

Maintenance automatisée d'une base de données MySQL

Dans cet exercice, nous voulons automatiser la maintenance de nos bases de données, c'est à dire planifier et exécuter la vérification des tables de la base de données ainsi que leurs sauvegardes sans intervention humaine. Ce procédé permet d'éviter ses taches chronophages à l'équipe de développement mais est aussi utile en cas de perte ou corruption des données si cela venait à se produire lorsqu'aucun membre de l'équipe ne soit présent.

Contexte du projet

Suivant l'environnement sous lequel vous travaillez, automatisez à partir de cron (Linux) ou du planificateur de taches (Windows) les taches suivantes :

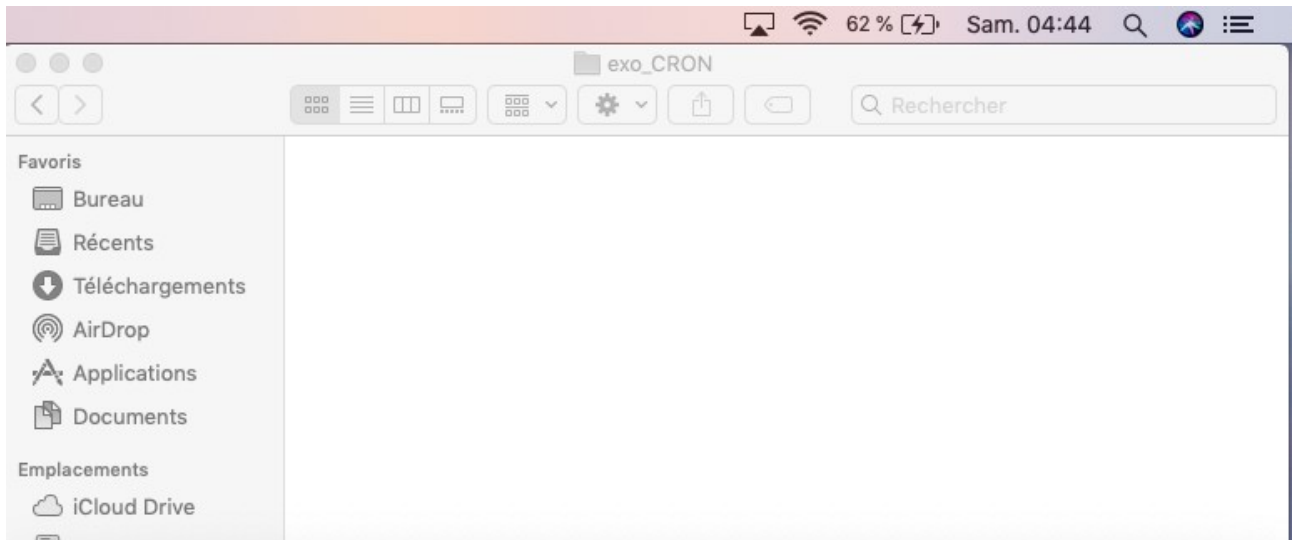
- sauvegarder Sakila en utilisant mysqldump
- sauvegarder toutes les bases de données en utilisant mysqldump
- faire un check de la base de données netflix et Simplon en utilisant mysqlcheck
- faire un check de toutes les bases de données en utilisant mysqlcheck
- restaurer le dump de Sakila sous un nom différent

La fréquence d'exécution est arbitraire.

La première étape consiste à vérifier que le processus que nous souhaitons mettre en place réponds aux instructions que nous voulons lui soumettre. Nous commençons par créer un dossier que l'on appelle **exo_CRON** dans lequel nous effectuerons nos essais.

```
-dossier a00000$ mkdir exo_CRON  
-dossier a00000$
```

Nous vérifions que le dossier est bien crée et qu'il ne contient aucun fichier.



Après avoir constaté que le dossier est vide, nous programmons une sauvegarde de la base de donnée **CLIIINK2** via le gestionnaire de tâche **CRON**. Pour se faire nous ouvrons l'éditeur de tâche, nous indiquons dans un premier temps à quel moment l'on souhaite enregistrer la base de données, **45(min) 4(heure) 12(jour) 12(mois) *(jour de la semaine)**, puis nous utilisons **mysqldump** pour sauvegarder le script **sql** de la base de données ainsi que les données insérées sous le nom de **test_cron.sql**.

Pour des raisons évidentes de sécurité nous dissimulons les mots de passe utilisés.

