

## POC - 4

# Creating a Simple Backup Script for a Git Repository on Windows

### **Step 1: Create the Backup Script**

1. Open Notepad or any text editor
2. Make changes according to your preferences

@echo off

set SOURCE=c:/Users/pcabd/codsoft

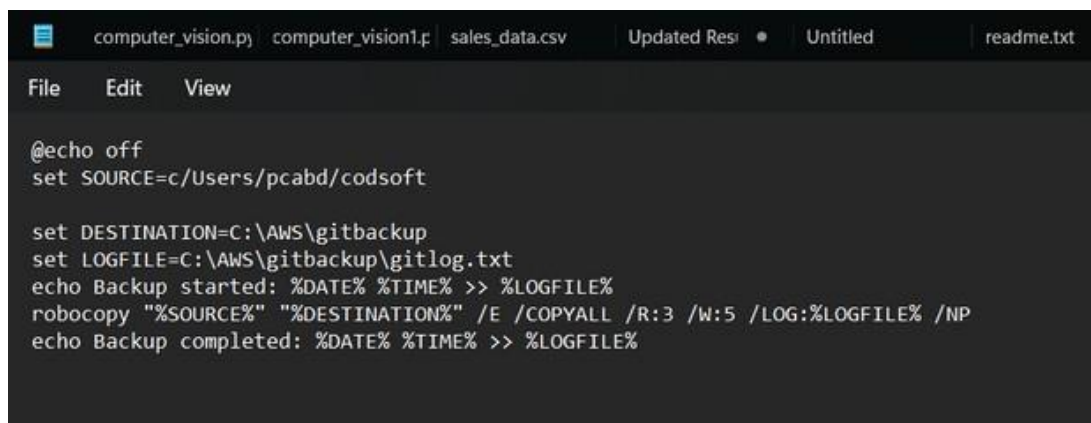
set DESTINATION=C:\AWS\gitbackup

```
set LOGFILE=C:\AWS\gitbackup\gitlog.txt
```

```
echo Backup started: %DATE% %TIME% >> %LOGFILE%
```

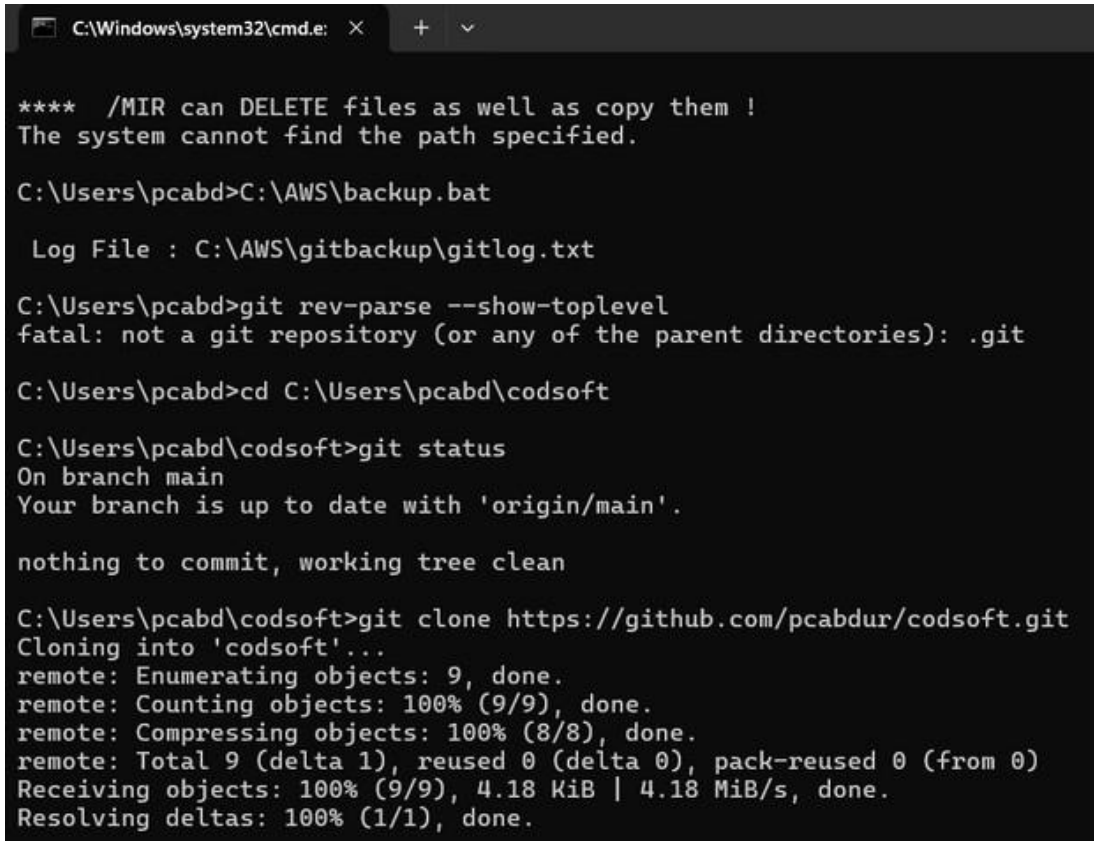
```
robocopy "%SOURCE%" "%DESTINATION%" /E /COPYALL /R:3 /W:5  
/LOG:%LOGFILE% /NP
```

```
echo Backup completed: %DATE% %TIME% >> %LOGFILE%
```

