



Linux et Bash - Crontab

🕒 30 minutes 📁 Normal



DataScientest • com

Introduction à Linux

Machine status



Ubuntu
Server
18.04 LTS
SSD
Volume
Type
64-bit x86

Online

34.245.135.23



Connect

Reset



Stop

Crontab

Qu'est ce que Crontab ?

Cron tire son nom du mot grec "Chronos", qui désigne le temps.

Il s'agit d'un processus qui exécute automatiquement des tâches selon un programme spécifique. C'est un ensemble de commandes qui sont utilisées pour exécuter des tâches régulières de planification. Crontab est également le nom du programme qui est utilisé pour modifier cette planification. Il est piloté par un fichier crontab, un fichier de configuration qui indique les commandes shell à exécuter périodiquement pour le programme spécifique.

Crontab est souvent utilisé pour l'automatisation de tâches telles que la gestion des logs, le nettoyage des données dans le cache, envoyer des emails de newsletter ou encore automatiser votre script de scraping de données.

Syntaxe d'un Cron Job

Un Cron Job est une automatisation de tâches à réaliser. La syntaxe est la suivante :

```
1 * * * * * command/script
2
```

De gauche à droite :

- Le premier * correspond aux Minutes (0-59) ;
- Le deuxième * correspond aux Heures (0-23) ;
- Le troisième * correspond au Jour du mois (1-31) ;
- Le quatrième * correspond au Mois de l'année (1-12) ;
- Le cinquième * correspond au Jour de la semaine (0-6, du dimanche au samedi).

Pour spécifier plusieurs valeurs dans un champ, utilisez les symboles d'opérateur suivants :

- Astérisque (*) : Pour spécifier toutes les valeurs possibles pour un champ ;
- Tiret (-) : Pour spécifier une plage de valeurs ;



Exemple

Voici quelque exemple grâce à la syntaxe ci-dessus :

Exécutez un Cron Job à 5h15 tous les jours.

```
1 15 5 * * * command/script
2
```

Exécutez un Cron Job à 5h15 chaque 2eme jour du mois.

```
1 15 5 2 * * command/script
2
```

Exécutez un Cron Job toutes les 5h.

```
1 0 */5 * * * command/script
2
```

Exécutez un Cron Job chaque lundi et mercredi du mois de janvier et février.

```
1 0 0 * jan,feb mon,wed command/script
2
```

Mise en pratique

Créer le script `cron.sh` suivant :

```
1 #!/bin/bash
2 echo "Cron Job" >> ~/data_cron.txt
3
```

Ensuite, donnez les droits d'exécution au script `cron.sh` via la consigne suivante :

```
1 chmod +x ~/cron.sh
2
```

À présent, exécutez la commande `crontab -e` pour ajouter un job.

Insérez dans le fichier la ligne : `*/1 * * * * ~/cron.sh` .

Le Cron Job défini exécute notre script à intervalle de 1 minute : Observez l'évolution du fichier `data_cron.txt` au fil des minutes.

Commandes usuelles

- `crontab -e` : Pour modifier le fichier crontab de l'utilisateur actuel ;
- `crontab -l` : Pour afficher le contenu du fichier crontab ;
- `crontab -u [nom d'utilisateur]` : Pour modifier le fichier crontab d'un autre utilisateur ;
- `crontab -r` : Pour supprimer le fichier crontab de l'utilisateur actuel ;
- `crontab -i` : Pour afficher une invite avant de supprimer le fichier crontab de l'utilisateur actuel.

Validated

