

**Synchronisation automatique des données critiques avec rsync**

**Mbolanirina Stephano Kevin**

**Joma Alfred Moustaki**

Sous la direction de **Mr Herimampionina**

Année universitaire 2024-2025

**Table des matières**

**[Introduction 4](#_Toc105393704)**

**[Partie 1 - Choix des technologies 5](#_Toc748699769)**

[Pourquoi rsync ? 6](#_Toc1386494667)

[Utilisation de cron pour automatiser 6](#_Toc658958098)

**[Partie 2 - Mise en oeuvre de la synchronisation 7](#_Toc1513386544)**

[Commande de base rsync 8](#_Toc1753587921)

[Script pour la synchronisation 8](#_Toc1528593652)

[Permission et execution du script 9](#_Toc563898457)

**[Partie 3 - Automatisation des sauvegardes 10](#_Toc294363148)**

[crontab: configuration et fréquence 11](#_Toc207519393)

**[Partie 4 - Restauration des données 13](#_Toc566693052)**

[Script pour la restauration 14](#_Toc672677324)

[Test de restauration 14](#_Toc1027650515)

**[Conclusion 16](#_Toc108652203)**

# Introduction

Dans le contexte actuel de la cybersécurité, la préservation et la récupération des données critiques sont des enjeux majeurs pour toute infrastructure informatique. Que ce soit pour un particulier, une petite entreprise ou un système à grande échelle, une solution de sauvegarde fiable, simple et automatisée est indispensable.

Ce projet s’inscrit dans une démarche pédagogique et technique : il s'agit de concevoir, mettre en œuvre et documenter un système de sauvegarde automatisé basé sur des outils robustes et accessibles tels que rsync et cron. L’objectif principal est d’assurer la sauvegarde régulière de données critiques, tout en permettant leur restauration simple et efficace, le tout à travers des scripts bash clairs et reproductibles.

# Partie 1 - Choix des technologies

## Pourquoi rsync ?

Avant de repondre à cette question, il est nécessaire de savoir que ce qu’est réelement rsync. rsync pour Remote Synchronization est un logiciel de synchronisation de fichiers. rsync travaille de manière unidirectionnelle c’est-à-dire qu’il synchronise, copie ou actualise les données d’une source (locale ou distante) vers une destination (locale ou distante) en ne transferant que les octets des fichiers qui ont été modifiés.

Il a été décidé d’utiliser l’outil **rsync** en raison de sa facilité d’utilisation notament avec des commandes très simples et une configuration assez facile. *rsync* fonctionne sur une large gamme de système d’éxploitation (Microsoft Windows, Mac OS, Linux, Unix) permettant ainsi de synchroniser des données à travers différents système d’éxploitation. Il est similaire à **rcp** (remote copy) d’unix mais avec plus de fonctionnalités et de nombreuses options suplémentaires. Il utilise un protocole de mise à jour à distance très éfficace permettant d’accelerer le transfert de fichies si le fichier de destination existe déja. On peux citer quelques uns de ces avantages ci-dessous:

* Préservation de la structure des données et permissions
* Utilisable en réseau et en local
* Fournit des **logs** détaillés
* Très personnalisable
* Automatisable

## Utilisation de cron pour automatiser

**cron** est un programme qui permet aux utilisateurs Linux d’executer automatiquements des commandes, des scripts, ou des logiciels à une date et une heure spécifiée à l’avance, ou selon un cycle defini.

C’est une fonctionnalité très utile pour des tâches d’administration système. **cron** est *daemon*, ce qui désigne un programme qui s’éxecute en arrière plan. Le service cron attend ainsi le momoent spécifié dans le fichier de configuration (que l’on appelle la **crontab** ) puis effectue l’action correspondante et se rendort jusqu’à l’événement suivant.

