כל יום בשעה 10:00 בבוקר (כפי שהוגדר).

אם אתה רוצה לוודא שגם **שירות cron עצמו רץ** (כדי לוודא שהמשימה תתבצע בזמן), תוכל לבדוק כך:

```
systemctl status cron
```

אם תראה **active (running)** – הכל מוכן לעבודה 🟢

## מה לגבי בקשת סיסמה (sudo)?

כשמריצים את הסקריפט ידנית, ייתכן שתתבקש להזין סיסמת sudo עבור פקודות כמו `apt update`. אבל כש-cron מריץ את הסקריפט, אין לו דרך להזין סיסמה. הנה שתי דרכים להתמודד עם זה:

### אפשרות 1 – להריץ בלי sudo

אם זה מחשב אישי שלך, והמשתמש שלך כבר בעל הרשאות מתאימות, פשוט הסר את המילה `sudo` מהפקודות בסקריפט שלך:

```
apt update
apt upgrade -y
apt autoremove -y
apt autoclean
rm -rf /tmp/*
```

כך הסקריפט ירוץ ללא צורך בסיסמה, גם מתוך cron.

### אפשרות 2 – לאפשר פקודות ספציפיות ללא סיסמה

אם אתה מעדיף להשאיר את `sudo`, תוכל להגדיר שהמערכת תריץ פקודות מסוימות בלי לבקש סיסמה:

1. פתח את קובץ ההרשאות:

```
sudo visudo
```

2. הוסף שורה (החלף את שם המשתמש שלך):

```
rafi053 ALL=(ALL) NOPASSWD: /usr/bin/apt, /usr/bin/apt-get
```

3. שמור וצא.

עכשיו ה-cron יוכל להריץ את הפקודות האלו בלי להזין סיסמה.

אם זו מערכת אישית שלך – הדרך הראשונה (להוריד sudo) לרוב פשוטה ובטוחה יותר. 💡

## איך יוצרים או עורכים crontab

כדי לפתוח את קובץ ה-cron שלך, הרץ בטרמינל:

```
crontab -e
```

אם זו הפעם הראשונה, תתבקש לבחור עורך (nano, vim, VS Code). מומלץ לבחור VS Code עם האפשרות `code --wait`.

## הגדרת VS Code כעורך ברירת מחדל

כדי ש- `crontab -e` יפתח אוטומטית ב-VS Code, יש להריץ את שתי הפקודות הבאות:

```
export VISUAL="code --wait"
export EDITOR="$VISUAL"
```

הן מגדירות את VS Code כעורך ברירת מחדל ומבטיחות שהמערכת תמתין שתסגור את החלון לפני שתמשיך. כדי לשמור זאת באופן קבוע, הוסף את שתי השורות האלו לקובץ `~/.zshrc` או `~/.bashrc` שלך והריץ:

```
source ~/.zshrc
```

כך בכל פעם שתפתח את `crontab`, הוא יפתח אוטומטית ב-VS Code. 💡

## הרצה ידנית של הסקריפט

אם אתה רוצה להריץ את הסקריפט שלך ידנית, לדוגמה כדי לבדוק אותו או להפעיל אותו מחוץ ל-cron, פשוט הרץ את הפקודה:

```
~/scripts/maintenance.sh
```

אם תרצה לראות בזמן אמת את הפלט (log) שהפקודה מייצרת, תוכל להשתמש ב:

```
~/scripts/maintenance.sh | tee ~/scripts/maintenance.log
```

כך תראה את הפלט על המסך וגם תשמור אותו בלוג.

## שמירה ובדיקה

לאחר ששמרת את הקובץ, תוכל לבדוק שה-cron שלך נוסף בהצלחה:

```
crontab -l
```

זה יציג את כל המשימות האוטומטיות שלך.

## דוגמה מלאה לסקריפט תחזוקה

```
#!/bin/bash
# סקריפט לתחזוקה שוטפת

echo " ♦ מתחיל עדכון מערכת..."
apt update
apt upgrade -y
apt autoremove -y
apt autoclean
rm -rf /tmp/*
echo "✅!התחזוקה הסתיימה בהצלחה"
```

עכשיו כל בוקר בשעה 10:00 הסקריפט ירוץ לבד, יעדכן וינקה את המערכת, ויכתוב את התוצאה ללוג אחד בלבד. ✅