A screenshot of a Windows command prompt window. The title bar shows several open files: 'computer\_vision.py', 'computer\_vision1.py', 'sales\_data.csv', 'Updated Res...', 'Untitled', and 'readme.txt'. The menu bar includes 'File', 'Edit', and 'View'. The command prompt contains the following text:

```
@echo off  
set SOURCE=c/Users/pcabd/codsoft  
  
set DESTINATION=C:\AWS\gitbackup  
set LOGFILE=C:\AWS\gitbackup\gitlog.txt  
echo Backup started: %DATE% %TIME% >> %LOGFILE%  
robocopy "%SOURCE%" "%DESTINATION%" /E /COPYALL /R:3 /W:5 /LOG:%LOGFILE% /NP  
echo Backup completed: %DATE% %TIME% >> %LOGFILE%
```

why because this is your repo and the repo is copy in your backup when its is starts to run



```
C:\Windows\system32\cmd.e: X + v

**** /MIR can DELETE files as well as copy them !
The system cannot find the path specified.

C:\Users\pcabd>C:\AWS\backup.bat

Log File : C:\AWS\gitbackup\gitlog.txt

C:\Users\pcabd>git rev-parse --show-toplevel
fatal: not a git repository (or any of the parent directories): .git

C:\Users\pcabd>cd C:\Users\pcabd\codsoft

C:\Users\pcabd\codsoft>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

C:\Users\pcabd\codsoft>git clone https://github.com/pcabdur/codsoft.git
Cloning into 'codsoft'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (9/9), 4.18 KiB | 4.18 MiB/s, done.
Resolving deltas: 100% (1/1), done.
```

## Step 2: Automate with Task Scheduler

1. Open **Task Scheduler** (Win + S, search "Task Scheduler").
2. Click **Create Basic Task**.
3. Enter a **name** (e.g., "Git Backup") and **description**, then click **Next**.
4. Select **Daily** (or any frequency you prefer) and click **Next**.
5. Set the **time** for the backup to run and click **Next**.
6. Choose **Start a Program** and click **Next**.
7. Click **Browse**, select backup.bat, and click **Next**.
8. Click **Finish** to save the task.

## Step 3: Verify Automation

1. In **Task Scheduler**, find the task under **Task Scheduler Library**.

2. Right-click the task and select **Run** to test it.
3. Check C:\GitBackup and backup\_log.txt to ensure the backup was created.

# THANK YOU !!