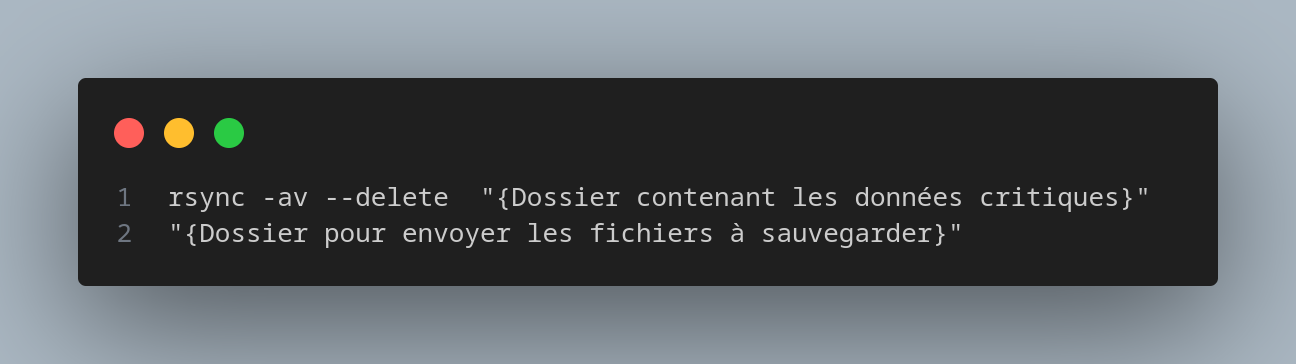
Utiliser cron est donc indispensable pour ce projet afin d’automatiser les sauvegardes et la synchronisation des données. Toutes les configurations nécessaire à la réalisation de ce projet seront donc expliqué dans la partie suivante.

# Partie 2 - Mise en oeuvre de la synchronisation

La mise en place d’un système de synchronisation fiable repose sur une série d’étapes allant de la configuration initiale jusqu’à l’automatisation du processus. Cette partie va décrire la création de scripts de synchronisation simple mais qui fonctionne bien.

## Commande de base rsync

Pour commencer, il est impourtant de preciser que la synchronisation va se faire en local, ce qui veut dire dans la même machine. Avec rsync, la commande Linux qui peut permettre de faire une synchronisation est la suivante:



* **rsync** : programme utilisé pour synchroniser des données entre deux emplacements
* **-a** : active le mode **archive**
* **-v** : active le mode verbose. Il permet d’afficher tous les actions faites par rsync
* **--delete** : supprime dans le dossier de destination tous les fichiers qui n’existent plus dans le dossier source

Ceci est juste la commande de base. Pour une action plus précise et complète, il va faloir créer un script.

## Script pour la synchronisation

Le script suivant va faire la synchronisation des données tout en génerant un fichier de log pour exécution. Cela va permettre un suivi précis de chaque opération.

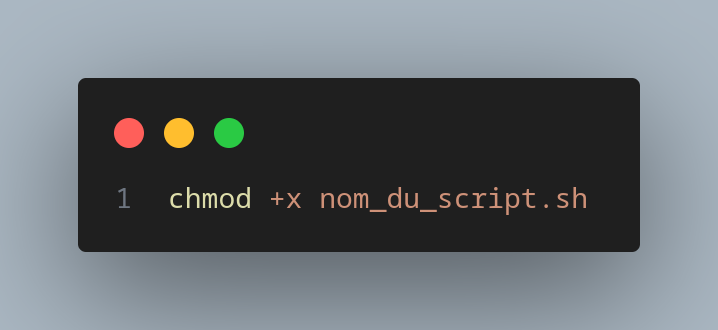
*L’éxtension du fichier doit être* **.sh**



* **DATE** : c’est une variable au format **année-mois-jour\_heure-minute** pour initialiser la date
* **LOGFILE** : fichier de log qui contiendra les détails de l’opération, nommé avec la date

## Permission et execution du script

Pour faciliter l’exécution du script ci-dessous, il est nécessaire de lancer la commande:



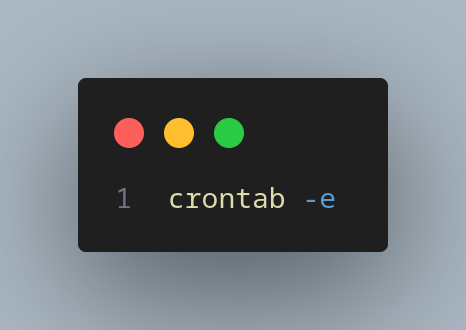
Ainsi, il est possible d’éxecuter le script et lancer la synchronisation. Mais le vrai but du projet est d’automatiser la synchronisation. La configuration de cette automatisation sera expliquer dans la parit suivante.

