🧠 **Kolejność wykonywania plików .bat po wygenerowaniu – PASZKO\_DEPLOY vFINAL**

Dzięki tym krokom uruchomisz **pełny, bezpieczny system backupu GPT z automatycznym pushem na GitHub**, bez ryzyka wycieku tokena. Oto dokładna sekwencja:

**✅ 1. git\_credentials\_enable.bat**

🔧 **Wykonaj tylko raz** na nowej maszynie

➤ Co robi:

* Ustawia Twoje dane użytkownika Git
* Włącza zapamiętywanie tokena (credential manager)

**✅ 2. reset\_token\_history.bat**

🔥 **Wykonaj tylko raz** jeśli kiedykolwiek commitowałeś token

➤ Co robi:

* Usuwa plik push\_with\_token.bat z całej historii Git
* Oczyszcza repozytorium z sekretów
* Wyświetla instrukcję do git push --force --all – **musisz wykonać ją ręcznie**!

📌 *Po tej operacji historia GIT zostaje przepisana – to finalny reset.*

**✅ 3. push\_with\_token\_safe.bat**

🔁 **Używaj zawsze od teraz**, jeśli chcesz wypchnąć ręcznie zmiany na GitHub

➤ Co robi:

* Wykonuje git push origin main
* Token pochodzi z credential managera, nie z pliku

**✅ 4. run\_and\_push\_FIXED.bat**

🚀 **Główne centrum dowodzenia – uruchamiaj codziennie lub przez harmonogram**

➤ Co robi:

* Wykonuje backup danych (backup\_engine.py)
* Potem automatycznie robi push (call push\_with\_token\_safe.bat)

**🗓️ 5. register\_backup\_task.vbs *(opcjonalne)***

📅 Rejestruje codzienne automatyczne uruchamianie run\_backup\_daily.bat  
🔁 Jeśli chcesz zamiast tego używać run\_and\_push\_FIXED.bat codziennie – daj znać, wygeneruję register\_push\_task.vbs

**🧭 TL;DR – Kolejność pierwszego uruchomienia:**

| **Kolejność** | **Plik** | **Cel działania** |
| --- | --- | --- |
| 1 | git\_credentials\_enable.bat | Konfiguracja Git + tokenów |
| 2 | reset\_token\_history.bat | Czyszczenie historii Git z tokena |
| 3 | git push --force --all (ręcznie) | Push czystego repo |
| 4 | run\_and\_push\_FIXED.bat | Backup + push |
| 5 | push\_with\_token\_safe.bat | Manualny push |
| 6 | register\_backup\_task.vbs | Automatyzacja |