

# Référence rapide Cron

Planification de tâches, syntaxe crontab, chaînes spéciales, journalisation

## Syntaxe

### Format crontab

```
# | minute (0-59)
# | heure (0-23)
# | jour du mois (1-31)
# | mois (1-12)
# | jour de la semaine (0-7, 0&7=Dim)
# * * * * * commande
```

### Valeurs des champs

**\*** Toute valeur (joker)  
**,** Liste de valeurs : **1,3,5**  
**-** Plage : **1-5**  
**/** Pas : **\*/10** (toutes les 10 unités)  
**0-59** Plage des minutes  
**0-23** Plage des heures  
**1-31** Plage du jour du mois  
**1-12** Plage du mois (ou JAN-DEC)  
**0-7** Jour de la semaine (0&7=Dim, ou SUN-SAT)

## Chaînes spéciales

### Planifications abrégées

**@reboot** Exécuter une fois au démarrage  
**@yearly** **0 0 1 1 \*** — une fois par an (1er jan)  
**@monthly** **0 0 1 \* \*** — une fois par mois (le 1er)  
**@weekly** **0 0 \* \* 0** — une fois par semaine (dimanche)  
**@daily** **0 0 \* \* \*** — une fois par jour (minuit)  
**@hourly** **0 \* \* \* \*** — une fois par heure

## Exemples

### Planifications courantes

```
30 2 * * * /usr/local/bin/backup.sh
0 9 * * 1-5 /opt/report.sh
*/15 * * * * /usr/bin/health-check.sh
0 0 1 * * /opt/monthly-cleanup.sh
0 8,17 * * * /opt/notify.sh
```

### Détail des planifications

**30 2 \* \* \*** Tous les jours à 2h30  
**0 9 \* \* 1-5** Jours ouvrés à 9h00  
**\*/15 \* \* \* \*** Toutes les 15 minutes  
**0 0 1 \* \*** Premier jour du mois, minuit  
**0 8,17 \* \* \*** Tous les jours à 8h et 17h

## Environnement

### Définir des variables

```
SHELL=/bin/bash
PATH=/usr/local/bin:/usr/bin:/bin
MAILTO=admin@example.com
HOME=/home/deploy
* * * * * /opt/myjob.sh
```

### Notes sur l'environnement

**SHELL** Shell utilisé pour les commandes (défaut **/bin/sh**)  
**PATH** Chemin de recherche (minimal par défaut dans cron)  
**MAILTO** Envoyer la sortie à cette adresse (vide = pas de mail)  
**HOME** Répertoire home pour la tâche cron

## Journalisation

### Sortie et journalisation

```
# rediriger stdout + stderr vers un fichier de log
*/5 * * * * /opt/job.sh >> /var/log/job.log 2>&1
# ignorer toute la sortie
0 3 * * * /opt/job.sh > /dev/null 2>&1
# journaliser uniquement les erreurs
0 3 * * * /opt/job.sh > /dev/null 2>> /var/log/job-err.log
```

### Consulter les logs cron

```
# Debian / Ubuntu (syslog)
grep CRON /var/log/syslog
# RHEL / CentOS
cat /var/log/cron
# journalctl (systemd)
journalctl -u cron --since "1 hour ago"
```

## Planifications courantes

### Patterns du quotidien

**0 \* \* \* \*** Toutes les heures pile  
**\*/5 \* \* \* \*** Toutes les 5 minutes  
**\*/30 \* \* \* \*** Toutes les 30 minutes  
**0 0 \* \* \*** Tous les jours à minuit  
**0 6 \* \* 1** Tous les lundis à 6h  
**0 0 1,15 \* \*** Le 1er et le 15 du mois  
**0 0 \* \* 5** Tous les vendredis à minuit  
**0 0 1 1 \*** Chaque 1er janvier

## Commandes crontab

### Gérer les crontabs

```
crontab -e # éditer votre crontab
crontab -l # lister votre crontab
crontab -r # supprimer votre crontab
crontab -u alice -l # lister le crontab d'un autre (root)
crontab /path/to/file # installer le crontab depuis un fichier
```

### Emplacements système des crontabs

**/etc/crontab** Crontab système (a un champ utilisateur)  
**/etc/cron.d/** Fichiers cron supplémentaires  
**/etc/cron.daily/** Scripts exécutés quotidiennement  
**/etc/cron.hourly/** Scripts exécutés toutes les heures  
**/etc/cron.weekly/** Scripts exécutés hebdomadairement  
**/etc/cron.monthly/** Scripts exécutés mensuellement  
**/var/spool/cron/** Stockage des crontabs par utilisateur

## Conseils

### Bonnes pratiques

**Utiliser des chemins absolus** Le PATH de cron est minimal ; toujours utiliser des chemins complets  
**Rediriger la sortie** Éviter les envois de mails avec **> /dev/null 2>&1**  
**Fichiers verrou** Utiliser **flock** pour éviter les exécutions simultanées  
**Tester d'abord** Exécuter la commande manuellement avant de planifier  
**Utiliser MAILTO** Définir **MAILTO=""** pour supprimer tous les mails  
**Vérifier le fuseau horaire** Cron utilise l'heure système ; vérifier avec **date**

## Éviter le chevauchement avec flock

```
# exécuter seulement si pas déjà en cours
* * * * * /usr/bin/flock -n /tmp/job.lock /opt/job.sh
```