# Partie 3 - Automatisation des sauvegardes

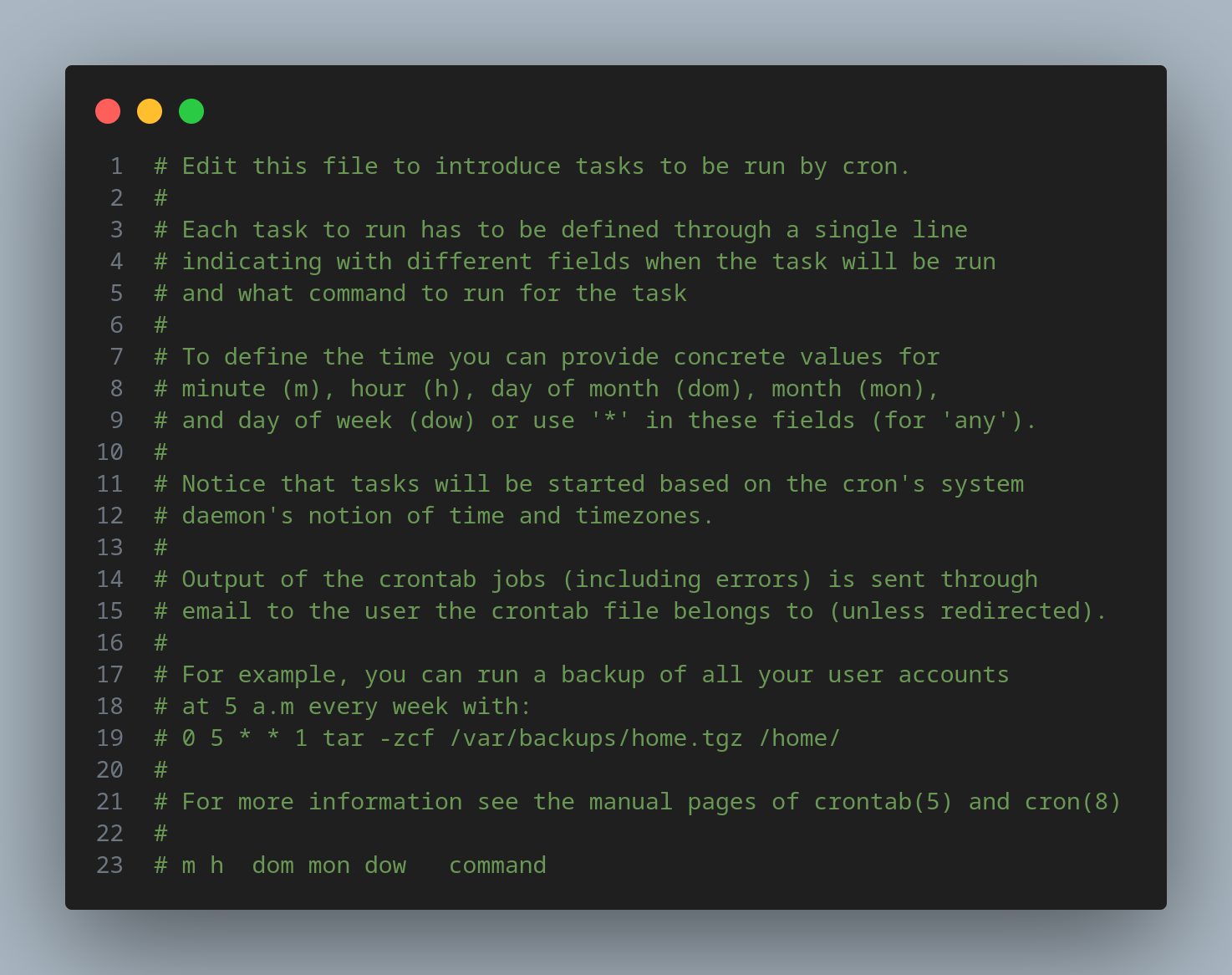
La fiabilité d’un système de sauvegarde repose non seulement sur sa capacité à préserver les données, mais aussi sur sa régularité d’exécution. Pour assurer que les synchronisations se fassent de manière systématique sans intervention humaine, il est nécessaire de les automatiser. Cette section présente l’utilisation de l’outil **cron**, un planificateur de tâches sous Linux, permettant de lancer le script de synchronisation à intérvales réguliers.