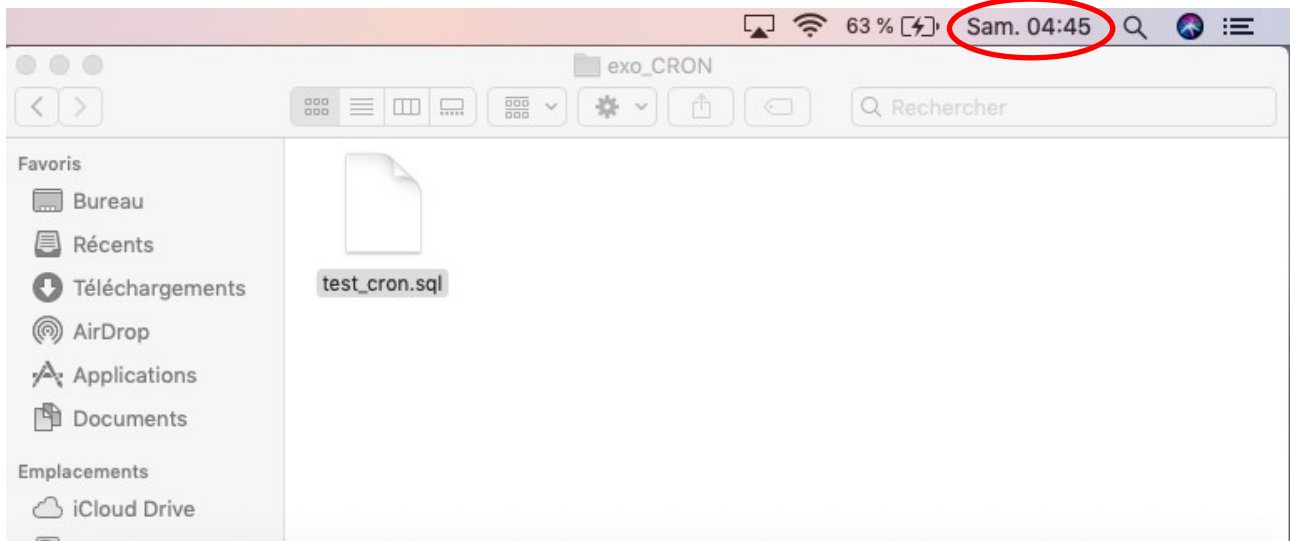
```
GNU nano 2.8.6 File: /tmp/crontab.jnkCPrLeu4  
45 4 12 12 * /usr/local/mysql/bin/mysqldump -u root --password= CLIIINK2 > /Users/a0000/Desktop/Dossier_simplon/premier-dossier/exo_CRON/test_cron.sql
```

Nous vérifions maintenant que la tâche a bien été enregistrée dans le gestionnaire de tâche **CRON**.

```
(base) MacBook-Pro-de-aaaa:exo_CRON a00000$ crontab -l  
45 4 12 12 * /usr/local/mysql/bin/mysqldump -u root -  
(base) MacBook-Pro-de-aaaa:exo_CRON a00000$
```

Nous constatons que la tâche est bien enregistrée dans le gestionnaire de tâche, nous pouvons maintenant nous assurer que la tâche est bien exécutée à l'heure indiqué. Pour se faire, nous attendons simplement qu'à l'heure que nous avons indiqué la base de données soit sauvegardée sous le **test_cron.sql**.

Nous constatons que le fichier **test_cron.sql** est bien apparu à l'heure indiquée.



Maintenant que nous avons vérifié que l'efficacité de l'automatisation de la sauvegarde des données, nous pouvons automatiser la maintenance de notre bases de données. En suivant le même procédé nous implémentons une tâche qui va automatiser la sauvegarde de la base de donnée **sakila** dans un fichier que l'on va nommer **save_sakila.sql**, sauvegarder toutes les bases de données dans un fichier que l'on va nommer **save_all.sql**, faire un check de la base de données **netflix** et **Simplon**, faire un check de toutes les bases de données puis restaurer le dump de **sakila** sous un nom différent.

Ici nous remplaçons la base de données sakila par la base de données CLIINK2 et le base de données Simplon par exo_netflix

Pour restaurer la base de données **CLIINK2** en changeant de nom, nous commençons par créer une base de données vide que l'on va appeler **testdb1**.

```
[(base) MacBook-Pro-de-aaaa:~ a0000$ /usr/local/mysql/bin/mysqladmin -u root --password= ... create testdb1
```

Puis on implémentes les tâches suivantes :

```
* * * 1 /usr/local/mysql/bin/mysqldump -u root --password= ... CLIINK2 > /Users/a0000/Desktop/Dossier_simplon/premier-dossier/exo_CRON/save_CLIINK2.sql
* * * 1 /usr/local/mysql/bin/mysqldump -u root --password= ... --all-databases > /Users/a0000/Desktop/Dossier_simplon/premier-dossier/exo_CRON/save_ALL.sql
* 1 * * /usr/local/mysql/bin/mysqlcheck -u root --password= ... --databases CLIINK2 exo_netflix > /Users/a0000/Desktop/Dossier_simplon/premier-dossier/exo_CRON/sa$
* 1 * * /usr/local/mysql/bin/mysqlcheck -u root --password= ... --all-databases CLIINK2 > /Users/a0000/Desktop/Dossier_simplon/premier-dossier/exo_CRON/save_all.t$
* * * 1 /usr/local/mysql/bin/mysql -u root --password= ... testdb1 < /Users/a0000/Desktop/Dossier_simplon/premier-dossier/exo_CRON/save_CLIINK2.sql
```

Description des taches :

- La première tache effectue une sauvegarde de la base de données **CLIIINK2** tout les lundi de chaque semaine dans un fichier que nous nommons **save_CLIIINK2.sql**
- La deuxième tache sauvegarde toutes les bases de données tout les lundi de chaque semaine dans un fichier que nous nommons **save_ALL.sql**
- La troisième tache effectue une vérification de la base de données **CLIIINK2** et de **exo_netflix** tout les jours à 1h et renvoie les résultats dans un fichier que l'on nomme **check_CLIIINK2_and_exo_netflix.txt**
- La quatrième tache effectue une vérification de toutes les bases de données tout les jours à 1h et renvoie les résultats dans un fichier que l'on nomme **check_all.txt**
- La dernière tache récupère les données sauvegarder dans le fichier **save_clink2.sql** tous les lundi de chaque semaine et insère son contenu dans la base de données **testdb1**

Nous avons choisi d'effectuer les sauvegardes des bases de données 1 fois par semaines puisqu'il n'y a pas fréquemment d'insertion ou de suppression de données.

Nous choisissons quand même d'effectuer la vérification des bases de données tous les jours à 1 heure par précaution.