## crontab: configuration et fréquence

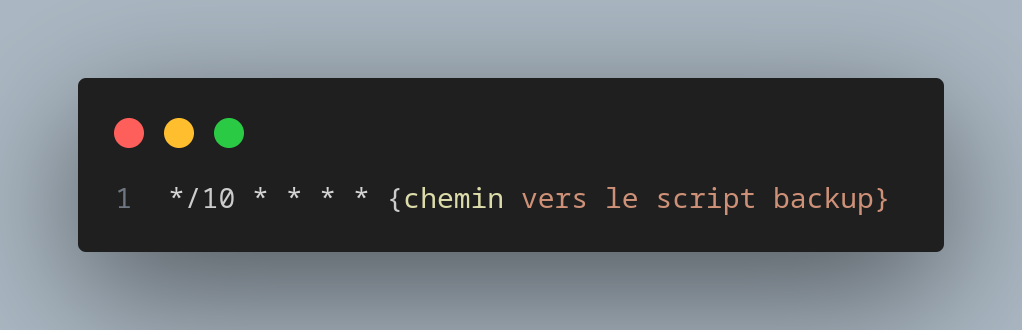
Le moyen de faire la configuration de l’automatisation de la synchronisation est d’ouvrir **crontab** dans un éditeur de texte avec la commande :



Dans ce fichier de configuration, il y a ces lignes :

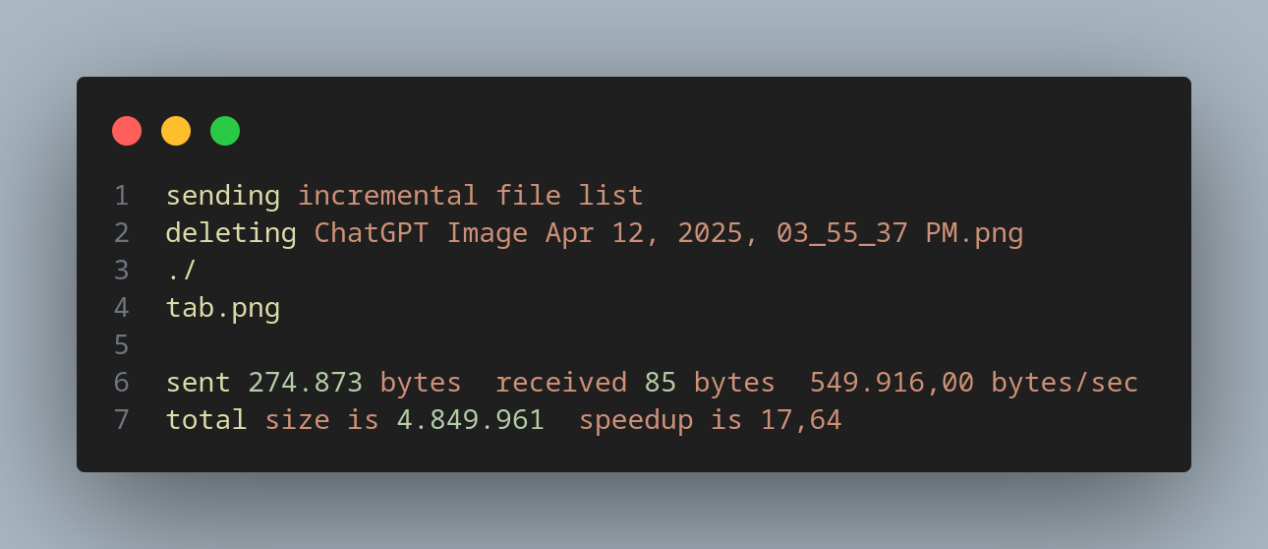


Dans le cadre du projet, nous allon ajouter la ligne ci-dessous dans ce fichier de configuration:



* cette ligne veut dire que le script backup de synchronisation va se faire toutes les **dix minutes** ( 10 minutes est juste un exemple ).

Une fois la configuration enregistrée, l’observation du fichier de log permet de constater que la synchronisation s’est réelement fait dans le délai demandé :



Ce fichier de log dit que :

* suppression de «ChatGPT Image Apr 12, 2025, 03\_55\_37 PM.png»
* **Ajout** de «tab.png»

Le principale but du projet est maintenant atteint. Pour une amélioration et pour plus d’option, il est très impourtant de pouvoir faire des restaurations.

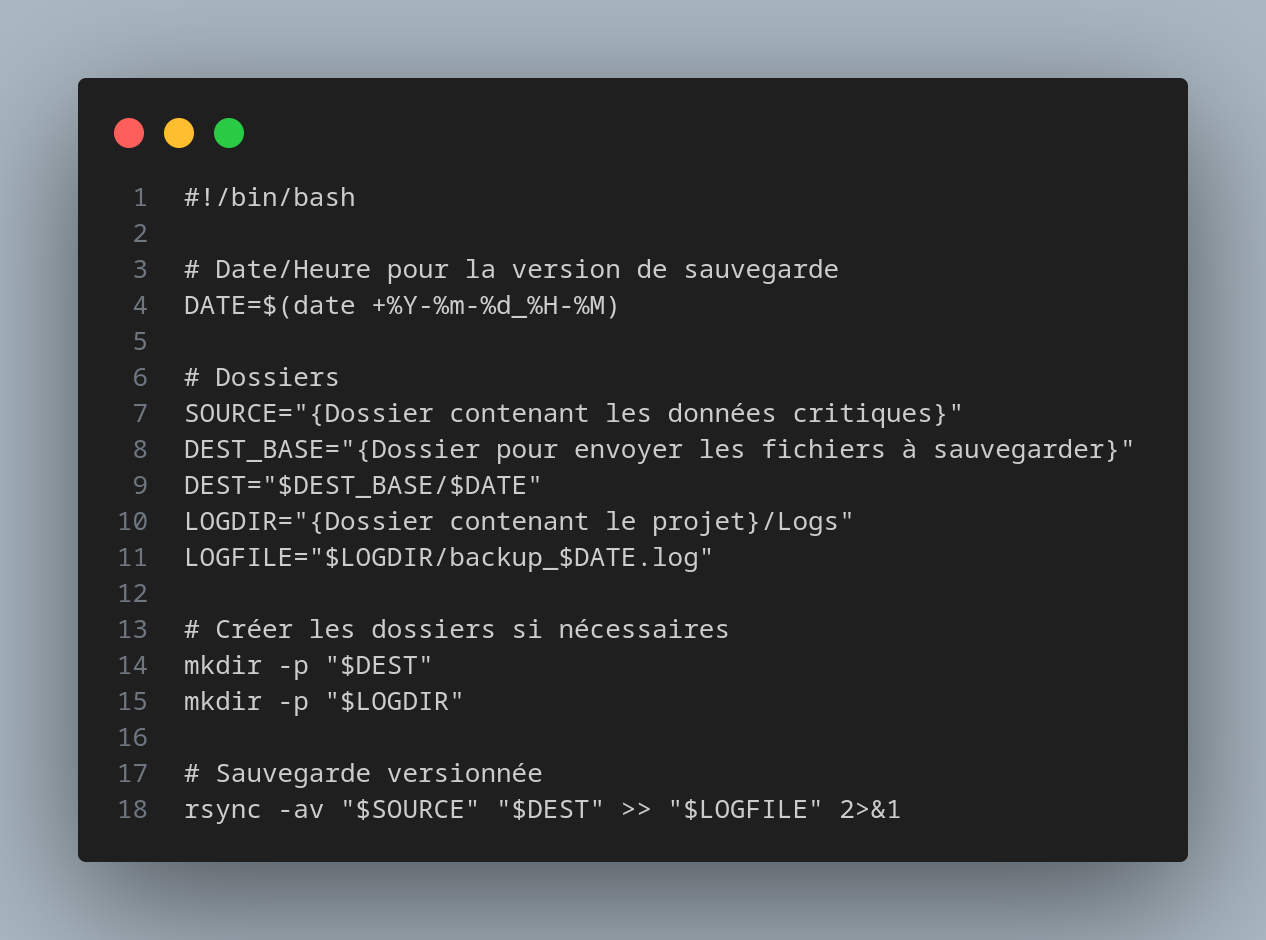
C’est ce qui sera éxpliquer dans la prochaîne séction.

# Partie 4 - Restauration des données

Une sauvegarde efficace ne doit pas se limiter à la simple copie des fichiers: elle doit également faciliter la restauration rapide et fiable des données en cas de besoin. Cette section explique comment organiser les sauvegardes sous forme de version datées pour preserver l’historique des fichiers, et détaille le processus permettant de restaurer les données à partir de ces copies. Le but est de garantir une récupération facile en cas de suppression par accident ou de corruption des données originales.

## Script pour la restauration

Le script suivant permettra de faire la restauration des fichiers :



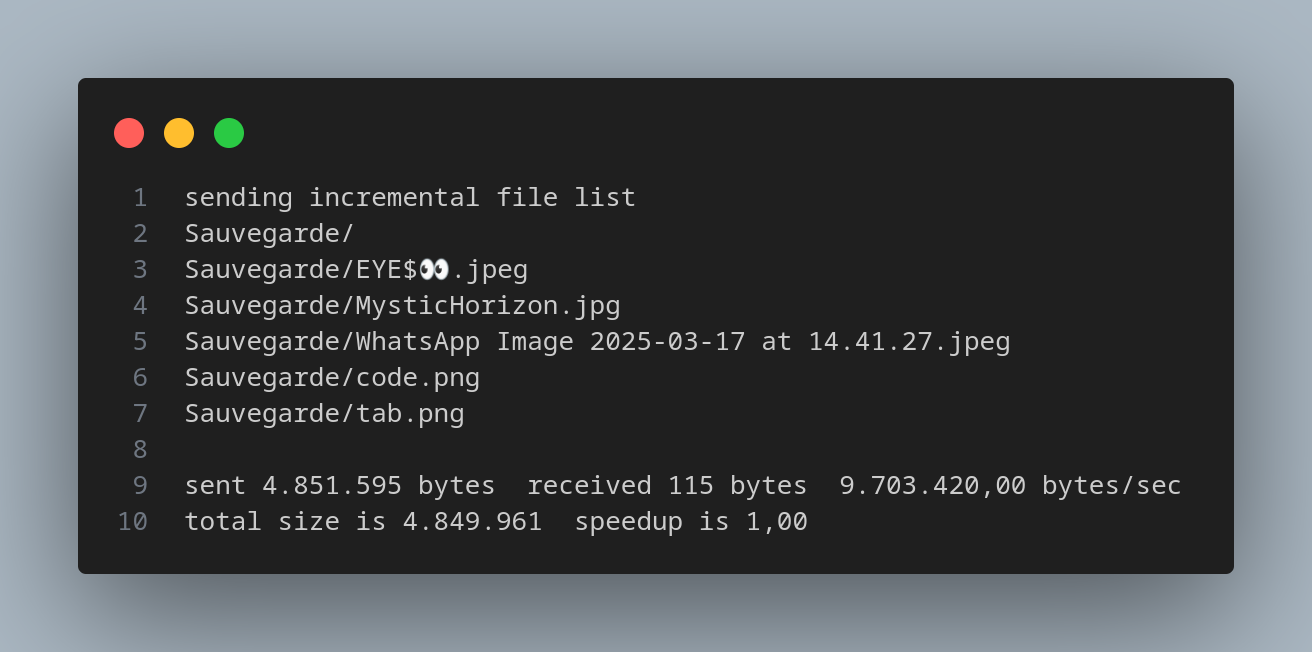
En faisant la restauration, ce script va en meme temps créer un dossier datté contenant les fichiers restaurer.

## Test de restauration

Afin de verifier que tout fonctionne bien, il est necessaire de faire des tests. Pour ce faire, il faut suivre les étapes suivantes :

* Effacer un ou plusieurs fichiers déja sauvegardés du dossier contenant les données critiques
* executer le script de restauration
* visualiser les logs pour verifier

Une fois ces étapes faites, le fichier de log doit contenir quelque chose comme:



C’est la preuve que la restauration s’est fait avec succés!

# Conclusion

La mise en place d’un système de synchronisation automatisé constitue une étape essentielle pour la protection des données critiques. À travers ce projet, une solution simple, fiable et efficace a été développée en s'appuyant sur des outils robustes tels que rsync et cron. La création de scripts adaptés, accompagnée de l'organisation rigoureuse des fichiers de logs et du versionnage des sauvegardes, permet d’assurer non seulement la conservation des données, mais aussi leur restauration rapide en cas de besoin.

Cette approche, tout en étant accessible et légère, offre une grande flexibilité et peut être facilement adaptée à des environnements plus complexes. Le projet met ainsi en évidence l'impourtance d'anticiper les pertes potentielles de données et de disposer de mécanismes de synchronisations adaptés aux besoins spécifiques de chaque système.

En perspective, l’intégration d’outils de notification, la mise en place d’interfaces graphiques ou encore l'utilisation de solutions de synchronisation sur le cloud pourraient enrichir cette base pour offrir une protection encore plus complète.